UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Eduardo Seiti de Oliveira

Towards the effective delivery of an affordable Classroom Response System

Considerações sobre a implantação efetiva de um sistema de baixo custo de resposta em sala de aula

Campinas

2018

UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Eduardo Seiti de Oliveira

Towards the effective delivery of an affordable Classroom Response System

Considerações sobre a implantação efetiva de um sistema de baixo custo de resposta em sala de aula

Supervisor: Prof. Dr. Eduardo Alves do Valle Junior

Master's dissertation presented to the Post Graduate Program of the School of Electrical and Computer Engineering of the University of Campinas to obtain a Master's degree in Electrical Engineering, in the area of Computer Engineering.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Elétrica, na área de concentração Engenharia de Computação.

This volume corresponds to the version of the dissertation submitted to the examining board by Eduardo Seiti de Oliveira, under the supervision of Prof. Dr. Eduardo Alves do Valle Junior

Este exemplar corresponde à versão da dissertação apresentada à banca examinadora pelo aluno Eduardo Seiti de Oliveira, sob orientação de Prof. Dr. Eduardo Alves do Valle Junior

Campinas

2018

Agência(s) de fomento e nº(s) de processo(s): Não se aplica.
ORCID: https://orcid.org/0000-0002-7882-6203

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

OL4t	Oliveira, Eduardo Seiti de, 1975- Towards the effective delivery of an affordable Classroom Response System / Eduardo Seiti de Oliveira. – Campinas, SP : [s.n.], 2018. Orientador: Eduardo Alves do Valle Junior. Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Aprendizagem centrada no aluno. 2. Sistema de consultas e respostas. 3. Interação em sala de aula. 4. Processamento de imagem. 5. Aplicativos móveis. I. Valle Junior, Eduardo Alves do. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Considerações sobre a implantação efetiva de um sistema de baixo custo de resposta em sala de aula
Palavras-chave em inglês:
Student-centered learning
Query and response system
Classroom interaction
Image processing
Mobile applications
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Eduardo Alves do Valle Junior [Orientador]
Julio Cesar dos Reis
Alim Pedro de Castro Gonçalves
Data de defesa: 20-03-2018
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA – DISSERTAÇÃO DE MESTRADO

Candidato: Eduardo Seiti de Oliveira RA: 940011

Data da Defesa: 20 de março de 2018

Título da Tese: "Towards the effective delivery of an affordable Classroom Response System"

"Considerações Sobre a Implantação Efetiva de um Sistema de Baixo Custo de Resposta em Sala de Aula"

Prof. Dr. Eduardo Alves do Valle Junior (Presidente, FEEC/UNICAMP)

Prof. Dr. Julio Cesar dos Reis (IC/UNICAMP)

Prof. Dr. Alim Pedro de Castro Gonçalves (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

Acknowledgements

I would like to thank the entire research team I have participated throughout the development of this work. The criticisms, discussions, and insights composed the collaborative work that has directly influenced the achieved results. In fact, this research was born from a previous effort of the group, and I owe them that legacy.

Foremost, the experience of being part of this group, following the discussions and details of all other research works, was what ultimately nourished my development as a researcher.

Special thanks to Lin Tzy Li for proofreading the final version of this manuscript.

We acknowledge the paperclickers original research team: Wilson Prata and Vinícius Ribeiro, for conducting the user tests, Lucas Tejada, who implemented the augmented reality first prototype, Jomara Bindá and Prof. Dr. Renato Lopes, for the overall research organization and conduction. Their work initiated the pursuit of social impact through technology in education.

Motorola Mobility Brazil and Samsung Eletrônica da Amazônia kindly provided devices used to test paperclickers.

Abstract

We propose the effective delivery of a classroom response system (CRS) has to overcome a series of infrastructural and psychological restrictions, intimately related to the technology used as well as to the intended target audience.

We carry on the research to create paperclickers, a low-cost CRS system, which requires a single mobile device for the teachers to capture students responses during a class, provided through paper cards with printed codes. We kept aiming at broadening the adoption of active learning techniques in developing countries, offering a tool for straightforward implementation and associated with Peer Instruction methodology; our specific goals are to analyze and reduce the existing adoption barriers, focusing on Brazilian public high school teachers.

We compiled and analyzed the results of the first usability tests round, performed by the paperclickers initial research; we then described how the findings affected the tool usability. We tackled the new challenges on the TopCodes machine encoding, the solution applied on the answering cards, related to the detection and decoding procedures in the classroom environment, which is very different from TopCodes original usage scenario. We proposed additional processing steps to improve the detection and decoding robustness; we then performed experiments to evaluate how those changes affected the overall solution usability. The resulting paperclickers version is currently available for the public at large as an open-source release.

We also designed the first part of training video tutorials, covering both paperclickers and Peer Instruction usage, illustrating the material to be created for the selected target audience, aiming to reduce the psychological adoption barriers, towards an effective delivery of our solution.

Resumo

Defendemos que a efetividade da implantação de um sistema de resposta em sala de aula depende da superação de uma série de restrições, tanto infra-estruturais quanto psicológicas, intimamente relacionadas com a tecnologia utilizada e com o público alvo pretendido.

Demos sequência à investigação da criação de um sistema de baixo custo de resposta em sala de aula, o paperclickers, que requer um único dispositivo móvel para o professor capturar respostas em sala de aula, fornecidas pelos alunos através de cartões com códigos impressos. Mantivemos o objetivo de fomentar a adoção de técnicas de aprendizagem ativa em países em desenvolvimento, oferecendo uma ferramenta de fácil implementação e associada a uma metodologia de ensino específica — a Instrução pelos Pares. Mas acrescentamos o enfoque de analisar e atuar sobre as possíveis barreiras de adoção, considerando como público alvo professores de ensino médio de escolas públicas brasileiras.

Compilamos os resultados dos testes de usabilidade realizados durante a pesquisa original, e descrevemos como a interpretação desses dados afetou a usabilidade da versão atual do software. Tratamos dificuldades de detecção e decodificação dos cartões de respostas, decorrentes do novo e dinâmico cenário de uso dos TopCodes, a codificação escolhida para nossa solução, muito diferente das suas condições originais. Propusemos e experimentamos melhorias de robustez no processamento dos TopCodes, analisando como a aplicação dessas melhorias afetou a usabilidade global da solução. Disponibilizamos paperclickers para o público em geral, numa versão inicial e de código aberto.

Projetamos também a primeira parte de uma série de tutoriais em vídeo, para treinamento tanto no uso do paperclickers quanto da metodologia de Instrução pelos Pares. Com isso, ilustramos o material a ser criado para nosso público alvo, com a intenção de reduzir as barreiras psicológicas de adoção, focando na efetividade de implantação da nossa solução.

List of Figures

List of Tables

Contents

1 Introduction

Delivering technology for pedagogy is challenging; effective delivery depends on technical, infrastructural, and human factors. The traditional, technical perspective focuses on infrastructure: create the technology first, then think about user experience. The third wave of Human-Computer Interaction research [Bødker, 2015] subverted this logic, putting users on the forefront. Technology applied to Education was no different [Almeida and Valente, 2016]: governmental policies frequently favored the creation of infrastructure and content; currently, integration on processes, and acceptance by people are understood as critical.

Teaching is a complex activity, requiring knowledge from different areas: at the very least pedagogy, and the specific subject being taught. Technological expertise on teaching tools is a burden few teachers can afford. The challenge is compounded when the technological intervention requires (or aims at) changing pedagogical practices and processes — requiring from teachers motivation for change and learning [Hao and Lee, 2016].

This dissertation follows Bindá [2015], whose authors have described the design, prototyping and user evaluation of *paperclickers*, an affordable Classroom Response System (CRS) aimed at fostering the use of active learning by disfavored communities. CRSs allow polling the students in real time, easing the dialog between instructors and learners. CRSs are often implemented as ‘clickers’, small remote-controls that send answers to an infrared or radio-frequency receiver, as illustrated in Figure 1, but such solution involves many direct and indirect costs.

In this work, we use CRSs as a case study for the deployment of technology for education. Our goal is to understand how the release of paperclickers could facilitate the adoption of an active learning methodology like Peer Instruction.

This dissertation belongs to the broader issue of how academic research can achieve social impact and help the most disfavored communities. All our work — including our survey of the literature — was conducted while seeking answers to that challenging question.

1.1 Motivation

Our primary motivation is to promote active learning methods, which in a straightforward definition are *“anything course-related that all students in a class session are called upon to do other than simply watching, listening and taking notes"* [Felder and

Figure 1 – Typical clicker embodiment as small radio-frequency or infrared devices. Reproduced from https://www.iclicker.com/instructors.

Brent, 2009]. Although those methods promote higher learning gains than regular lecturing [Hattie, 2009], the latter still prevails in the classrooms [Smith and Valentine, 2012; Eagan et al., 2014].

Tools like Classroom Response Systems (CRS) can facilitate constant feedback between teachers and their students, a central aspect of active learning methods. Broadening the usage of CRS might be a factor in fostering the adoption of Peer Instruction, a proven active learning methodology.

Clicker solutions involve several costs, from acquiring the devices, installing the receivers, training the personnel, and managing the operation (e.g., dealing with batteries, etc.). The total cost of the infrastructure is often unfeasible for schools in developing countries. To address that issue, previous work [Bindá, 2015; Tejada, 2014; Ribeiro et al., 2015; Neto, 2015] studied the proposal of *paperclickers*, an image processing CRS, prototyped as a smartphone application, using the camera to scan the classroom for the students' answers. Paperclickers solution is easy to use, it does not require Internet access to operate, and requires a single hardware device per classroom — which can be the smartphone that the teacher already owns.

Paperclickers was designed and prototyped. Usability tests were conducted, but not analyzed, and knowledge from those tests was not acted upon. This dissertation starts from that pointm analyzing the user experiments and completing a new cycle of development, and achieving public release.

Since the inception of Peer Instruction in 1991, several researchers [Mazur, 1997;

Crouch and Mazur, 2001; Vickrey et al., 2015] presented evidence on its effectiveness, at the same time indicating the complexity of applying a new pedagogical methodology. Novelties within the classroom require motivational and attitudinal changes both on teachers but also on the students — *"Peer Instruction requires students to be significantly more actively involved and independent in learning than does a conventional lecture class"* [Crouch and Mazur, 2001] — besides the knowledge on how to apply new methods or to use new tools.

Studies about technology adoption on educational activities confirm that understanding, providing analytical tools for the intricate relationships among the knowledge involved when applying a new technology inside the classroom [Koehler and Mishra, 2009]. Those studies also provide insights about the new technology acceptance [Venkatesh et al., 2003], and depicts the myriad of factor influencing the technology adoption within a school [Osterweil et al., 2016] or even an entire country [KENNISNET, 2015; Almeida and Valente, 2016].

Backed by that literature, we understand paperclickers needs to be packaged along with training material, covering both the tool usage and the active pedagogical practices it facilitates — Peer Instruction — to promote the most effective adoption of our CRS solution. That understanding was the basis for the development and analysis of paperclickers official release.

Lectures are rooted in the ancient form of knowledge transmission, applied long before mechanical printing became the main recipient of human knowledge. In the past decades, digital technologies and the Internet had transformed once again that landscape, but lectures persist, since *"we tend to teach the way we were taught"* [Mazur, 1997]. Training on teaching methods and new tools is required to transform that tradition.

1.2 Contributions

The major contributions of this work are the following:

- The discussion of the challenges related to create a image processing based classroom response system, especially from the user experience perspective.
- Improvements on the detection and decoding process of TopCodes machine encoding, increasing its robustness for its new usage scenario in image-processing CRSs.
- A discussion about the challenges of achieving social impact through the research and development of a technological pedagogical tool.
- Paperclickers and Peer Instruction training material outline, targeting Brazilian public middle-schools teachers.

- The release of an open-source image-recognition based classroom response system, establishing a baseline for further research and development.

1.3 Outline

This work is organized in the following parts:

Chapter 2 presents the state-of-the-art on the related studies: Sections 2.1, 2.2 and 2.3 present the pedagogic effects of using CRSs, detailing experiments with image processing CRS alternatives, and establishing the importance of feedback among instructors and students. The final sections present the current discussion about the effective use of technological pedagogical tools and solutions. Section 2.4.1 considers restrictions related to general environmental aspects — infrastructure, organizational and even political perspectives — while Section 2.4.2 discusses that from the perspective of the people involved.

Chapter 3 presents paperclickers CRS, delineating its creation by the original research in Section 3.1; Section 3.2 presents the compilation and analysis of the previous user experiences. The improvements in application flow and features, corresponding to the findings, are described in Section 3.3.

Chapter 4 describes the contributions on the paperclickers answering cards processing, presenting in Section 4.1, the detection and decoding issues TopCodes solution faced in our usage scenario. Section 4.2 details all the experiments executed to verify the effectiveness of the proposed changes to improve the detection and decoding robustness of paperclickers.

Chapter 5 discusses the factors and restrictions influencing the efficient deployment of a technological pedagogical tool, to reach the targeted users and to achieve social impact. Facing those restrictions, we embedded usage instructions in our solution, Section 5.2, and we proposed in Section 5.3 the design of a training material focused on reducing the adoption barriers from the teachers.

Chapter 6 provides a closure to the developed work by compiling and interpreting the overall results, and highlighting the achievements, the challenges and the future work towards the effective delivery of an affordable Classroom Response System (Section 6.2).

Parts of this present document — especially sections 2.1, 2.2, 2.3, 3.1, 3.3, 4.1, 3.2 and 6.1 — are excerpts from the preprint Oliveira et al. [2017].

2 Literature Review

This survey follows our research journey: we started looking for a way to promote active learning within the classroom, reaching the classroom response systems as a tool to facilitate question-driven methodologies. With the hypothesis that an easy-to-use CRS would promote such learning methodologies, we studied Peer Instruction, a proven question-driven and active learning methodology. Since PI is not widely employed, we broadened our studies looking for works investigating the adoption barriers for technological pedagogical tools and active learning methodologies. We verified the centrality of the human elements involved, especially the teachers, for the effective adoption of innovations inside the classroom; we then investigated the psychological aspects moderating the teachers' adoption of new technological pedagogical tools and teaching methodologies.

We surveyed three aspects related to our main case: CRS effectiveness in promoting learning gains (Section 2.1); the pedagogically proven results of facilitating feedback among students and teachers; and the Peer Instruction methodology (Section 2.3).

We then surveyed more broad works, focused on the challenges of deploying new technology for pedagogy (Section 2.4). We started with the study of a general analysis framework, created to evaluate the alignment of a technological pedagogical intervention in a given environment; we then analyzed a model asserting the successful deployment of technologies on education depends on the correct balance among specific factors (Section 2.4.1). We completed the analysis (Section 2.4.2) with studies focused on the psychological aspects affecting the people using the new technologies during the educational process, mostly the teachers, who have the ultimate responsibility for implementing any pedagogical action.

We conclude this chapter considering that literature lacks actionable instructions to guide the deployment of new technologies for education, even though several works explore the state and adequacy of a given technological pedagogical tool, or also propose interpretations for the adoption restrictions.

2.1 CRSs effectiveness

Few years after the initial studies on CRS usage, in early 1990s [Beatty, 2005; Lane and Atlas, 1996], researchers started to investigate the tool pedagogical effectiveness [Hunsu et al., 2016]. Some of the highlighted advantages of CRS use included the following: the constant monitoring of students understanding throughout the classes; the increase of students engagement, mainly due to the anonymity it enables; the indirect benefits as the automatic class attendance recording. Overall, the major positive aspect of CRS teaching usage is the fact it facilitates question-driven pedagogical methods, which are examples of active learning.

The Hunsu et al. [2016] meta-analysis investigated the cognitive and non-cognitive effects of clicker-based technologies usage when compared to conventional lecture classes, establishing a unique scale over 53 selected articles, covering a total of 26,085 participants. That work considered 111 independent learning outcomes, coded from all the variables present on the original studies, allowing their effect sizes comparison through 86 cognitive and 25 non-cognitive outcomes. The meta-analysis suggests CRS effects are small to medium on both non-cognitive (*engagement and participation, self-efficacy, attendance, perception of quality, interest, and likeness*) and cognitive aspects. It indicates CRS usage improves students' participation in large lecture halls, a typical environment in STEM courses and particularly challenging to achieve students engagement. There are also small but positive cognitive learning outcomes from clicker-based technologies usage, especially on the higher learning goals like *knowledge transfer or knowledge application*; measurable learning gains vary on several aspects, being the knowledge area one of them — most probably related to its adequacy to question-driven methodology and peer discussions [Hunsu et al., 2016].

Hunsu et al. state CRS positive pedagogical effects can be related to the classroom dynamic they facilitate and enforce. Hence, CRS usage might not directly be responsible for the learning gains, but it can promote the adoption and augment the learning efficiency of an adequate learning methodology — most of the learning gains come from the pedagogical method, not from the employed tool [Hunsu et al., 2016]. CRS, as a tool, can represent not only a facilitator for specific class practices, but it can also be a stimulus for applying a new pedagogical method.

Table 1 presents the available CRS solutions, comparing them regarding features, advantages, and difficulties of use. That information indicates some of them privilege easiness of usage, with a limitation of features, while others include more features, but require additional training effort either from teachers and instructions, but also from students.

2.2 Image processing CRSs

Image processing CRSs minimize costs by giving the students passive devices, usually cards with distinctive colors or codes, and concentrating all the active processing into a single device, which remains with the teacher. Most often, the students use a card printed with a 2D barcode, which serves both as a location and orientation marker, and as a unique ID for each student. The students can answer multiple-choice questions by

Method	**Advantages**	**Disadvantages**
"Low-tech" alternatives (show of hands, color cards)	Very low-cost Available immediately everywhere Very easy to use	Classmates tend to "follow the majority" Individual answers unrecoverable Only multiple-choice answers possible
Dedicated hardware ('clickers')	Wide commercial availability Classmates cannot see answers Instructor recovers individual answers Moderate to complex answers possible	High direct and indirect costs Complex training required for teachers
SW on students' devices (BYOD)	Good commercial availability Classmates cannot see answers Instructor recovers individual answers Very Complex answers (e.g. drawings) possible Low-cost for institutions	High-cost for students Devices can be distracting Requires reliable network infrastructure Training required for teachers and students
SW on teachers' device + cards with codes for students (image processing)	Low-cost for students and institution Classmates cannot see answers Instructor recovers individual answers Simple training required for teachers, virtually no training for students	Few (mostly experimental) solutions Only multiple-choice answers possible Requires line of sight to each student

Table 1 – Summary of classroom response system technologies. Image processing CRSs — like paperclickers — are the only ones at the intersection of low cost, simplicity, anonymity to classmates, and trackability of answers by instructors. Reproduced from Oliveira et al. [2017].

rotating the cards.

Amy and Amy [2015] patented a low-cost optical polling framework with a generic computing element, which recognizes the orientation of fiducial marks on printed cards. The proposed solution is available as a smartphone app — Plickers[1] —, which currently accommodates up to 63 students, who must enroll on a web-based system. Fiducial markers on printed cards had already appeared on previous works, as the augmented reality system ARTag Fiala [2005]. Amy and Amy [2015] innovate by exploiting them for low-cost CRSs; however, as a commercial solution, it requires Internet connectivity to be used — even for the smartphone application sign in — which might be a problem on no-connectivity scenarios. Also, that proposed solution does not describe the challenges related to recognizing fiducial markers in a classroom environment — which indeed imposes size and encoding power restrictions to the student cards.

Cross et al. [2012] also proposed a system that recognizes the answers through the orientation of printed cards with unique IDs for each student, which they called 'qCards'. The teacher captures the responses with an off-the-shelf webcam mounted on a laptop with software to recognize, tabulate, and display the results. The authors ran initial trials on secondary schools in Bangalore, India, with 99.8% recognition accuracy, and 97% captured responses in a 25-student classroom.

Miura and Nakada [2012] presented similar work: they used printed cards with fiducial markers as codes, a similar camera setup, and PC running the software. Their

[1] https://plickers.com/

system recognizes three rotational parameters for each card (roll, pitch, and yaw), allowing students to select one of many possible multiple-choice answers in a screen. A preliminary experiment with 19 students, succeeded in tracking 18 markers.

The solution proposed by Gain [2013] uses cardboards with printed colored blocks and a camera-phone to capture images. Students select answers by picking different colors. They report 85% recognition accuracy in a medium-size class (up to 125 students). Although the system forgoes peer anonymity, and the possibility to track responses to individual students for later analysis, it is the only image processing system tested in classes that big.

Finally, Ito and Miura [2015] experimented a portable version of that previous system, recognizing the same fiducial marks in a tablet computer, including the capability of detecting the response printed card bending amount as an additional input mechanism — it could encode, for instance, the student mood.

As far as we know, paperclickers is the only existing image processing CRS solution at the intersection of being an academic work, having its entire source code publicly released, and being available for download on user's devices for actual, practical use. Being on that intersection allows future contributors to quickly test hypotheses and add improvements to paperclickers, with real-world impact to users.

2.3 Feedback and Peer Instruction

According to Hattie [2009], feedback is one powerful answer for effective teaching and learning, specially when it is *"from the student to the teacher"*. In other words, *"when teachers seek, or at least are open to, feedback from students as to what students know, what they understand, where they make errors, when they have misconceptions, when they are not engaged"*.

Feedback needs to provide information directly related to the task or process of learning — instead of focusing only on exercises correctness or the students, as praising, punishing or giving external rewards — to effectively work on filling *"the gap between what is understood and what is aimed to be understood"*. However, feedback should be based on something: it is what happens after an instruction is provided through a pedagogical method [Hattie, 2009].

Peer Instruction (PI) was created in 1991 [Mazur, 1997] as a structured questioning process, defined to establish feedback among students and teachers. PI fosters a visible learning process which is *"when teachers become learners of their own teaching, and when students become their own teachers"* [Hattie, 2009].

PI aims to engage students in the learning process as an adaptation of traditional lectures; that is one aspect which encourages the methodology adoption in different knowledge areas [Crouch and Mazur, 2001]. PI improves students scores related to content concepts verification (formalized by the called ‘concept tests’), but also for quantitative problems resolution — regardless the fact PI methodology intentionally moves the focus from the quantitative/repetitive problem solving or rote memorization techniques.

Peer instruction most underscored characteristic is the students’ discussion inside the classroom, when they are requested to convince their peers about the correctness of their own answers to a concept test. However, that dynamic can be considered just the summit of an active learning methodology, composed by several stages, resulting from an attentive class preparation which aims to create opportunities for active discussions and feedback among students and teachers.

A particular advantage of PI is the fact during the peer discussions, students can easily address their difficulties, since they are not affected by the *curse of knowledge* — when a common understanding background is assumed [Mazur, 1997]. Therefore, PI massively parallelizes personalized teaching, inducing the one-to-one discussions.

The following steps describe Peer Instruction process:

1. The definition of a reading assignment to be completed before class — employed to reserve in-classroom time for the discussion activities;
2. Students completion of reading incentive questionnaire, aiming to verify the reading completion, not the concepts conveyed — used to be an in-classroom reading quiz;
3. Presentation of a short lecture focused on specific key-point(s);
4. Posing a concept test, aiming to verify the concept(s)/key-point(s) presented in the short-lecture;
5. Students commitment to their own answers, without talking to each other;
6. First polling for students answers;
7. Students discussion with their neighbors, trying to convince each other about their responses, explaining their reasoning; instructors observation (without interfering) to identify students reasoning, questions, and doubts;
8. Second polling for new students answers, on the same concept test;
9. Instructor final explanation about the correct answer, considering the overall classroom voting results and the impressions gathered from the discussion phase.

Steps 3 to 9 can repeat during a class, depending on the teachers’ planning. Figure 2 illustrates the PI cycle, emphasizing it can repeat during the class dynamic.

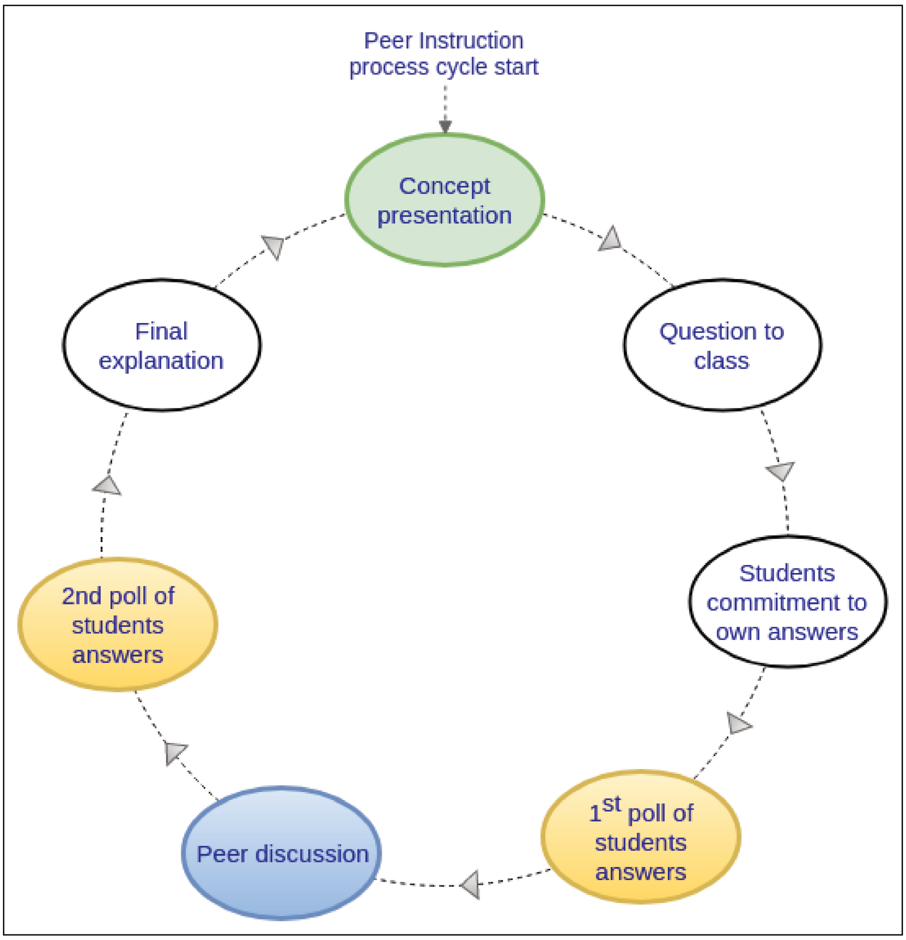

Figure 2 – PI defines a structured question-answer process, with the peer discussion as the main phase, when the students have the opportunity to deepen their understanding.

Although PI requires teachers engagement to prepare the classes, Crouch and Mazur [2001] reports the overall time spent does not increase, since lectures careful preparation also takes extensive time, especially when the instructor tries to anticipate and cover all possible questions the students might pose during the lecture.

Also according to Crouch and Mazur [2001], despite corresponding to clear and specific steps, Peer Instruction admits adaptations, being able to accommodate different learning contexts, needs, and teaching styles. Some customizable parameters are the number of concepts covered in a single class; the choice of questions applied; the time devoted to each question; the amount of lecturing between questions and during each class; requiring or not reading assignments before classes.

Vickrey et al. [2015] meta-analysis reviewed 56 studies conducted at STEM college level, and reporting PI implementation mostly in large (>50 students) classrooms. Their compilation indicates PI has measurable learning gains, with *"regularly twice as large as those observed with traditional lecture"*. PI also improves problem-solving skills, with some researches indicating students in PI groups showed increased *"ability to answer questions designed to measure mastery of material"* and *"to solve novel problems (i.e. transfer* *knowledge)"*, as well as improvements in *"quantitative problem-solving skills"*. Finally PI also reduces attrition rates, with several studies reporting dropout-rate (difference between the number of students enrolled and taking the final exams) reduction, lower failure rates and increased retention.

The evidence-based analysis of PI implementation compiled by that meta-analysis also reveals the relevance of each PI step, indicating how changes in the methodology implementation might affect the overall results. The study also suggests additional specification for the PI process, depicted in figure 3, establishing heuristics for executing each step.

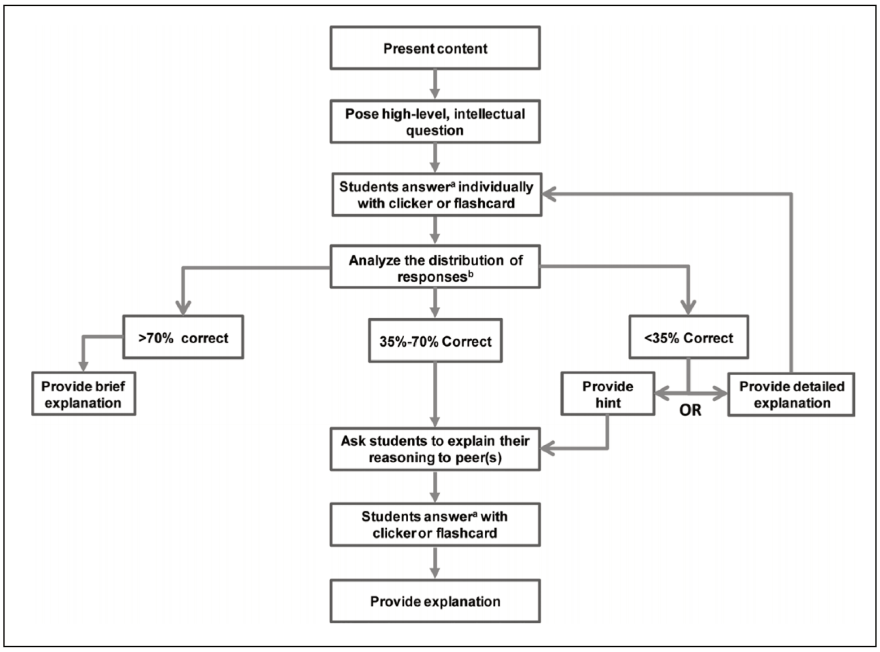

Figure 3 – PI process suggested heuristics, reproduced from Vickrey et al. [2015].

2.4 Fostering technology usage on education

Deploying a new technology for education is a complex and difficult task, depending on several variables and frequently facing opposition from different aspects, since physical devices availability restrictions — including all the associated services and support — but also procedural and even political aspects, as well as other human based resistance. Especially when we consider the reality of disfavoured communities, those variables might play prohibitive restriction roles, which condemn the technology to become marginal or even to fail to be adopted.

When thinking about the human restrictions to the usage of a new technology, psychological aspects need also to be included, since no novelty will be implemented inside a classroom if the teachers or instructors are not convinced — as well as motivated

— to use it, frequently changing their common teaching behavior and style, leaving the comfort zone created by familiar habits.

2.4.1 Educational technologies adoption restrictions

The Comprehensive Initiative on Technology Evaluation (CITE) [2] is an interdisciplinary program at the Massachusetts Institute of Technology which aims to develop methods for products evaluation, especially focused on analyzing their suitability for developing areas — *"does a product perform its intended purpose?"* —, their scalability — *"can the supply chain effectively reach consumers?"* — and their sustainability — *"is a product used correctly, consistently & continuously over time?"* [Osterweil et al., 2016]. The **CITE framework** was proposed as a tool to evaluate the appropriateness of educational technology use in global development programs, depicting an extensive list of variables influencing the effective deployment of a given educational technology intervention; the analysis of those variables measure the compatibility degree of that action within a specific learning environment. More than determining if an intervention should or should not be applied, that framework helps understanding where — an even how much — effort is needed for the proper and effective use of the educational technology in the specific scenario.

The framework is organized in the form of questions, grouped into eight areas which identify concerns and variables affecting and restricting the delivery of new educational technology. The framework poses critical questions to evaluate the openness to the new technology, ranging from basic infrastructure questions (*"There should be reliable electricity available to use computers? Can this need be moderated by the use of smartphone applications?"*), and exploring the needs and concerns of various stakeholders (like political and community representatives, students and teachers). Table 2 lists those major areas and their sub-sections.

Two of the framework areas — **Infrastructure** and **Sustainability** — uncover the challenges related to the physical infrastructure required by the pedagogical technology: what is needed first to deploy it, and then to keep it working. Other areas — **Community, Social, Political**, **Scalability & Market Impact** and **Culture** — discuss the initiative dependence of external actors, like politics or technology facilitators, and even the need for approval of broader audiences, like having the community approval of the technology, or alignment with cultural characteristics. Finally, the last three areas — **Teachers**, **Students** and **Learning** — explore the human factors directly linked to applying the pedagogical activities within the classroom.

Table 3 depicts the framework questions for the **Teachers** area. Those questions explore several aspects from the teachers perspective, aiming to build a clearer picture of

[2] http://cite.mit.edu/

Teachers	Community, Social, Political
Comfort Competence Openness to Change Role Classroom Management	Implementation Support
Students	**Learning**
Comfort Access Openness to Change	Learning Goals/Impact on Learning Pedagogy Curriculum
Culture	**Infrastructure**
Culturally Relevancy	Equipment Electricity Internet
Sustainability	**Scalability & Market Impact**
Funding Maintenance & Repairs	Broader Community Impact Adoption & Scaling

Table 2 – The areas in the MIT framework cover several aspects which can impose restrictions to the deployment of a new technological pedagogical tool in a given environment. From Osterweil et al. [2016]

how they are prepared to work with the new technologies — the **Competence**, **Role**, and **Classroom Management** sub-areas —, and how opened the teachers are to work with those new technologies — **Comfort** and **Openness to Change** sub-areas. Similarly, the **Students** area explores the students perspective on similar aspects. Finally, the **Learning** area focuses on exploring the alignment of the new technology and the educational methodologies in use at the learning institution; with questions exploring the **Learning Goals / Impact on Learning**, **Pedagogy** and **Curriculum** sub-areas, this component is particularly relevant when the new tool implies a new educational methodology.

We considered this framework relevant since it clearly indicates the need for knowing the characteristics of each specific learning reality, where a new technological pedagogical tool will be deployed, as a condition for understanding its adequacy and, therefore, its effective usage.

This framework provides a broad and structured view of the factors restricting the deployment of a new technology, considering a specific learning reality. Directly working on the aspects explored by those questions, would increase the probability of success on having a developed learning tool truly used by the intended audience.

Another work specifically focused on developing and evaluating ICT usage on education, the **Four in Balance model** was developed by the Kennisnet, a public or-

Comfort
Comfort with Technology How comfortable are the teachers with technology? In terms of general use as well as in an educational setting.
Comfort with Teaching Students Technology How comfortable are teachers in teaching students how to use the technology? As is, and then with additional training.
Competence
Professional Development Required How much learning of the technology would teachers need? And what is the structure? (one day vs. multiple sessions?)
Resources for Professional Development Who would provide the instruction? Outside vs. in-school employee
Professional Development Scheduling When would the instruction happen? Are additional work hours needed?
Professional Development Costs What additional costs are associated with the instruction? Do the teachers, school, or technology company cover these costs?
Openness to Change
Learning Technology Are teachers willing to learn how to use the technology? How much time are they willing to put in to learn how to use the technology? Is there an associated job training benefit of learning the technology?
Learning New Pedagogies Are teachers willing to change their pedagogy to accommodate the use of technology? Has it been made clear to teachers why they are using the technology? Is the technology in alignment with teachers' current learning goals for students? Is the technology in alignment with the school-wide goals for learning?
Role
Role with Technology What is the role of the teacher in the implementation of the technology? Is the technology seen as an "added responsibility" or a "teacher replacement" without any benefits? Is the technology perceived in a positive light, as a tool to aid in teaching/learning? How does the teacher interact with students using the technology?
Classroom Management
Monitoring Technology Use How will the technology use be monitored (so students cannot access inappropriate content)? Does the technology company put restrictions in place? Are the teacher/school responsible for monitoring content? Do they know how to effectively set up monitoring?
Demands by the Technology Does the technology create a burden of extra management for the teacher? Does the technology make learning more efficient and effective in terms of time for the teacher? Is the teacher aware of how the students are using the technology at an individual level? Does the teacher receive usage and progress reports or can they monitor usage easily? Does monitoring the usage take a lot of extra effort for the teacher?

Table 3 – The questions on **Teachers** area from MIT framework explore possible restrictions for a new technological pedagogical tool adoption, mostly considering the teachers competencies required to make proper usage of the tool and the pedagogical methodology it implies. From Osterweil et al. [2016].

ganization[3] responsible for Netherlands' national ICT-infrastructure and advise for the educational sector. The Four in Balance model is composed of four-axis which have to be in balance, in order to properly support the effective and efficient ICT usage in education. Those axes are grouped in human and technological elements [Almeida and Valente, 2016]:

Human element:

- **Vision**: the starting point for an effective and efficient ICT usage in education, the vision has to be shared among all educational instances, from the government until each school. It encompasses the definition of *"how an educational institution envis-*

[3] https://www.kennisnet.nl/about-us/

ages qualitatively sound and efficient education and what ICT's role is in achieving it" (KENNISNET [2015]).

- **Expertise**: includes all the competencies for the people involved in deploying, managing and indeed using ICT's in education must have. It emphasizes that the skills and knowledge required to effectively and efficiently make pedagogical use of ICT's differ from having ICT knowledge — it is necessary to know when, how and why to use ICTs during educational activities.

Technological element:

- **Content and applications**: this axis encompasses all the digital learning materials, educational software tools, and activity management packages. The chosen material has implications for an educational organization since it implies associated goals, ideas, and approaches.
- **Infrastructure**: the aspect traditionally associated with ICTs usage, this axis comprises all physical requirements, including their acquisition and maintenance.

Similar to the MIT framework, the Four in Balance model implies that the interdependence between the elements and their axes happens throughout instances of different levels — from the public government until the teacher inside the classroom. This model clearly indicates the dependence of people training and formation for the ICT usage in education, taking the educational vision as guidance [Almeida and Valente, 2016].

The Four in Balance model identifies the balance among the four axes as a condition for the technology deployment success on pedagogical activities: balance is required among the human an technological elements, and missing actions in any of them hinder the effective use of technology in education. In that sense, that model advances the discussion when compared to the MIT framework, which proposes only the exploration of several different factors affecting the technology adequacy for a pedagogical setup. The Four in Balance model, however, does not clearly indicates what consists that balance, and achieving it depends on particular analysis of each scenario.

Anyway, we considered this framework relevant because it also explores several aspects regulating the effectiveness of technology usage on pedagogical activities. However, it makes clear the relevance of the human elements, which are as crucial to the successful usage of technology as the technology — tools, infrastructure, and content — itself.

2.4.2 Teachers' adoption restrictions of educational technologies

In order to be effective, new technologies have to be embraced and used by their target audience.

Frameworks like the MIT and the Four in Balance already pointed to the importance of the human elements on the technological pedagogical tools acceptance; in fact, no innovation either technological or methodological will be effective inside the classroom, if the teachers are not motivated and confident to make successful usage of them.

Recognizing that fact, we expanded our literature review to include works specifically related to what we called psychological adoption barriers, to analyze the effect of factors like teachers and students motivation, confidence, comfort, performance and effort expectancies on the usage of technological pedagogical tools and their related methodologies.

Venkatesh et al. [2003] formulated the **Unified Theory of Acceptance and Use of Technology (UTAUT)**, analyzing eight theoretical models that compete to explain the user acceptance of Informational Communication Systems. From an experimental comparison of those models prediction power, the authors identified four constructs that play a significant role as technology acceptance determinant, measured through the intention to use and usage behavior dependent variables:

- **Performance expectancy**: the performance gain expectancy an individual has when using a given technology.
- **Effort expectancy**: the belief of how difficult is to use the technology.
- **Social influence**: the individual perception that the system should be used, in the opinion of people influential or relevant to the user.
- **Facilitating conditions**: the degree of belief an individual has about the existence of all the required conditions — both organizational and technical or infrastructural — to use the technology.

Those constructs were built from the theoretical and empirical similarities among the eight models, and they were considered to directly determine the *Behavioral Intention* — modulated by the **performance expectancy**, **effort expectancy**, and **social influence**. As shown on figure 4, those constructs also define the *use behavior* — influenced by **facilitating conditions**, besides the *Behavioral Intention* itself.

UTAUT represents a common baseline unifying the previous theories of technology acceptance; with the focus on analyzing the individual behavior, all those theories emphasize the relevance — and even the prevalence when explaining usage effectiveness — of human factors deriving from personal and social conditions, which are modeled by the intrinsic characteristics of a given technology.

Although not specific for technological pedagogical solutions, we considered this theory relevant to our work, since it clearly indicates the need of knowing the target

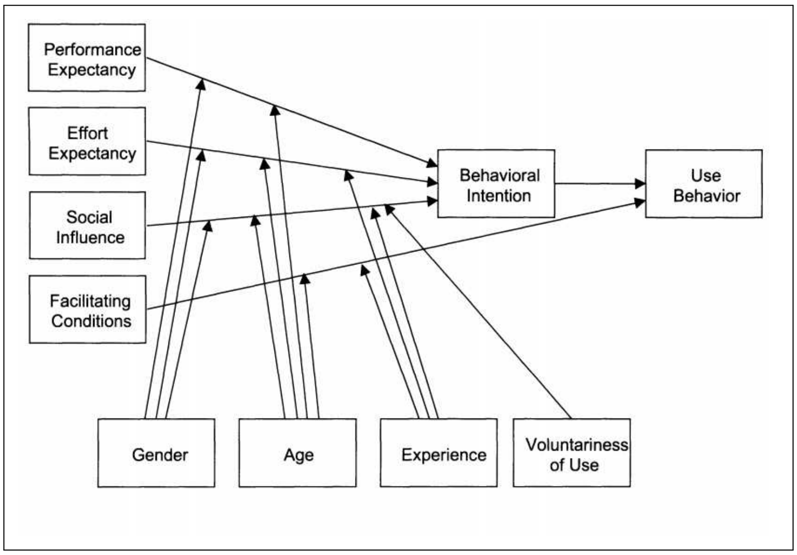

Figure 4 – Unified Theory of Acceptance and Use of Technology (UTAUT) constructs, their modulator factors, and their effects. Reproduced from Venkatesh et al. [2003].

audience when evaluating the psychological concerns related to the technology acceptance and effective usage.

Further perspectives are possible when investigating the factors influencing the technology adoption by the individuals, narrowing the analysis perspective to consider the human activities a given technology affects. Teaching is a complex activity which occurs in equally complex and dynamic classroom contexts, requiring the combination of specialized knowledge from different domains. Applying technology in teaching activities adds to that complexity. Koehler and Mishra [2009] state the fact digital technologies can be used in different manners, and they are unstable and frequently work in opaque ways, creates even more challenges to teachers, used to other technologies traditionally applied in educational practices.

Technologies are not neutral and imply certain applications, tasks, and understanding. Therefore they have a direct effect on pedagogical practices [Koehler and Mishra, 2009] — which is, in fact, a positive aspect when considering how a given technology can influence people's behavior: due to CRSs facilitation of question-driven methodologies, they can have a propensity to active learning methodologies.

Since technology changes rapidly, most teachers were trained under very different educational technologies they now have available to work with their students; the knowledge required to apply those new technologies must align with teachers' pedagogical beliefs, to be effectively integrated into their specific realities [Koehler and Mishra, 2009].

Koehler and Mishra [2009] developed the **Technology, Pedagogy, and Content Knowledge (TPACK)** framework, a way to describe the complexity of integrating tech-

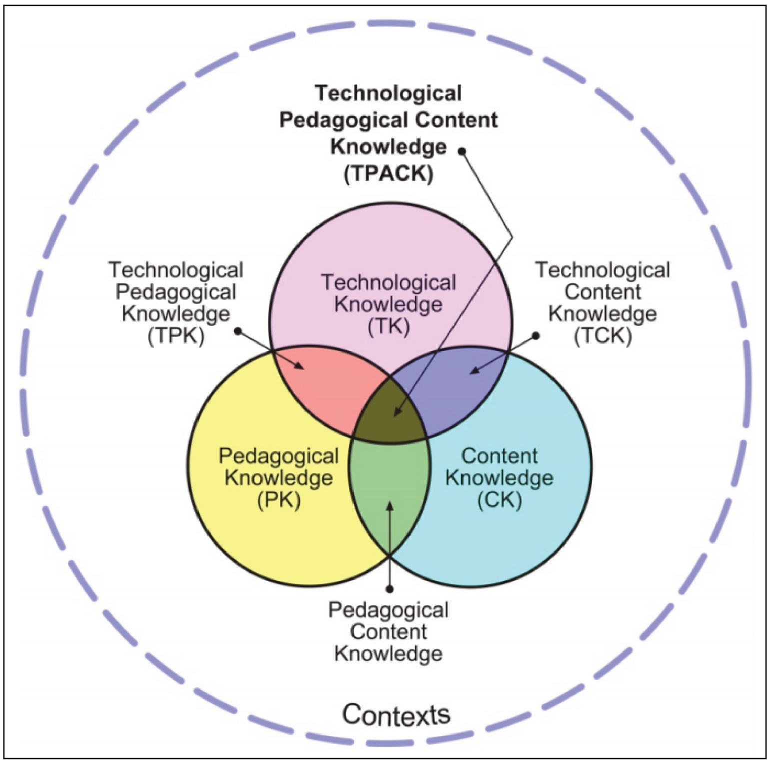

Figure 5 – TPACK framework, reproduced from Koehler and Mishra [2009].

nology in teaching activities. As shown in figure 5, TPACK is the resulting interaction among three main components — **Content Knowledge (CK)**, **Pedagogical Knowledge (PK)** and **Technological Knowledge (TK)**. CK is teachers' knowledge on the subject to be taught and learned; PK is the teachers' knowledge about the processes, practices or methods of teaching and learning; finally, TK is the teachers' knowledge about the technologies to be employed on the teaching activities [Koehler and Mishra, 2009].

TPACK builds on previous work, the Pedagogical Content Knowledge framework, adding the technological domain as a separated one in the body of knowledge teachers need to develop their work, clearly separating it from the two other domains — the pedagogical and content — which addresses the teaching methodologies and the knowledge specific of a given area.

It is possible to consider the interplay of each one of those knowledge areas, creating specific domains — **Pedagogical Content Knowledge (PCK)**, **Technological Content Knowledge (TCK)** and **Technological Pedagogical Knowledge (TPK)** — when the characteristics of each original domain are combined, composing nuances which do not exist separately. Hence, PCK is the knowledge about pedagogy applied to specific content, TCK is the knowledge on how the technology can transform the understanding of specific content and TPK is the knowledge about pedagogical practices enabled or transformed by specific technologies. Finally, TPACK *"is an understanding that emerges from interactions among content, pedagogy, and technology knowledge"* [Koehler and Mishra, 2009] altogether; using their TPACK, teachers can apply specific technologies, considering specific pedagogical frameworks when teaching an equally specific content domain.

After its conception, TPACK gained lot of attention and that might result from the fact *"the notion of a unifying conceptual framework was lacking in the educational technology literature"*; actually, TPACK added value to the discussion *"especially when conceptualizing how the affordances of technology might be leveraged to improve teaching and learning"* [Archambault and Barnett, 2010].

However, there are important criticism on TPACK, since it does not provide an actionable body of knowledge, as a theory or framework should. Graham [2011] rises relevant questions from the fact TPACK is built on the Pedagogical Content Knowledge theory, a *"theoretical framework that lacks theoretical clarity"*. Like its base framework, TPACK *"is easy to understand at a surface conceptual level"*, since it is easy to advocate *"the importance of integrating knowledge domains related to pedagogy, subject matter, and technology"*. However, that simplicity *"hides a deep underlying level of complexity, in part because all of the constructs being integrated are broad and ill-defined"*.

Another criticism towards the TPACK basis model is regarding the impossibility of clearly separating the knowledge domains; Archambault and Barnett [2010] research indicates measuring each of TPACK domains is complicated and even impossible, perhaps due to the fact they cannot be even separated. Therefore, the model *"provides limited benefit to administrators, teachers, and most importantly, students"*.

Anyway, TPACK has been a useful inspiration for other works, which indeed offered more practical results. Koh and Divaharan [2011] suggested the **TPACK-Developing Instructional Model**, recognizing teachers' instruction on Information and Communication Technology (ICT) tools implies building on TPACK domain, and that requires not only technology instruction, but changes on teachers' attitudes and motivations. That model *"prescribes three instructional phases for developing teachers' TPACK as they learn to use new and unfamiliar ICT tools"*. The *phase 1* is when the tool acceptance should be fostered, with the teachers understanding and accepting the pedagogical benefit. In *phase 2* teachers deepen their technological proficiency and pedagogical modeling, working with exemplary materials of their content domain. Finally, in *phase 3* teachers explore and consolidate the ability to apply the tool in their classes. Figure 6 depicts TPACK-Developing Instructional Model phases.

Urban-Woldron research, published within the European Science Education Research Association (ESERA) effort [Urban-Woldron, 2011], indicates proper training materials can have a positive impact on the TPACK for prospective physics teachers. That research also indicates the training effectiveness has a direct relation to the prospective teachers motivational orientation — characteristics like goal orientation (*"an integrated pattern of motivational beliefs that is represented by different ways of approaching, en-*

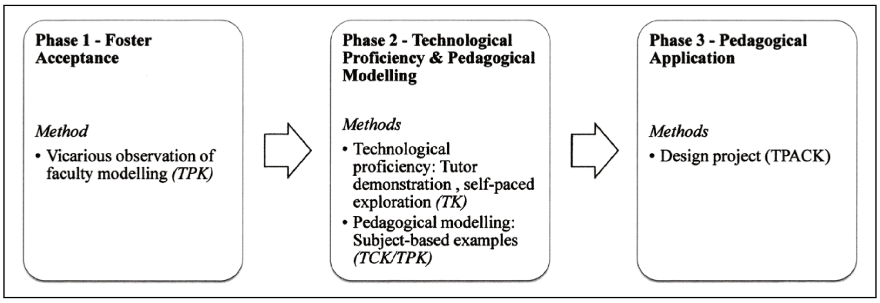

Figure 6 – TPACK-Developing Instructional Model, reproduced from Koh and Divaharan [2011].

gaging in, and responding to achievement activities" [Neuville et al., 2007]), content task value (*"students' perceptions of the interest, usefulness, importance and cost of a task"* [Neuville et al., 2007]) and self-efficacy (*"the student's belief that he or she can successfully perform a task"* [Neuville et al., 2007]). Hence, the training needs also to focus on motivating the teachers to use the technology on their specific content topics; teachers need to know how they can transform their pedagogical strategies using technologies, since the best way that can be achieved strongly varies according to the content being covered. Teachers have a critical role as curriculum designers, taking technology, pedagogy, and content together [Urban-Woldron, 2011].

The Teaching Teachers for the Future (TTF) project was created to promote the ICT usage among Australian higher education institutions, through the focus on ICT for teacher formation, curriculum development, and national support network creation. TTF has adopted TPACK concepts and developed the **TTF TPACK Survey** instrument to create an evaluation tool for the teacher's abilities. Jamieson-Proctor et al. [2013] present the process of creating the evaluation instrument, which was focused on the TPACK components directly considering the technology influence: TPK, TCK, and TPACK itself. As a self-report instrument, TTF TPACK Survey explored the pre-service teachers *"perceived level of confidence with ICT, as well as their perceived level of usefulness of ICT to undertake the task described by each item"*. Table 4 reproduces the questions applied by the TTF TPACK Survey, specially created to measure the TPACK knowledge component.

We considered the TPACK model inspiring for our work, as it suggests the importance of acting on different knowledge areas when aiming effective technology usage on pedagogical activities. In fact, our overall proposal for achieving greater acceptance of our tool is to foster both the technology and the methodology usage.

2.5 No clearly actionable directives for effective delivery

Teaching methodologies offering learning gains over traditional lecture have long been available; educational technologies are continuously developed, most of the time trying to facilitate the usage of those methodologies. However, the gap from their conception

How *confident* are you that you have the knowledge, skills and abilities to support students' use of ICT to... **How *useful* do you consider it will be for you, as a teacher, to ensure your students use ICT to...**
...provide motivation for curriculum tasks
...develop functional competencies in a specified curriculum area
...actively construct knowledge that integrates curriculum areas
...actively construct their own knowledge in collaboration with their peers and others
...analyze their knowledge
...synthesize their knowledge
...demonstrate what they have learned
...acquire the knowledge, skills, abilities and attitudes to deal with on-going technological change
...integrate different media to create appropriate products
...develop deep understanding about a topic of interest relevant to the curriculum area/s being studied
...support elements of the learning process
...develop understanding of the world
...plan and/or manage curriculum projects
...engage in sustained involvement with curriculum activities
...undertake formative and/or summative assessment
...engage in independent learning through access to education at a time, place and pace of their own choosing
...gain intercultural understanding
...acquire awareness of the global implications of ICT-based technologies on society
...understand and participate in the changing knowledge economy
...critically evaluate their own and society's values
...facilitate the integration of curriculum areas to construct multidisciplinary knowledge
...critically interpret and evaluate the worth of ICT-based content for specific subjects
...gather information and communicate with a known audience

Table 4 – TTF TPACK Survey questions, focused on directly measuring the TPACK knowledge component; exploring the teachers' self-reported confidence and judgment, the answers should varies from 0 (*"Not confident/useful"*) until 7 (*"Extremely confident/useful"*). From Jamieson-Proctor et al. [2013].

until their extensive usage is still not closed. The studies we analyzed propose models and frameworks providing clever, innovative and even inspiring interpretations for the different forces and restrictions preventing the adoption of new teaching methods and technologies inside the classroom. Those studies cover a broad spectrum of causes, including any infrastructural, processual or political issues, and also the psychological aspects preventing the people involved in the teaching activities — especially the teachers — to adopt new methodologies and tools.

Although some of those studies have strong conceptual background — sometimes from different knowledge areas — all of them lack experimental proofs, as well as an actionable set of guidelines on how to promote the effective deployment of a new technological pedagogical tool. Some of the works excel on providing tools to analyze and diagnose the deployment scenario, but they fail on the next step of offering clear guidance on how to tackle the difficulties found, an effort always done through a case-by-case analysis, due to the multitude of possibilities. Other works do inspire that final task of pursuing the effective delivery of a new technology for education, not only providing scattered interpretations of the factors influencing its adoption, but also missing a clear analytical guidance.

We based our work on extensive literature, including considerations about CRS pedagogical effectiveness, Peer Instruction methodology usage and results, adoption barriers and technological adequacy to usage scenarios, as well as psychological aspects of the people involved — most crucially the teachers and instructors. That combination allowed us to build the notion that achieving social impact through the research of a technological pedagogical tool is a multidisciplinary effort.

3 Paperclickers solution

In this chapter, we describe the paperclickers solution, an affordable classroom response system, created to foster active learning methodologies, and the work we developed to improve its overall usability. We briefly present in Section 3.1 the tool functionality and its initial development, from previous research. In Section 3.2, we describe the compilation and analysis, of the available user experiments results — also performed during the initial paperclickers research. Then, we describe in Section 3.3 the corresponding changes in the application flow, focused on reducing the adoption barriers resulting from the new technology. We close this chapter in Section 3.4, indicating the need for future user experiments to evaluate the released version with our target audience.

3.1 Paperclickers — an affordable CRS

Our work builds on paperclickers, a previously designed solution, described by several other studies of our research team: Bindá [2015], Tejada [2014], Ribeiro et al. [2015], and Neto [2015]. Paperclickers creation was focused at lowering the cost of operating CRS, and thus increasing its adoption. It employs a mobile device to film the classroom and image processing to capture the students' answers, appearing as printed cards with *fiducial markers* optimized for fast and reliable detection. The students answer multiple-choice questions by rotating their cards into one of four orientations.

Figure 7 represents a typical paperclickers usage scenario: to use paperclickers, teachers have to distribute the answering cards to the students — typically, the students will keep their answering cards to use whenever requested. Teachers will ask a multiple choice question, up to 4 different answer options. The students choose their answer and show them using their answering card: it has a unique ID coded in its front, and the proper answer option is indicated by rotating the card up to 4 different orientations: 0°, 90°, 180° or 270° rotation. On the back of the card, there are reference indications for the correct orientation corresponding to each answer option (see Figure 8). Once the students have chosen their answers, they should hold the cards ensuring the teacher can have a direct line of sight to them; the teacher then uses the paperclickers application on her mobile device to collect the given answers. Having captured all the answers, the teacher can verify each one of them using the detailed answers screen. Figure 19 shows the prototype version of the detailed answers screen, where the teachers had the opportunity to manually change any answer — for instance, to include a missing response —; Figure 20 shows the definitive screen version (see Section 3.3 for further details). On the final chart screen (Figure 22), the teacher can check the overall class performance. The paperclickers application usage flow is presented in Figure 9.

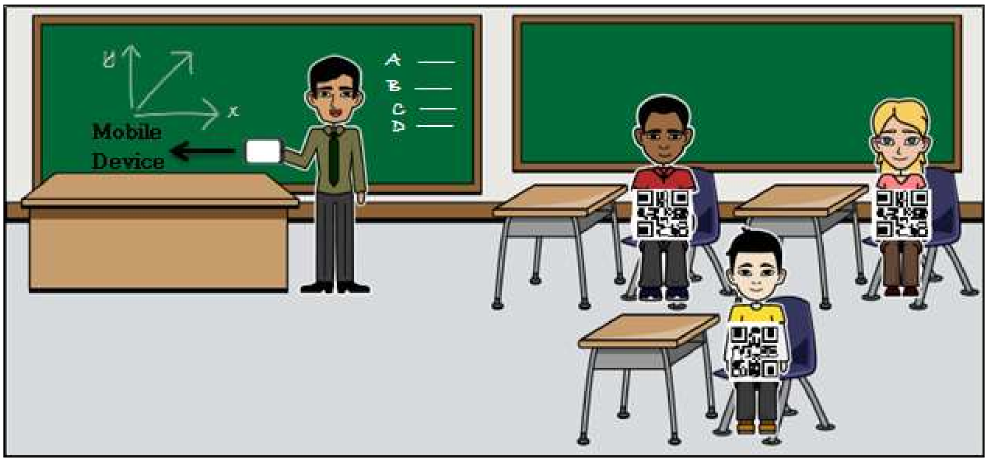

Figure 7 – Basic usage scenario defined during paperclickers initial design (reproduced from Bindá [2015]). Note that the students' answering cards are still represented by QR Codes, which were later replaced by TopCodes.

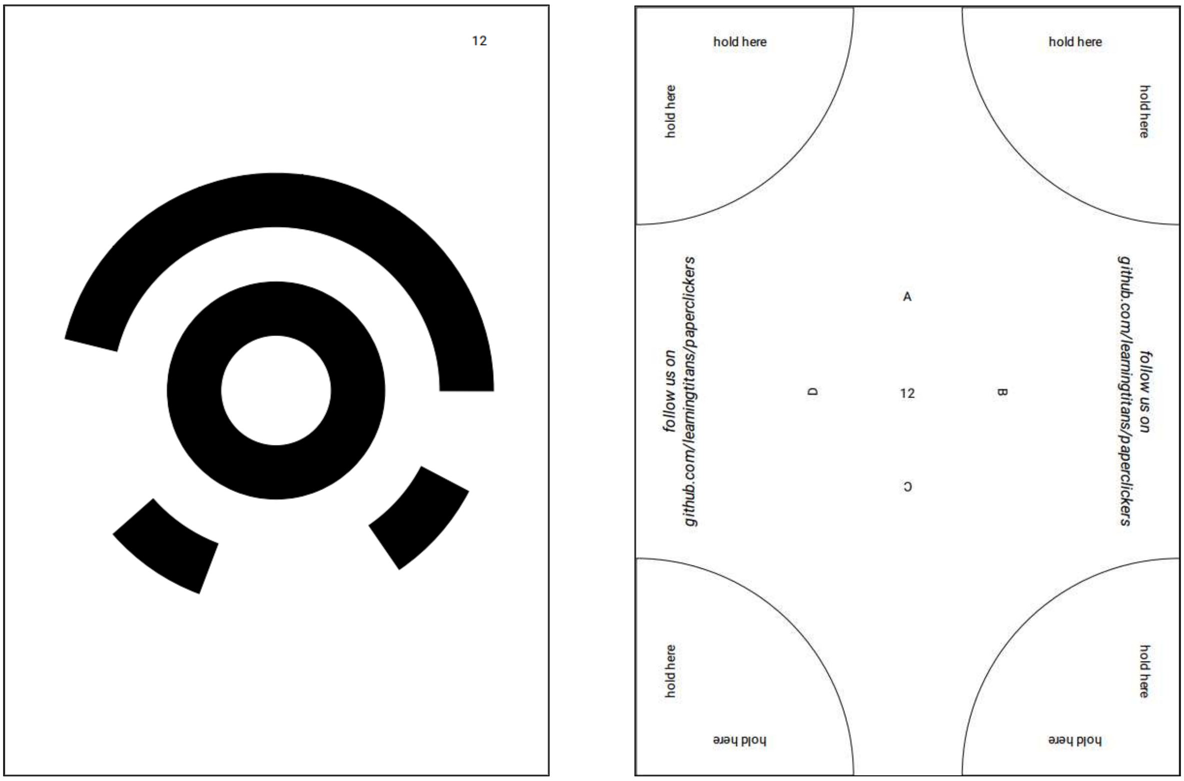

Figure 8 – Example of paperclickers answering card showing the TopCode encoding (code 12) on the front and the answer options rotation reference on the back. To select the proper answer option, the student should hold the answering card front facing the teacher, and rotate the card until the answer option letter appears in the correct orientation for her to read.

As an image processing solution, paperclickers has reduced requirements: a single device per classroom (which can be the personal device the teacher owns) and no Internet connection. The answering cards must be printed and distributed to the students, but the cost may be as low as a few cents per card. When compared to the existing CRS, our

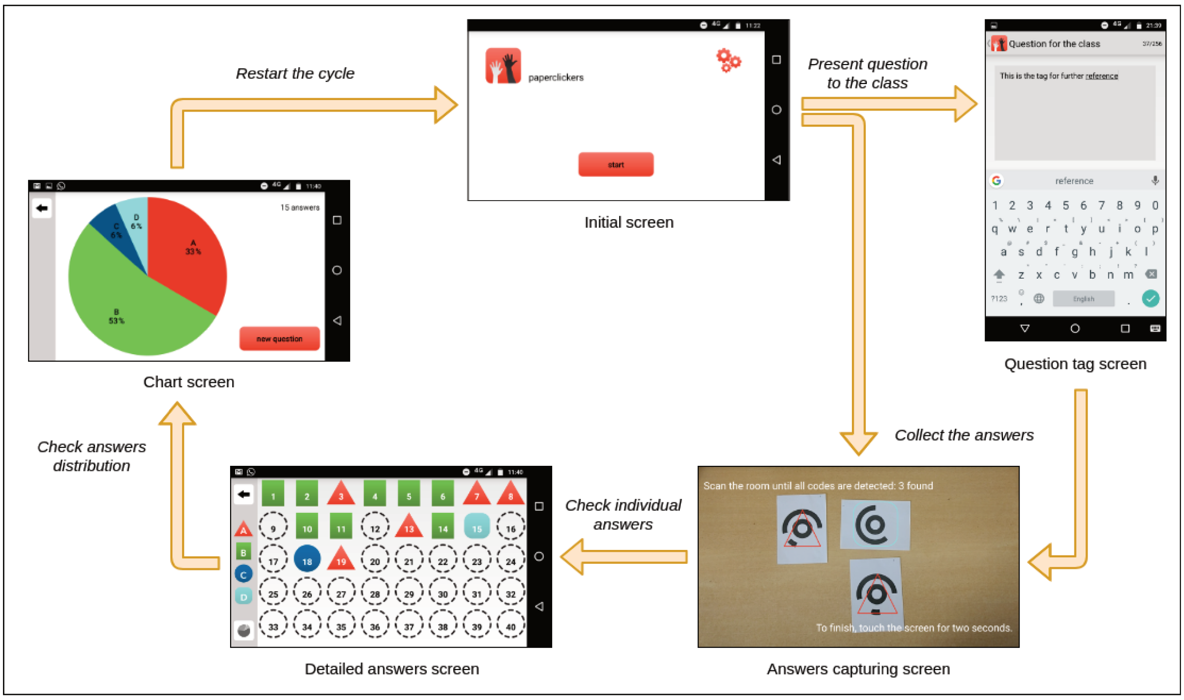

Figure 9 – Paperclickers mobile application usage flow; the question tag entering screen is optional, disabled by default.

solution requires little setup, and it has simplified usage, reducing costs of installation and training.

Paperclickers concept is similar to Cross et al. [2012], but employing a smartphone to capture students' answers, instead of a fixed camera and a desktop computer. Instructors can use their own smartphone to film the class, as the students hold up cards with their answers, using four different orientations to pick among four possible answer choices. After scanning the entire classroom, the instructors can view all students' answers in detail or summarized through a pie chart.

Paperclickers prototype design and development was guided by storyboards, which proved to be an effective tool for such product size: the design started with brainstorming sessions to storyboard the application use cases, its interface, and its behavior. The storyboards were then the primary planning tool for the development — they provided a good compromise between our desire for an informal, lean process, and the need to design the application, document and communicate the decisions among the team.

The choice of machine encoding technique to be employed in paperclickers was a central question, since it defined critical capabilities: the solution was targeting enough encoding power in a robust code, which should to be recognized in real-time. Paperclickers original researchers evaluated QR Codes and TopCodes [Horn, 2012] as machine encoding techniques, considering the usage of a handheld device would impact both the image capturing (user direct camera manipulation) and image processing (reduced computational power) steps. Through experimental analysis, the two techniques presented opposite strong points of encoding power (QR Codes), contrasting with robustness and decoding speed (TopCodes).

The final prototype employed TopCodes since its original use case was compatible to paperclickers, specifically related to the detection robustness and speed: TopCodes solution was designed to recognize tangible objects on a camera-orthogonal surface — the Tern[1] tangible programming environment [Horn and Jacob, 2007] — providing quick and robust detection. TopCodes also provides a unique ID and orientation, but it can recognize only 99 codes.

Paperclickers initial research performed usability tests to verify the user interface and experience; the user experiments results indicated the proposed solution accomplished the initial goals, but it required some rework before creating a release candidate version; the work done by the original research had not incorporated those changes on paperclickers. In the next section, we present the details on the user experiments setup and their results compilation, performed to support the solution changes designed by the present research.

3.2 User experiments compilation and analysis

We started the experimental analysis revisiting all the data gathered through the user experiments, performed by Bindá [2015] original paperclickers work, and with following main characteristics:

- A sample size of N = 11, selected from volunteer participants of at least 18-year old, from graduate and undergraduate students of the research team institute; some of them had previous teaching experience.
- The test experiment procedure considered only one participant in the teacher role, using the application to collect the answers; answering cards fixed on the backrests of classroom seats, simulating the students answering during a class.
- Tests were based on two predefined scripts: performing the class roll-call (task 1) and performing a question poll (task 2). The testers were asked to follow scripts, indicating each activity they had to execute using paperclickers and in which sequence — further details can be found in Bindá [2015] and Oliveira et al. [2017].
- Employing methods of direct observation, user and in-device interaction recording, semi-structured interviews, and questionnaire after the execution of the scripts.

[1] Tern - http://hci.cs.tufts.edu/tern

Our main data source was the in-device interaction recording, performed using a commercial tool[2]. That was able to save all the user interaction over the paperclickers application, including verbal comments. Through the captured videos we could follow the users' execution of the testing scenario, identifying where they touched the interface, their interaction pace and also their verbal expressions and comments. All that information allowed a good understanding of the users' performance over the application.

The analysis of the semi-structured interviews provided additional information about the users' opinions and perceptions about the paperclickers application experience; Table 5 combine the findings of both data sources.

Finally, we analyzed the questionnaires, compiling the information on Table 6, and looking for patterns on the users' opinions about interface items and behavior; that data was essential to provide additional qualitative information to compose and support the hypothesis created from the analysis of the interaction recording and the semi-structured interviews.

Paperclickers overall usability obtained good results, indicating the proposed solution is applicable in a classroom; however, those tests pointed two major usability issues, related to application *convenience* (roll call feature identification and initialization) and *consistency* (backward navigation throughout the application's screens):

- 3 users were unable to execute the *roll call* feature at all; 4 faced difficulties to start the *roll call* feature; 2 failed to change the user *presence/absence status*. The *roll call icon* was unclear and hard to notice as a clickable element (see figure 14, the icon on the upper-right corner); the *presence/absence students' status icons* (see figure 15) were also not noticed as clickable elements.

- 3 users found the *"Camera close"* message misleading, when they tried to go back to the scanning screen; 4 users got confused with the backward navigation, due to its inconsistent behavior. Figure 10 depicts the source of confusion: different elements trigger the backward navigation to different and unclear returning points, not behaving accordingly to the user expectation on the running platform (the Android mobile platform), which is to always return to the previous screen in the navigation path.

The information obtained from the analysis of the user experiments data was the guide for the application changes performed on this present work, described in the next section (3.3).

[2] Recordable — http://recordable.mobi/

Tester	Backward navigation	Roll call (task 1)	Answers scan (task 2)	Test script	Additional comments
1	Misleading "Camera close" message; Inconsistent behavior	Could not change student status	Did not use "New question" for next question		Might reduce the teachers/students relation
2	Inconsistent behavior	Unable to execute roll-call feature	Did not use "New question" for next question		
3		Could not start feature without help; Could not change student status	Looked for additional information on chart screen	Did not understand how to answer script question	
4		Unclear presence/absence icons	Looked for additional information on chart screen		Liked roll call feature agility
5		Problems to start roll-call feature; Detection problems	Detection problems; Looked for additional information on chart screen	Did not understand how to answer script question	Problems to dismiss about screen
6	Misleading "Camera close" message	Unclear roll-call icon; Detection problems	Detection problems; Did not use "New question" for next question		Application low speed; Missing back option in detection screens; Focused on specific usage scenarios
7	Misleading "Camera close" message; Inconsistent behavior	Problems to understand roll call feature	Detection problems		The single device requirement might not be low cost; Students having to keep big cardboard signs might be a problem
8		Problems to start roll call feature	Detection problems		Scanning large classroom could be cumbersome
9	Inconsistent behavior	Unable to execute roll call feature	Used "New question" but couldn't realize the question number auto increment		
10		Problems to start roll call feature	Detection problems	Did not understood how to answer script question	Forced landscape orientation; Found inconsistent the ability to change student presence while changing answer; Asked for more than 4 answers' choices; Question about detection in real classrooms
11		Detection problems	Detection problems		Problems to dismiss class selection list; Would be nice to have the question text; Its usage might distrait the students

Table 5 – User device interaction findings — Recording the interaction of user with the app provided the most actionable information on the usability tests test, which were positive regarding application usage, but revealed that some features and navigation were confusing to users.

Element	L/A	In	D	DU
Application forced landscape	2	3	6	
Initial screen – class selection option	8			3
2^{nd} screen – question selection option	8		1	2
2^{nd} screen – question auto increment	4	7		
2^{nd} screen – roll call separated from answers scanning	6	2	3	
2^{nd} screen – roll call icon	5		3	3
Scanning screen – understood augmented reality cardboard indications	8	3		
Scanning screen – augmented reality cardboard indications	6	5		
Scanning screen – found augmented reality feedback slow	6	5		
Scanning screen – cardboards capture finalization method		3	8	
Roll call results screen – easily understood	9	2		
Roll call results screen – presence/absence icons understanding	5	4		1
Roll call results screen – would like to have student name or picture along presence/absence icons	5	6		
Roll call results screen – easily understood presence/absence icons were clickable	5	6		
Roll call results screen – screen closing icon	5		6	
Detailed answers screen – layout	10	1		
Detailed answers screen – easily understood answers were clickable and could be changed	6	1		4
Detailed answers screen – understood "X" answer indication	7	3		1
Detailed answers screen – understood chart screen icon	11			
Detailed answers screen – "BACK" icon	8		3	
Detailed answers screen – correctly understood "back" icon would return to the scanning screen		2		9
Detailed answers screen – understood "back" icon would return to the process beginning	4	7		
Chart screen	10		1	
Chart screen – correctly understood "Try again" button would return to scan screen keeping the question	7	1		3
Chart screen – understood "Try again" button would return to detailed answers screen keeping the question	1	1		9
Chart screen – understood "Try again" button would return to the initial screen for class selection screen	2	1		8
Chart screen – correctly understood "New question" button would finalize the question and return to the question selection screen	8			3
Chart screen – understood "New question" button would return to the answers scanning screen	2			9
Chart screen – understood "New question" button would return to the detailed answers screen	1			10
Chart screen – correctly understood "back" button would finalize the question and return to the initial screen for class selection	3			8
Chart screen – understood "back" button would finalize the question and return to the question selection screen	4			7
Chart screen – understood "back" button would finalize the question and return to the answers scanning screen	1			10
Chart screen – understood "back" button would finalize the question and return to the detailed answers screen	3			8

Table 6 – User questionnaire findings compilation — paperclickers usability testing. L = Liked; A = Agree; D = Disliked; DU = Did not understand; In = Indifferent.

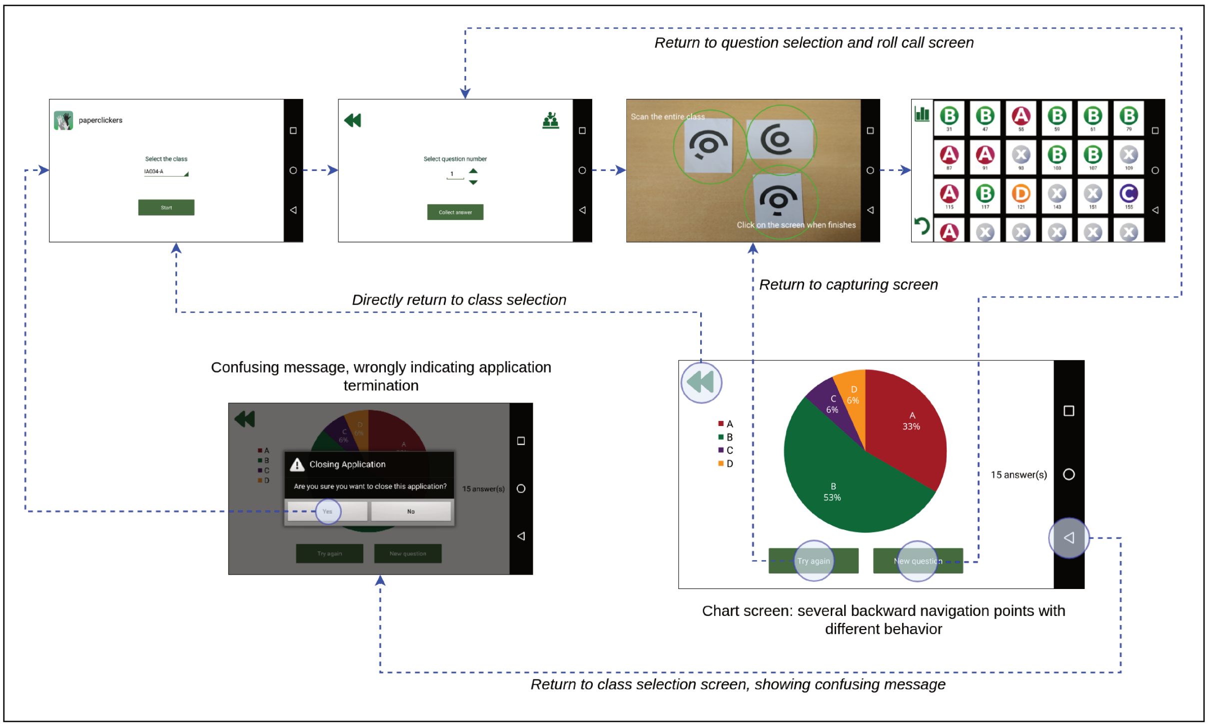

Figure 10 – Inconsistent backward navigation on paperclickers prototype: different elements caused the application to return to different points in the navigation, not respecting the mobile platform behavior of returning to the immediately previous screen.

3.3 Changes in the application flow and features

Considering the findings of the user experiments, we conducted a new development phase to design and implement changes to accommodate the required adjustments: we focused on usage flow issues, mostly related to the *roll call* feature identification and use and the overall backward navigation.

We once again relied on storyboards to redesign and guide the implementation changes (Figure 11), since that tool has proven efficient for a small development team, during the initial development. We concluded this phase with the first public release of the paperclickers solution, as an Android Platform application[3], open-sourcing its code licensed as GNU General Public License v2 (GPLv2[4]).

The research team analyzed the user experiments findings and redesigned the application flow as an attempt to minimize the issues. We applied the Minimum Viable Product approach [Ries, 2011] for paperclickers release, focusing on the essential functionality towards its main goal of providing an affordable CRS solution — the ability to collect and summarize students' responses for a given multiple-choice question. That clear goal leads the new development cycle and the work on the identified issues, resulting in a simplified application version.

3 https://play.google.com/store/apps/details?id=com.paperclickers
4 https://github.com/learningtitans/paperclickers

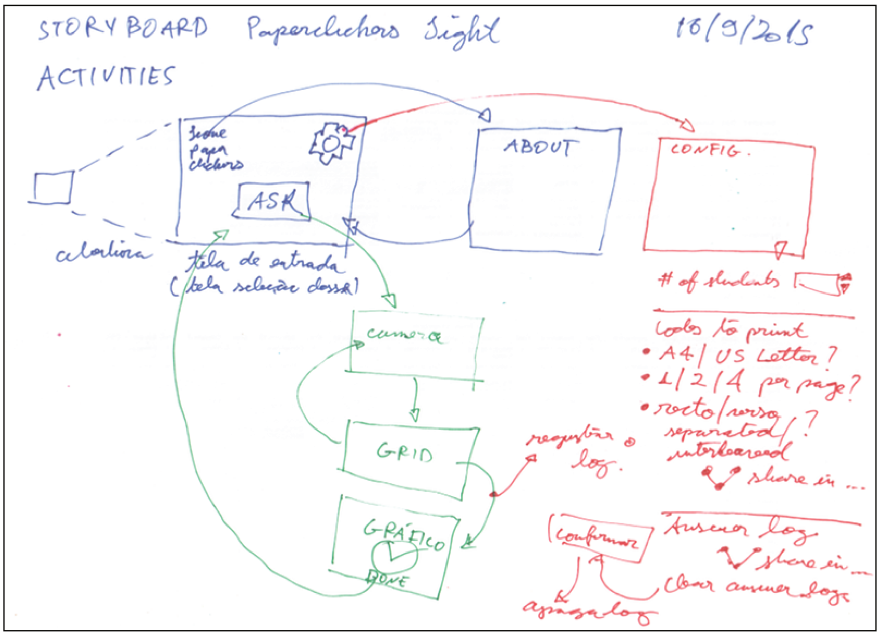

Figure 11 – Storyboard for the paperclickers second development cycle, used to define and document the new and simplified application flow.

We simplified the first screen (Figure 12) by removing the class definition. We also removed the second screen (Figure 14) entirely, along with its two features — the question definition and roll call feature initialization — since one of the detected usability problems was the confusion created by their combination in a single screen (see Section 3.2 for further details). Users commonly think the roll call feature useful — Hunsu et al. [2016] meta-analysis indicates additional tools like roll call are considered attractive aspects of CRS —; however, we preferred to remove it from the released version due to the following: the preliminary definitions and roll call required a separated module (e.g., desktop or web application) to enter classes an students identifications, and we have not considered developing yet; and according to the user experiments, the roll call feature required a major redesign. We realized the roll call feature could be practically replaced by an answers scan result, since the students' presence can be indirectly taken from their answers.

In order to simplify the flow, but still allowing offline class management support, we redesigned the answers log format, transforming it into a table, stored in a text file using the ".csv" file format, able to be easily shared and opened in regular spreadsheet software. Since we removed the question definition, we added a sequence number to automatically identify the question done at a single execution round. We included the ability to share the answers log, creating a data export channel allowing teachers to manipulate the students' answers using other tools — for instance to track class or individual history throught a spreadsheet software. We have also added a new screen (Figure 16) to allow teachers to include an optional textual tag for each question, a valuable information for further reference during those classroom management activities, once that tag is also included in the answers log.

To have a standalone solution, we included the students' answering cards generation feature, providing the ability to share Portable Document Format (PDF) files with the codes required for the students. The answering cards can be generated in different sizes — one per page, two per page and four per page — and in different page sizes — letter and A4. The detection and decoding experiments executed (Section 4.2), indicated the two per page code sizes provides the best detection and portability for a medium sized class (60 students, 10 meters longest distance).

The released paperclickers user interaction included the following main elements:

- **Settings** — encompassing the following elements:
  - **Minimal preliminary definitions**: simple class size definition, required parameter for the responses detection speed and robustness.
  - **Answers log sharing**: added functionality, included to provide the ability to further manipulate the detected answers.
  - **Students' codes printing**: added functionality, included to enhance the solution completeness.
- **Question tag definition** — added functionality, allowing an optional tag definition for each question, providing information in the answers log for further reference, facilitating offline classroom management activities.
- **Enhanced answers capture display** — added on-screen feedback on class scanning screen, providing instant feedback regarding the detected and validated[5] answers.
- **Enhanced results screen** — detailed answer screen with improved colors and design, using the fact we had to remove the roll call related icons.
- **Enhanced chart screen** — answers chart view screen with improved colors and design, in order to simplify the available options, keeping the back button consistency across all the screens and offering only a button for new question capture.
- **Enhanced about screen** — added the open source license for all used software, as well as the copyright and privacy policy information, complying with the requirement for a product release.
- Revised application backward navigation, aligned with the mobile platform expected behavior of always returning to the immediately previous screen.

[5] A TopCode validation step has been included; see Section 4.1 for details

Figures 12 until 23 compare each one of the main screens of the prototyped and released paperclickers versions, properly indicating the screens removed from the prototype (question selection and roll call initialization; roll call results) and the screens added in the released version (question tag; settings). Since we had touched several points of the paperclickers application, we used this development phase to apply a distinctive visual signature — colors and screen elements — for it, essentially following the researchers' design decisions.

We performed an informal usability inspection within the research team, after applying the changes described above: we followed the basic usability heuristics set defined by Nielsen and Molich [1990] for the evaluation by the members of the project team. Our goal was to make an initial evaluation of the new application flow, looking for very basic flaws: after the analysis we have done over the paperclickers prototype user experiments results, we realized some of the issues could have been detected by an inexpensive and quick evaluation technique like heuristic evaluation.

From that evaluation, we decided to turn off the option to manually change the students' answers, available on the **Detailed answers** screen, since some of the evaluators found it might be creating an unclear application flow — manually added answers were considered detected when the user makes additional detection attempts, and that information was not clear and totally consistent with the represented system status. Also, the overall usability was error prone to unwanted touches, once it was too easy to change a student answer unintentionally.

Finally, we included the feature of collecting application usage anonymous data, instrumenting paperclickers to track user behavior on its interface. We used the analytics framework provided by the target mobile platform[6], including a **Settings** option for the user to disable that data collection. We also defined a *Privacy Policy*, according to the platform regulations, available in the **About** screen and in a public website[7].

3.4 Usability as a continuous work

Compiling and analyzing the user experiments results, and then applying changes in the application's user interface, demonstrated the need of considering several usability evaluation and adaptation cycles, to reduce the usage barrier related to the knowledge required to operate the technology. The focus on the intended target audience of the technological pedagogical tool unveils new requirement to meet, potentially simplifying the provided functionality.

We have evaluated paperclickers with graduate and undergraduate students, and

[6] https://firebase.google.com/docs/analytics/
[7] https://sites.google.com/view/paperclickers/home

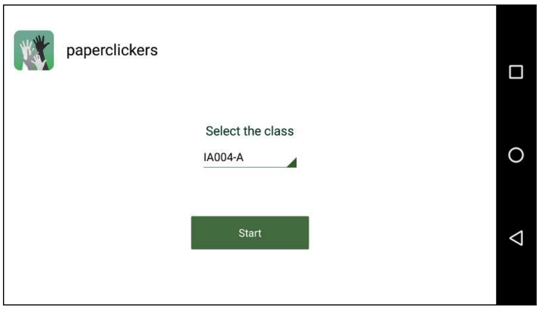

Figure 12 – Initial screen – original version

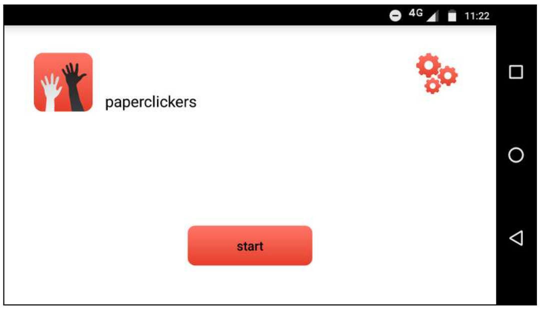

Figure 13 – Initial screen – released version

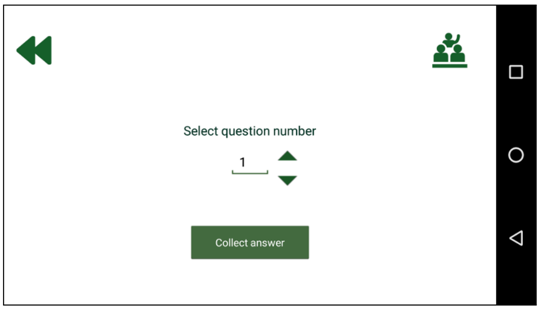

Figure 14 – Question selection screen – original version; roll call feature accessible in upper-right icon

Screen removed on released version

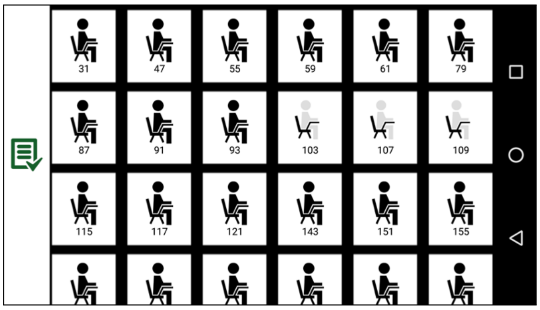

Figure 15 – Roll call result screen – original version

Screen removed on released version

Not included in original version

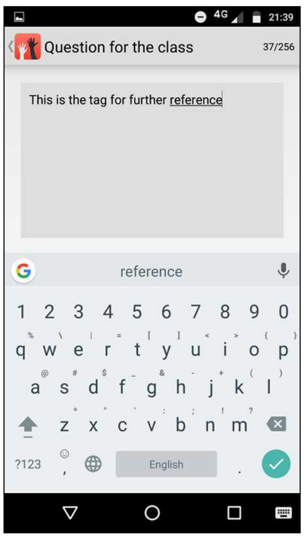

Figure 16 – Question tag screen – released version

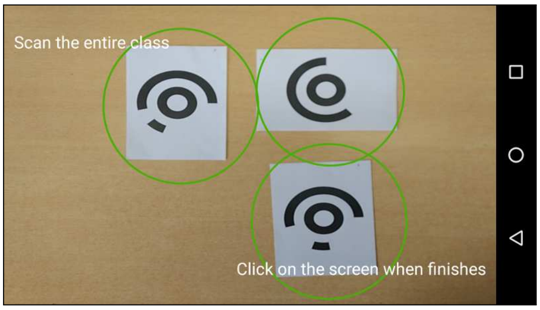

Figure 17 – Scan screen – original version

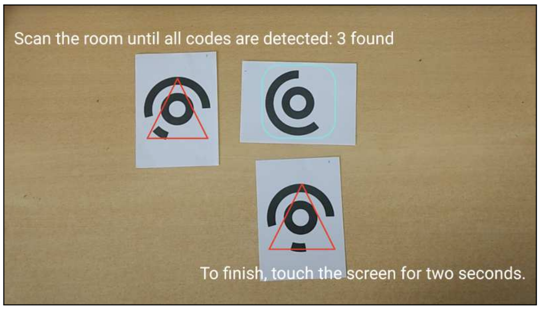

Figure 18 – Scan screen – released version

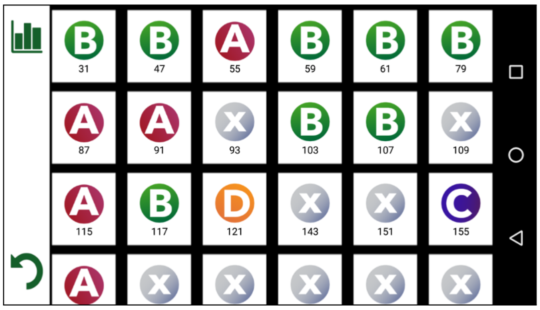

Figure 19 – Detailed answers screen – original version

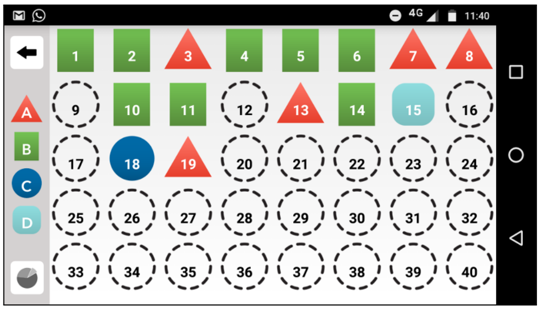

Figure 20 – Detailed answers screen – released version

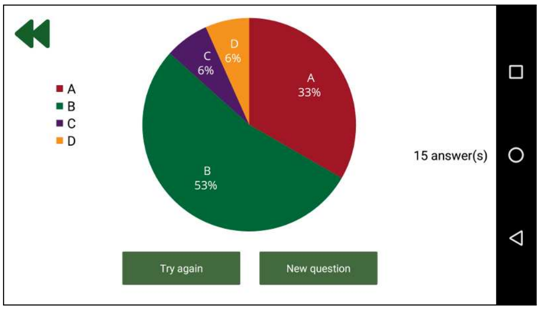

Figure 21 – Chart screen – original version

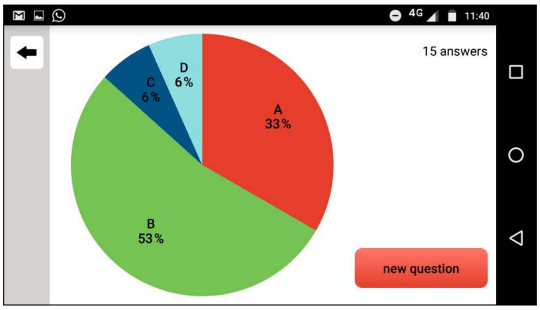

Figure 22 – Chart screen – released version

Not included in original version

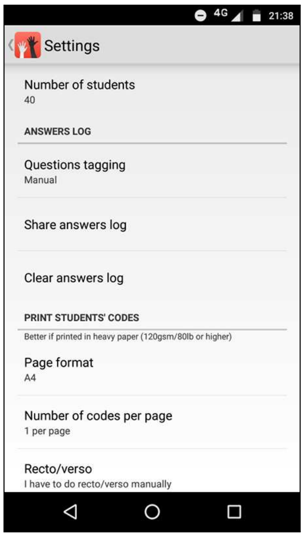

Figure 23 – Settings screen – released version

relevant findings allowed designing changes we believe would improve the overall tool usability. However, considering we have defined a different target audience for paperclickers — high school classrooms of Brazilian public schools (see Section 5.3) —, additional user experiments are needed with teachers and students to properly evaluate the reduction of the technology usage adoption barriers. That requirement presents new challenges to this work, due to the great diversity of environments and conditions.

4 Answering cards processing — Topcodes changes

In this chapter, we present the changes in the answering cards detection and decoding operations, motivated by the user experiments results and also by problematic situations identified during our development cycle. In Section 4.1, we briefly describe the TopCodes — the machine encoding technique selected for paperclickers. Then we analyze the issues, and present our solution design and implementation. In Section 4.2, we describe the experiments designed to verify the effectiveness of our solution and their results. We conclude this chapter (Section 4.3) resuming the discussion about the need for the balance between the detection robustness and the overall solution usability. We present the experiments' quantitative results to demonstrate how both the detection speed and maximum distance — two very sensitive parameters for paperclickers user experience — are affected by the approaches to improve the solution's robustness.

4.1 Changes in TopCodes detection and decoding

As presented in the previous chapter, paperclickers employed TopCodes as the machine encoding for the students' answering cards. However, using TopCodes in a scenario different from its original one created new challenges: TopCodes were designed for a reasonably static usage scenario, with a controlled, and even clean, background. Applying TopCodes to detect answering cards in the dynamic environment of a classroom, presented issues that we had to overcome, towards creating a functional image processing CRS.

Figure 24 represents the typical setup of TopCodes original usage scenario, in the Tern (the tangible programming environment) proposal. There, the overall elements and background are supposed to be static: both the source code composition, represented by commands identified through TopCodes, and the detection experience (commands interpretation) are considerably less dynamic than the paperclickers regular usage scenario proposition.

4.1.1 Errors due to partial occlusion

During this second development phase, we also worked to improve a decoding fragility identified on the TopCodes reference library implementation[1]: partially occluded

[1] http://users.eecs.northwestern.edu/ mhorn/topcodes/

Figure 24 – Typical usage of the Tern tangible programming environment, the TopCodes original use case scenario, is very different from its application as a answering card in an image processing CRS solution. Reproduced from Horn et al. [2009].

codes could be erroneously decoded, registering wrong answers, as shown in Figure 25. We also improved the overall detection speed — one of the user complains about the prototype.

As already mentioned, the reference TopCodes detection and decoding library were created for a controlled usage scenario, differing from a classroom with students showing their answers during a teacher pool. In this new scenario, the answering cards partial occlusions are a reasonably expected occurrence: the varying camera baseline and position (caused by both the teacher scanning movement and the students holding their cards), combined with the dynamic partial occlusions, make the spurious decoding possible.

In order to investigate and propose fixes for those issues, we have further investigated the TopCodes implementation. Figure 26 depicts the basic TopCodes structure:

- TopCodes are composed of a bulls-eye marker, which is its center closed circumference.
- The outer circumference is where the data is encoded, using 13 sectors (arcs) of approximately 27.69° each, defined counterclockwise.
- Each TopCode is composed of 8 units, counted as regular segments of its diameter, defined by each one of the concentric circumferences; sections 1 and 8 corresponds to the data ring; sections 2, 4, 5, and 7 are always **white**; sections 3 and 6 are always **black**, since they composed the bulls-eye marker.

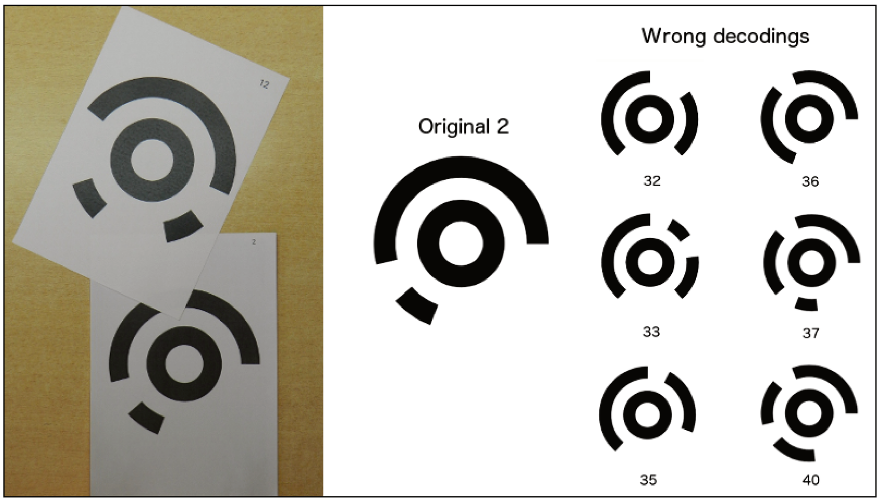

Figure 25 – Decoding error due partial occlusion; on the right spurious detected codes for TopCode 2 partially occluded; in exactly 123 scan cycles, TopCode 32 appeared in 0.82% of the scan cycles, 33 in 0.82%, 35 in 17.89%, 36 in 8.94%, 37 in 3.25% and 40 in 8.13%.

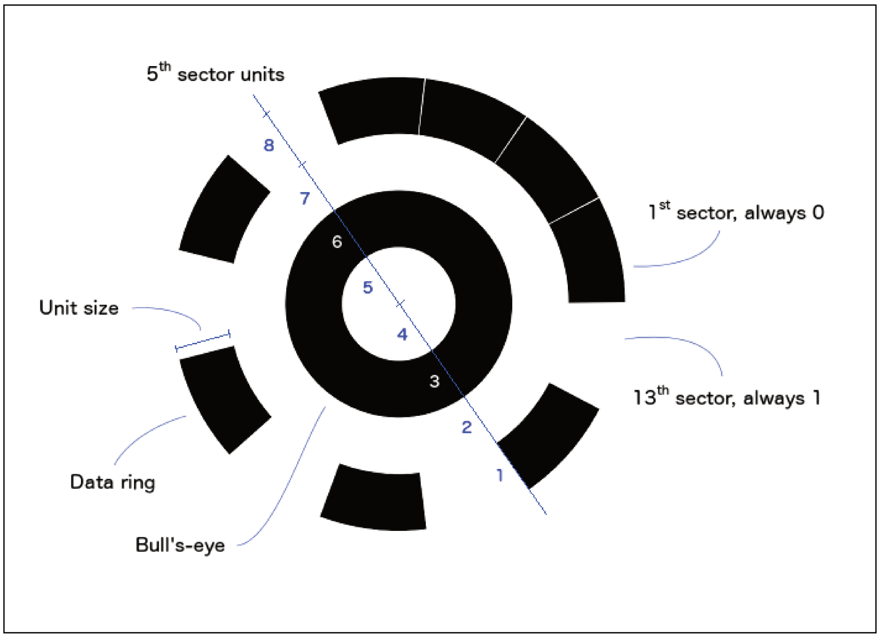

Figure 26 – TopCodes structure: the TopCode's units — represented by the blue line — are a central element for the code detection; note that a sector middle position (5^{th}) is exactly opposed two other sectors limit (11^{st} and 12^{nd}).

- Encoding considers only 5 bits 1 (one), distributed among the 13 sectors; bits 1 (one) are encoded as **white** sectors, leaving 61.5% of the outer ring **black**.
- For TopCodes orientation detection, the first sector is fixed in 0 (zero), **black**, and the last sector is fixed 1 (one), **white**; the first sector is the Most Significant Bit.

The TopCodes processing algorithm is presented below: the original operations to detect and decode the TopCodes are detailed, as well as the included changes are further discussed in the sequence:

First loop through the image data — horizontal scan to mark candidates

1. Apply adaptive thresholding [Wellner, 1993] over the pixels to binarize the image.
2. Keep track of horizontal bit sequence, looking for the TopCode bulls-eye (BE): a sequence of **black/white/black** pixels of proportional length — **black** sequences of $n > 2$ pixels, **white** sequence of $2 \times n$ pixels.
3. Mark as TopCode candidate the middle of identified BE sequences.

(1st change point) Added loop through the image data — vertical scan to mark candidates

4. *Keep track of vertical bit sequence, looking for the TopCode bulls-eye (BE): a sequence of **black/white/black** pixels of proportional length — **black** sequences of $n > 2$ pixels, **white** sequence of $2 \times n$ pixels.*
5. *Mark as TopCode candidate the middle of identified BE sequences.*

Third loop through the image data — candidates decoding attempt

6. Go through the image until finding each TopCode candidate, previously marked.
7. Determine the TopCode diameter and unit size; **irregular units void the candidate**.
8. Test 5 unit variants (90%, 95%, 100%, 105%, 110%) and for each one of them, test also 10 angle adjustments (adding from 0 until 90% of sector size) — a total of 50 different combinations — computing the TopCode decoding confidence as follows:

   a) For each one of the 13 sectors, compute the **color confidence** of each one of the 8 units, averaging the colors from a 3×3 pixels samples from the diameter crossing the middle of the sector; **wrong unit colors void the candidate**.

      The **color confidence computation** considers **white** units should have higher values (near 255) and **black** units should have lower values (near 0). Since the encoded bit is taken from 8th unit, it also has higher confidence the closer its value is from **white** or **black** (near 255 or 0 respectively); 1st unit has higher confidence for uncertain values (near 128), since the diameter should cross exactly at two sectors limit — see Figure 26.

9. TopCode encoded value is the bit sequence of the variant with the highest reading confidence — the sum of the sectors color confidences.

10. Define the TopCode rotation, by rotating the bit sequence to the left until the last bit 1 has jumped from the 13th position to the 1st — that uses the fact the Less Significant Bit is always 1.

(2nd change point) Added TopCode validation — time consistency

11. *Keep track of the decoded TopCode value, registering how many successive image frames it has been seen.*

12. *Only if the same decoded TopCode value has been verified during the last N consecutive image frames, mark the decoded TopCoded value as valid.*

To overcome the detection and decoding error, we decided to create an additional validation phase, after the TopCodes decoding (2nd change point above) and before registering a given answer: any code should be detected across subsequent scan cycles for a certain number of times; only after this arbitrary *validation threshold*[2], the code is declared valid and the corresponding answer registered. That approach considers the fact the spurious decoding fluctuates and is intermittent throughout the reading cycles, due to the dynamic nature of paperclickers scenario.

Spurious detection could still be reproduced if a fixed threshold is applied and the scan period is arbitrarily long. To reduce the spurious detection probability during such long scans, we applied an increasing threshold[3], considering the spurious decoding probability also increases with longer exposures; we called this approach the *time-consistency* verification, since the verification threshold proportionally increases with the total scanning time.

4.1.2 Dealing with too many code candidates

We realized the overall detection and decoding cycle time presented huge changes depending on the image background. The analysis indicated the detection phase was marking a vast amount of TopCode candidates if the background presented vertical lines pattern, due to the horizontal scan used to search the image for black/white sequences to mark the TopCodes bulls-eye candidates. To reduce that sensitiveness to the background, we included an additional vertical scan step in the detection process (1st change point above), looking for the same black/white sequences also in that direction. Since the bulls-eye are a complete circle, TopCodes candidates would only be points found on both the horizontal and the vertical scans. That approach drastically reduced the TopCodes

[2] Applied initial threshold is 3 cycles
[3] The initial threshold starts increasing after 32 cycles, and it then increments every 32/3 cycles

candidates after the first image scan phase, keeping the detection and decoding process execution time less variable. Even for regular scanning scenarios, the smaller number of TopCodes candidates reduced the execution time, compensating the vertical scan addition. Check section 4.1.4 for further details on the performance gains.

4.1.3 Sensitivity to hairline code effects

Although TopCodes are extremely robust to affine transformations (scale, rotations, moderate camera baseline changes, etc.), we found them very sensitive to hairline defects, i.e., situations where a single row or column of the code becomes entirely white or black after binarization. We found those defects would be very common if the codes were printed in less-than-perfect printers, or if the students ignored the admonition to not fold the cards. After considering several complex solutions, we attempted using morphological operations to seal those small gaps. We tested many alternatives, but a binary closing followed by a binary opening using a 3×3 pixels square element offered the best compromise between eliminating defects and preserving details. However, further tests showed that the best solution — both regarding precision and speed — was simply to instruct the user to not film from too close, as the final image resolution would remove small imperfections. See section 4.2 for details about the performed detection experiments.

Indeed, the ability to recognize a given code is predictable, given the camera parameters; equations 4.1 and 4.2 define the horizontal field of view and the final image resolution.

$$\text{HFOV} = \text{distance} \times \frac{\text{width}}{\text{focal}} \tag{4.1}$$

$$\text{resolution} = \frac{\text{pixels}}{\text{HFOV}} \tag{4.2}$$

Where:

- HFOV = Horizontal Field of View (meters)
- distance = camera distance (meters)
- width = camera chip width (meters)
- focal = focal length (meters)
- resolution = final digital image resolution (pixels/meter)
- pixels = horizontal pixels count (pixels)

Camera distance (meters)	HFOV (meters)	Sensor res. (pix/meter)	Image res. (pix/meter)	A4 unit (pix)	A5 unit (pix)
1	1.27	3615	1004	12	17
2	2.55	1808	502	6	9
3	3.82	1205	335	4	6
8	10.20	452	126	2	2
10	12.75	362	100	1	2

Table 7 – Depending on the camera distance, each pixel in the image cover longer portions of the real image, which limits the maximum distance for detection. Using morphological operations reduces even more that distance. The data has been captured using a camera sensor width of 1.3 μm, focal length of 4.7 mm; the analyzed image was 1280×720 pixels.

According to Table 7, morphological square elements of 5×5 pixels cannot detect A4 sized answering cards from a 3 meters distance, since the TopCodes unit size [Horn, 2012] is 4 pixels. As verified by our experiments (see section 4.2), after 2 meters distance the TopCodes detection starts to fail, once the morphological close operation ends up joining the TopCode's bulls-eye and data rings. Using a 3×3 pixels square element allows successfully decoding TopCodes with a hairline effect of 1.5 mm thickness.

4.1.4 Performance improvements and considerations

After all the changes, we improved the overall detection and decoding cycle execution time; Table 8 presents performance measurements, taken from different combinations of experimented features. Even though we included two steps in the overall process — the vertical scan and the time-consistency verification — we ended up reducing in around 64% the frame processing time, when compared to the original paperclickers prototype. That improved the usage in one aspect the users have complained — application slowness.

To guarantee the performance gain, we fine-tuned the grayscale conversion — one step of the Adaptive Thresholding —, and adapted the processing to use the Android's native image processing multi-core CPUs and GPUs usage[4]. After those changes, the TopCodes detection and decoding functionality reached the scan cycle performance of about 5 frames per second, running on 2017 mid-tier Android devices[5].

As verified through the detection experiments analysis, higher camera resolution is an effective way to increase the maximum detection distance, which is an important constraint in the solution usability, since it would allow using paperclickers in bigger rooms and audiences, as well as applying robustness detection measures, like the morphological

[4] Android Renderscript computation framework – https://developer.android.com/guide/topics/renderscript
[5] 1.5 GHz Cortex-A53 CPU, 1280×720 pixels image

Implementation variant	frame processing time (ms)	comparison (%)
original version (before changes)	492	
new version (released) (+) time consistency (+) vertical scan (+) renderscript (–) morphological operations	174	-64.63[a]
delta test version (+) morphological operations	232	+33.33[b]
delta test version (–) renderscript (+) morphological operations	827	+375.29
delta test version (–) renderscript	166	-4.60
delta test version (–) time consistency	191	9.77
delta test version (–) time consistency (–) renderscript	156	-10.34
delta test version (–) vertical scan	249	+43.10
delta test version (+) median filter[c]	526	+202.30

[a] Reduction when compared to the original version performance
[b] All delta test versions are compared against the new version baseline
[c] See next section (4.2) for further information

Table 8 – Performance comparison for several paperclickers variants: the frame processing time reduction improved more than 2.5 times the frame rate, even considering the inclusion of additional processing steps, the time-consistency and the vertical scan. The execution times are the average of 60 consecutive frames processing, in the same test setup — single TopCode (code 1), captured from a ~0.8 meters distance.

operations. However, during the experiments, we concluded the currently achieved performance admits a good balance between speed and detection quality using HD images (1280 × 720 pixels); also, most of the common mobile phones currently available delivers a good camera preview performance for resolutions up to HD.

4.2 Detection and decoding experiments

Due to the detection and decoding issues found, we proposed a new experiments cycle: the designed fixes needed to be validated, and the resulting overall performance should also be evaluated, especially related to the maximum distance coverage. According to their nature, the morphological operations — proposed to the overcome the hairline effects — reduce the final image resolution used for TopCodes detection and decoding, when pixels within the morphological element are considered enabled or disabled by the central pixel status.

To evaluate the changes, we considered varying (including or not) the following characteristics for the tests:

- **Time-consistency**: as explained in Section 4.1, this was the approach for reducing the spurious decoding, resulting from the TopCodes partial occlusion.

- **Morphological operation using** 3×3 **and** 5×5 **elements**: one of the approaches for dealing with the hairline effects; we needed to fine tune the morphological element size, since it directly affects the overall execution time; we also suspected this operation would reduce the maximum detection distance. Those two sizes were selected based on quick tests which determined 7×7 element would result in prohibitive execution time.

- **Median filter using** 5×5 **element**: the second approach considered for dealing with the hairline effects. Although this approach was dropped due to its efficiency, when compared to the morphological operations — in Table 8 this difference can be noticed —, we included this parameter only for the maximum detection distance experiments, since it would be worthwhile to measure its effect on that relevant user experience factor.

- **Image resolution — HD and Full HD**: the final characteristic we considered was the image resolution used for the TopCodes detection. We used HD images (1280×720) for most of the test scenarios, including Full HD images (1920×1080) only for the detection distance scenario — we had the hypothesis higher resolution would increase the overall detection/decoding power. However, by the time of the experiments, HD resolution images were the most common for the available mid-range smartphones and, as previously discussed in Section 4.1.4, increasing the resolution also increases the processing time, affecting the overall experience.

The combination of those variables was defined for each test scenario according to the researchers' criteria, essentially based on what seemed reasonable for each one of them. We present the applied rationale along with the executed tests.

Our main hypothesis, to be verified by those experiments, were the following:

1. The **time-consistency** mechanism would eliminate the wrong TopCodes decoding due to partial occlusion scenario.

2. The hairline effects can be successfully avoided using either approaches — morphological operations or median filter —; however, they will reduce the maximum detection distance.

4.2.1 Detection and decoding robustness evaluation

We have introduced two changes related to improving the TopCode detection and decoding robustness: the first one is related to the partial occlusion, and the other is related to the codes corruption by hairline effects. We designed tests to verify the results of applying those changes.

4.2.1.1 Partial occlusion

We considered three basic scenarios for the partial occlusion verification:

1. **Occlude the code with fingers**: a common occlusion scenario, since a student holding the answering card, might not respect the holding areas.
2. **Occlude the code with another code — white (light) element**: overlapping answering cards might also be a typical scenario in a classroom; our intention here is to include an occlusion by a white element, one of the TopCode's composing colors.
3. **Occlude the code with a black (dark) element**: we included this final test scenario to evaluate the partial occlusion effects by a black element, the other TopCode's composing color.

For the occlusion, we considered a static A5-sized code in the background and a moving A6-sized code in the foreground. We fixed the static code in a student desk, and the moving code was manipulated by the researcher. The measurements considered only HD resolution images. That setup was selected due to the easiness of manipulation: the small-sized TopCode is easy to be manipulated using one hand. We also tried to remain near to what we believed would be a common classroom setup — we included a clear recommendation for the paperclickers users to prefer the A5-sized TopCodes, since they offer a good compromise between easiness of manipulation and detection performance. Figure 27 illustrates this test setup.

We created a dynamic test environment, where the occlusions happened for a reasonably short period. We considered it would emulate the classroom environment, where regular students holding the answering cards would not remain in a static position, as well as the teacher, who would also be moving while scanning the classroom for the answers.

4.2.1.2 Codes corruption

For code corruption analysis, our first goal was to verify the changes aiming the hairline effects correction. We also added other scenarios, especially considering we would like to evaluate the overall detection anddecoding robustness in an environment like a classroom. Here is the complete list of codes corruption scenarios analyzed:

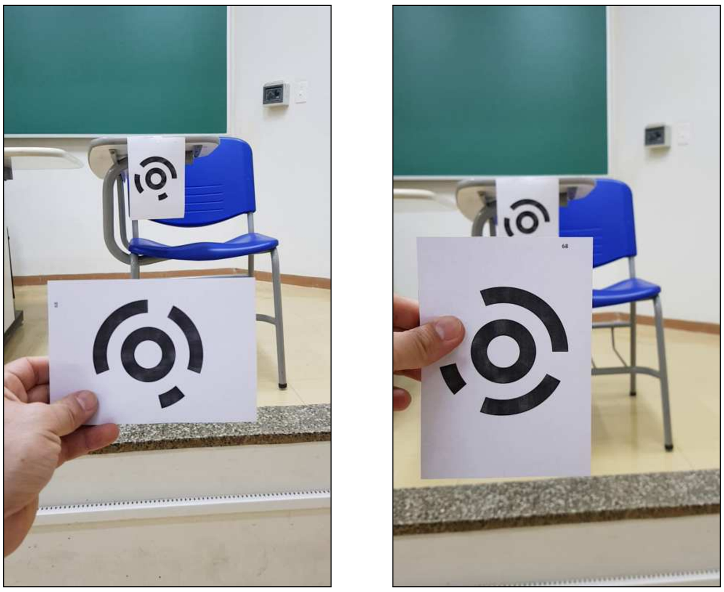

Figure 27 – The setup for the partial occlusion experiments considered two different TopCodes sizes: A5 fixed in the chair's backrest, A6 manipulated to cause the occlusion. The image at the right presents a moment with two partial occlusions: the finger on the foremost TopCode, which is partially covering the lower part of the TopCode on the chair.

- **Folded and creased answers cards**: a common scenario linked to the answers cards care, which creates hairline effects due to the thresholding image preprocessing operation.
- **Answering cards with white and black traces of different widths and quantities**: with this scenario we intended to cover common printing problems, due to a dirty cylinder or weak toner.
- **Answering cards with writings — pen and pencil — over it**: once again a common scenario related to the cards care.
- **Answering cards with different levels of "salt and pepper" and also Gaussian noise**: general image noise, to simulate darker environments or weak printing.

We considered measuring the correct TopCodes decoding from 9 different distances — approximately ranging from 0.55 meter until 8 meters — keeping the A5-sized cards in a fixed position: we used the recommended TopCode size, focusing on the corruption correction at different distances, since we had the hypothesis the hairline effects impact the maximum camera distance. Once again the measurements considered only HD resolution images. Figure 28 illustrates this test scenario.

4.2.2 Detection and decoding distance evaluation

Another experiment we included is the maximum detection and decoding distance, a crucial parameter for classroom usage, which needed to be evaluated for the three differ-

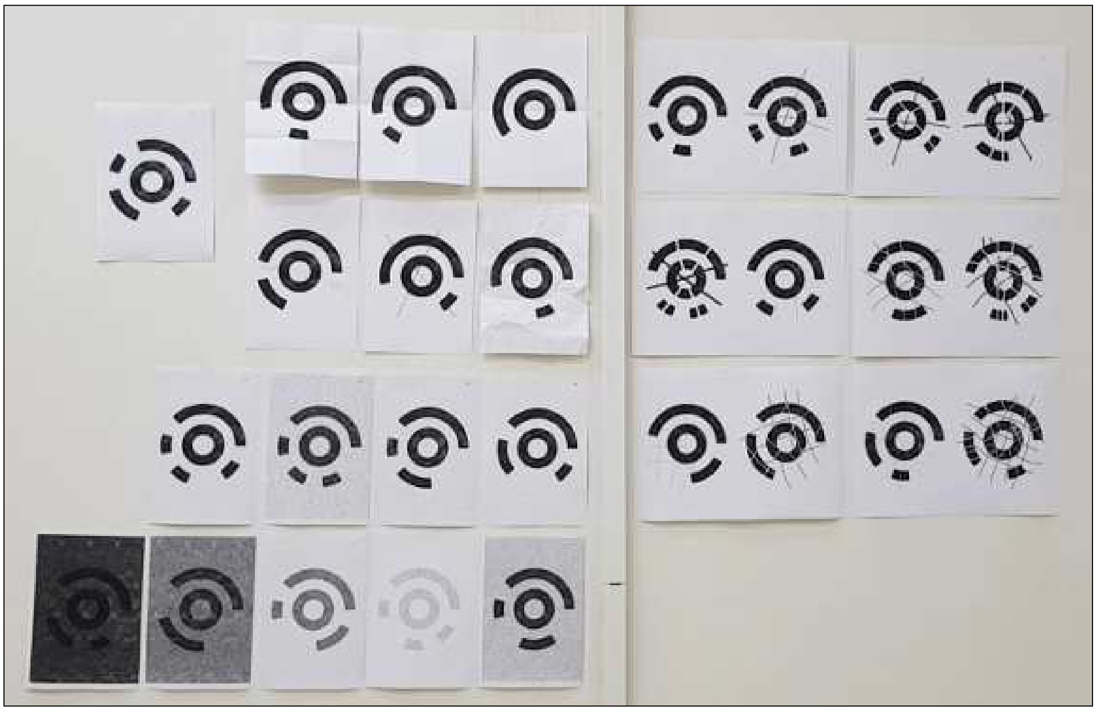

Figure 28 – All the TopCodes corruption cases were tested at once, in a setup which evaluated the corruption effect on different distances.

ent TopCodes sizes. We had the understanding some of the added robustness operations would affect the maximum distance; hence, this experiment intended to confirm and quantify that effect, exploring the following variations:

- Applying only **time-consistency** operation: that would provide the baseline values, since we considered it could not be dropped, since it fixes the partial occlusion. Also, due to its nature, it would not affect the maximum detection distance.
- Applying **time-consistency** and **median filter**, using a 5×5 **filter**. We did not consider bigger filter sizes due to the huge execution time impact, verified during the method implementation.
- Applying **time-consistency** and **morphological operations**, using a 3×3 **element**.
- Applying **time-consistency** and **morphological operations**, using a 5×5 **element**.

We considered for this experiment fixed TopCodes of all three different sizes (A4, A5, and A6), at 12 different distances — approximately ranging from 2 meters until 13 meters; Figure 29 illustrates the experimental setup. This test was the only considering HD and Full HD resolution images.

Figure 29 – Experiment setup for the detection distance evaluation; TopCodes of 3 sizes were positioned on the classroom's back wall, for detection distances ranging from 2 to 13 meters – in this picture, the distance is 11 meters.

4.2.3 Full class detection and decoding scenario

The last experiments scenario was also an attempt to mimic a classroom paperclickers regular usage: we added 72 different TopCodes, printed on all the 3 different sizes, and fixed them on the desks of approximately 15 meters by 8.5 meters classroom, emulating a crowded environment. The diagram in Figure 30 reproduces the classroom organization, including the TopCodes position and sizes. The goal was to verify the overall performance in a classroom setup closest to the one of a crowded class, using the results to guide paperclickers usage.

In this scenario we applied TopCodes of the three different sizes, making sure we have at least a TopCode of each given size at a specific distance; we described the following scanning procedures, trying to reproduce possible teacher behaviors:

- Start scanning from the center of the classroom; then, remaining at the same position, turn left and then turn right, to capture the entire class.
- Start scanning from the center of the classroom; then, keep scanning with the camera as parallel as possible to the students' cards, and start walking to the left until you have captured all students on the left; then, start walking to the right, until covering all the missing students on the right of the initial position.
- Start scanning from the left of the classroom trying to keep, as much as possible, the camera as parallel as possible to the students' cards; then, start walking to the right, until covering all the students in the classroom.

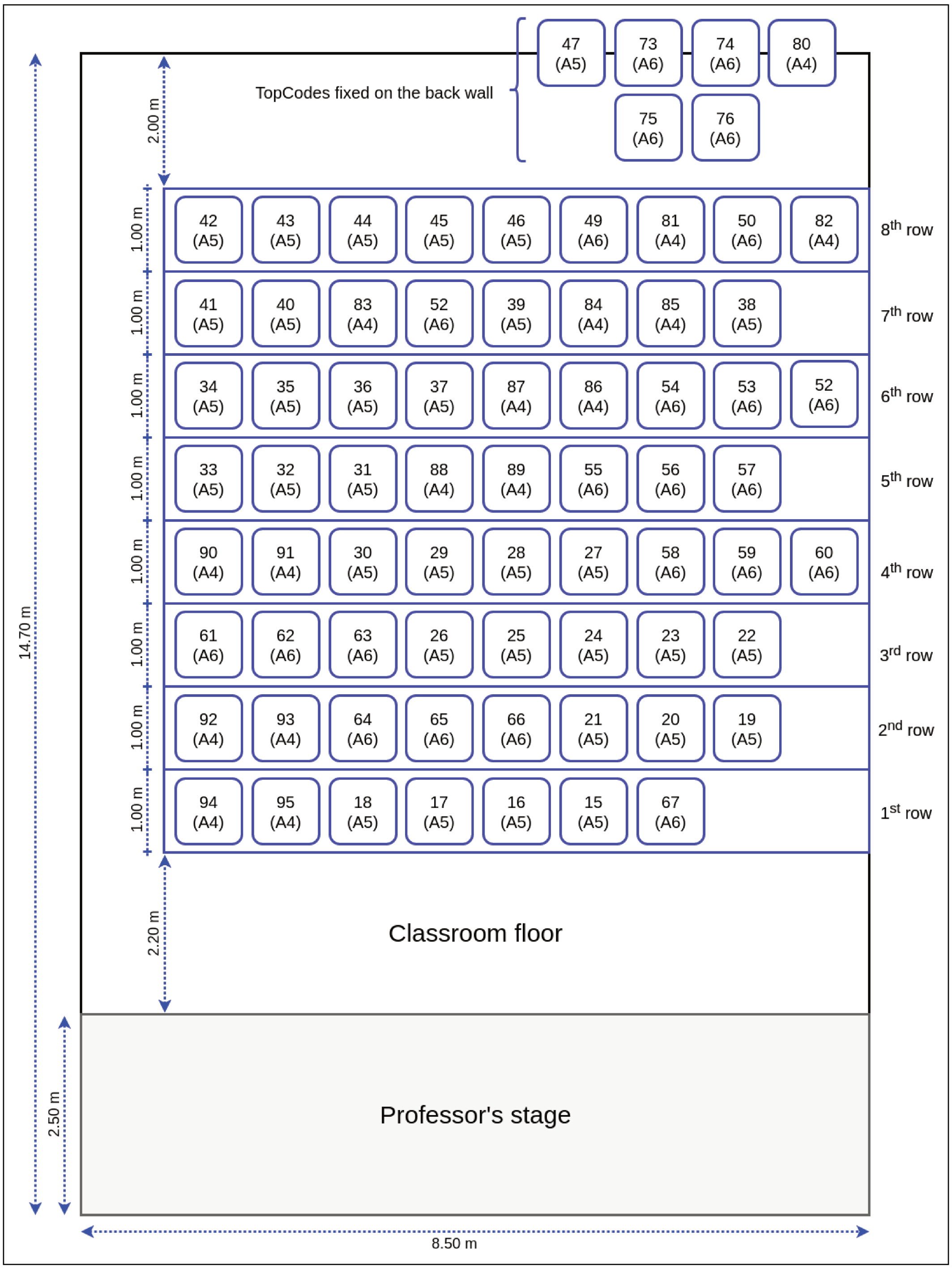

Figure 30 – To emulate a regular paperclickers usage scenario, we distributed 72 TopCodes throughout the students' chairs, including codes of each size at every distance. This diagram depicts the TopCodes distribution inside the classroom; that setup was used for the detection distance (TopCodes on the back wall) and the full class detection experiments.

Those 3 scanning procedures were executed from two different locations: from the classroom floor and from over the stage, near the blackboard: that produced not only distance differences, but also camera baseline changes. Figures 31 and 32 depict the overall classroom aspect.

Figure 31 – One of the entire class detection experiments, to evaluate the overall performance and the effect of the scanning procedure. In this setup, the scanning occurred from the class floor, starting scanning on the left and walking to the right.

Figure 32 – Another setup for the entire class detection experiment: in this one the scan started from the professor's stage, starting scanning from the center of the classroom.

4.2.4 Experiments common setup

All the experiments were performed within a regular classroom of the researchers institution. The paperclickers application was configured to recognize the complete set of 99 TopCodes: even though none of the test scenarios included all of them, we preferred that setup to force the application to consider valid all the TopCode values, increasing the overall processing time and reducing the detection robustness of validating only a small set of codes — paperclickers tries to decode only the defined number of valid TopCodes, corresponding to the **number of students** defined in **Settings**.

The used classroom had a theater setup, with the students rows in increasing elevation from the teacher's stage. Also, the lighting conditions were practically only artificial, since the classroom had no direct daylight exposure other than the entrance door.

4.2.4.1 On experiments reproducibility

In order to control the experiment conditions, we considered the best approach would have the scanning operation through a video recording analysis: in that way, we could make sure the environment conditions were the same when analyzing all the different tested variables. To achieve that, we redesigned the scanning modules, in order to allow the TopCodes detection and decoding process from a recorded video feed, instead of from the live camera, creating what we called the playback feature.

The video playback experience was mostly the same as the one using the live camera feed, besides being a little bit slower — we have simply used basic video playback capabilities offered by the application platform. For the detection and decoding experiments, however, the most important part was guaranteed, since the image processing steps were the same.

The only difference was exactly due to the slowness of the video frames extraction of the regular playback interfaces: depending on the overall phone load, some video frames were dropped and not analyzed. This behavior, however, was not exclusive for the video playback interfaces: although less frequent, under processing load pressure, frame dropping could also happen when processing the live camera feed.

We also expanded the paperclickers settings feature of the application, adding new debug mode parameters, to allow the tests configuration without the need of rebuilding the application. The debug mode is accessible through a hidden touch sequence in the initial screen[6]. The application ended up with a fairly extensive list of configurable development parameters; we present below the most important ones:

1. **Enable TopCodes validation**: control the usage of time-consistency process to validate the TopCodes during the detection phase.
2. **Validation threshold**: TopCodes *validation increasing threshold*, with the default value of 32; it represents the number of cycles to start increasing the *TopCodes validation threshold*, which has the initial value 3. After the *validation increasing threshold*, the *validation threshold* is incremented at every 32/3 cycles.
3. **Show TopCodes validation process**: a simple visual parameter, enabling the presentation of a decreasing validation counter for each TopCode identified in the camera frame.
4. **Allow answers changing**: we decided to leave configurable at run-time the detected answers changing option, on **Detailed Answers** screen; we have disabled that after the internal heuristic evaluation (see section 3.3); future user experiments could easily evaluate the effectiveness of that feature.

[6] Debug mode can be toggle touching 5 times the *paperclickers* word

5. **Use camera emulation from file**: this option enables the video playback feature, essential to the experiments reproducibility.

6. **Use morphological operations**: this option enables the morphological operations usage, as described in section 4.1.3.

7. **Morphological element size**: whenever the morphological operations are enabled, this parameter defines the morphological element size — it will be a square with this number of elements on each side.

8. **Reset onboarding**: to force presenting again the onboarding sequence, even after the very first execution.

With the structure above, all the tests scenarios could be executed through the following procedure: the test sequences were initially video recorded, creating a set of 46 different video files; then, all the different test scenarios setups were executed and analyzed, using the paperclickers playback feature. That procedure made sure all the tests variants were executed using almost the same images sequence. As an attempt to minimize the frame dropping effect, we executed all the testing scenarios letting the same video sequence play twice, and the reported result was the final detection and decoding status.

All the effort above were initially intended to create a testing environment to facilitate the designed experiments: the video playback feature facilitated the tests execution, allowing the tests scenarios preparation and the video capture all at once, in a single afternoon; with the videos recorded, the analysis could be done later, through "laboratory" work.

An unintended, but relevant, collateral of that approach was also making the experiments reproduction easier: we included the recorded videos as part of the open-source repository[7]; any further development can use the exactly same test cases, establishing a common baseline.

4.2.5 Experiments analysis

The experiments analysis allowed the following conclusions:

- The time-consistency mechanisms drastically reduces the detection errors due to partial occlusions;

- Using morphological operations indeed reduces the maximum detection distance in a near prohibitive way;

[7] https://github.com/learningtitans/paperclickers/tree/master/experiments/videos

	Image resolution of 1920 × 1080						Image resolution of 1280 × 720					
	A4		**A5**		**A6**		**A4**		**A5**		**A6**	
	Max	%	Max	%	Max	%	Max	%	Max	%	Max	%
Time-consistency only	13.00	100.00	10.00	100.00	7.00	100.00	11.00	100.00	8.00	100.00	5.00	100.00
Median filter (5 × 5)	9.00	69.23	6.00	60.00	5.00	71.43	7.00	63.64	4.00	50.00	3.00[b]	60.00
Morphological operations (3 × 3)	11.00	84.62	7.00	70.00	5.00	71.43	7.00	63.64	4.00	50.00	3.00	60.00
Morphological operations (5 × 5)	6.00	46.51	4.00	40.00	3.00[a]	42.86	4.00	36.36	2.00	25.00	2.00	40.00

Max – Maximum detection distance for the given TopCode size, in meters
% – Percentage of the time-consistency only
[a] Only 3 of the 4 available TopCodes were detected
[b] One A6 TopCode (75) were detected at 4.00 meters

Table 9 – Maximum detection distance decreases when applying operations to reduce the wrong decodings due to hairline effects; increasing the image resolution might balance that reduction. Time-consistency was active in all cases — detection distance experiments results.

- Morphological operations increase the detection robustness, but can be dropped for a typical classroom environment;
- The chosen scanning procedure improves the final result.

The complete tests results compilation can also be found in paperclickers github[8]. We discuss each of the conclusions in the following sections.

4.2.5.1 Maximum detection distance

The maximum detection distance analysis provided quantitative data about the answering card size effect, as well as how it is moderated by the operations added to overcome the detection and decoding issues (section 4.1).

As previously predicted (see Table 7 in section 4.1.3), morphological operations reduce the maximum detection distance, once the morphological element size causes the TopCode sectors and units to be merged. Median filter performance was the same of morphological operations using 3 × 3 element.

A critical improvement factor for the maximum detection distance is the camera resolution: using FULL HD camera preview images — not widely available on current smartphones — improves the maximum detection distance, turning the smaller answering card size (A6) into a usable solution.

The maximum distance experiments demonstrated the morphological operations greatly penalizes the overall usability, being usable only when the answering cards are printed in the bigger size (A4), and used within a short distance — a setup which might not be the most common one. Table 9 compiles this experiment results.

[8] https://github.com/learningtitans/paperclickers/tree/master/experiments

		Nothing[a]					Morphological operations 3 × 3					Morphological operations 5 × 5				
		Sc 1	Sc 2	Sc 3	Sc 4	Sc 5	Sc 1	Sc 2	Sc 3	Sc 4	Sc 5	Sc 1	Sc 2	Sc 3	Sc 4	Sc 5
No time-consistency	Total detected	18	7	14	10	5	12	8	12	7	3	14	9	13	8	3
	TopCode 18 answer(s)	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	TopCode 68 answer(s)	A	B	D	-	-	A	B	D	-	-	A	B	D	-	-
Using time-consistency	Total detected	2	2	2	1	1	2	2	2	0	1	2	2	2	1	1
	TopCode 18 answer(s)	A	A	A	A	A	A	A	A	-	A	A	A	A	A	A
	TopCode 68 answer(s)	D, A	B, A, B	B, A, D	-	-	D, A	B, A, B	B, A, D	-	-	D, A	B, A, B	B, A, D	-	-

[a] Indicates no treatment for hairline effects avoidance applied
Sc 1 – Scene 1 – Finger and white object (answering code) occlusion 1
Sc 2 – Scene 2 – Finger and white object (answering code) occlusion 2
Sc 3 – Scene 3 – Finger and white object (answering code) occlusion 3
Sc 4 – Scene 4 – Dark object (phone) occlusion 1
Sc 5 – Scene 5 – Dark object (phone) occlusion 2

Table 10 – Using the time-consistency approach drastically reduced the wrong decodings due to TopCodes partial occlusion, regardless the additional image treatment applied — partial occlusion experiments results.

4.2.5.2 Time-consistency efficiency

Time-consistency was the approach to minimize the detection errors due to codes partial occlusion, and these experiments confirmed detection and decoding using time-consistency only returned correct TopCodes, regardless of the partial occlusions. However, one experimental scenario — when morphological operation with 3×3-sized element were applied along with the time-consistency — no TopCode was recognized at all. Without using time-consistency, the number of returned TopCode answers varied from 18 (when the correct answers were only 3) to 2 (when there was a single correct answer).

The tests demonstrated the time-consistency procedure is effective to reduce the detection and decoding errors due to the partial occlusions, and its efficiency is not affected by the use of morphological operations. However, the methods combination might reduce the overall detection rate, which is indeed an already expected result of the morphological operations or the median filter usage, when related to the maximum detection distance. Table 10 consolidates the overall results of the partial occlusion tests.

4.2.5.3 Detection and decoding robustness to answering cards quality

Morphological operations do increase the detection robustness, but at more than 2 meters distance, the increased robustness is very marginal, compared to the simple usage of time-consistency. On the other hand, morphological operations greatly decrease the maximum detection distance, as already demonstrated. As shown in Figure 28, the corrupted codes not detected at all represent really hard scenarios.

Taking into consideration the experiments results and usability concerns — the

		Time-consistency only									Morphological operations 3 × 3									Morphological operations 5 × 5								
	Distance (m)	0[a]	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
	one fold (1)	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0
	two fold (2)	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
	three fold (3)	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
	creased (4)	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
	written 1 (5)	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0
	written 2 (6)	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
2 + 2	1 pixel (7)	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
2 + 2	3 pixels (8)	0	0	0	0	1	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
2 + 2	6 pixels (9)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2 + 2	9 pixels (10)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2 + 2	12 pixels (11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 + 4	1 pixel (12)	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
4 + 4	3 pixels (13)	0	0	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
4 + 4	6 pixels (14)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
6 + 6	1 pixel (15)	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
6 + 6	3 pixels (16)	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
8 + 8	1 pixel (17)	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
8 + 8	3 pixels (18)	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
S & P	5% (19)	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
S & P	10% (20)	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
S & P	25% (21)	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
Gaussian	m=0; v=0.01 (22)	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
Gaussian	m=0; v=0.02 (23)	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gaussian	m=1; v=0.2 (24)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
Gaussian	m=0.5; v=0.2 (25)	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0
Gaussian	m=-0.5; v=0.2 (26)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gaussian	m=-1; v=0.2 (27)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total detected		8	14	16	17	20	20	19	14	0	19	21	22	18	18	2	0	0	0	20	23	16	2	0	0	0	0	0

[a] The detection distance is 0.55 meters
<n> + <n> – <n> black and <n> white traces corruption, with the number of pixels (column 2) width
S & P – Salt and Pepper noise, of given percentage (column 2)
Gaussian – Gaussian noise; m = mean, v = variance (column 2)
The number in parenthesis (column 2) indicates the TopCode tested

Table 11 – Although the morphological operations provided the best recognition rates, they reduce too much the maximum detection distance; using only time-consistency allows similar detection rates starting from 3 meters — detection robustness experiments results.

strong reduction of maximum detection distance; the minimally acceptable quality of the TopCodes answering cards; and the minimum distance the teacher is supposed to be from the students — we decided to turn off the morphological operations usage. We privileged the maximum detection distance factor, a characteristic with great impact on the overall paperclickers user experience. Table 11 compiles this test scenario results, supporting that decision.

		Time-consistency only			Morphological 3 × 3			Morphological 5 × 5		
		Proc 1	Proc 2	Proc 3	Proc 1	Proc 2	Proc 3	Proc 1	Proc 2	Proc 3
72 TopCodes, from floor	Total detected	48	53	52	25	28	27	5	10	7
	Max A4 distance	10.00	10.00	9.00	8.00	7.00	8.00	2.00	3.00	3.00
	Max A5 distance	9.00	9.00	9.00	5.00	5.00	8.00	2.00	3.00	2.00
	Max A6 distance	5.00	5.00	5.00	3.00	3.00	3.00	-	-	5.00
	Errors	0	1[a]	0	1[a]	0	0	0	0	1[b]
66 TopCodes, from stage	Total detected	47	47	49	21	21	19	2	3	4
	Max A4 distance	11.20	11.20	11.20	9.20	9.20	9.20	4.20	5.20	5.20
	Max A5 distance	11.20	10.20	11.20	7.20	6.20	6.20	-	-	-
	Max A6 distance	6.20	6.20	6.20	5.20	4.20	-	-	-	-
	Errors	0	0	0	0	0	0	0	0	0

Proc 1 – Scanning procedure 1 – “start center, turn left, turn right”
Proc 2 – Scanning procedure 2 – “start center, walk left, walk right”
Proc 3 – Scanning procedure 3 – “start left, walk right”
Detection distances are in meters
[a] Errors corresponding to the detection of a TopCode not present
[b] Error corresponding to the detection of a wrong answer

Table 12 – Simple recommendations on the scanning procedure can clearly affects the overall detection results — full classroom detection experiments results compilation.

4.2.5.4 Scanning procedures recommendations

The last setup of the detection experiments demonstrated the effects of the scanning procedure, in the paperclickers normal usage environment. Depending on how the user performs the scanning sequence, the detection results changed. The greater difference was due to the camera baseline: keeping the camera parallel to the codes produced the best scanning results: walking through the front of the classroom is the recommended procedure, instead of turning to the sides from a fixed position. We included that recommendation in paperclickers usage training material (section 5.3.1). This test scenario also reinforced the previous results, regarding the maximum detection distance and the reduction effect due to applying the morphological operations. Table 12 compiles the experiment results.

One relevant result of this test scenario was the TopCodes wrong decoding due to partial occlusions: after all 18 scanning rounds (9 detection procedures, 2 scanning positions), there were 3 wrongly detected and decoded TopCodes, on the following conditions:

1. When applying **morphological operations** the with 3×3 **pixels** size elements, during the first detection procedure — *“start center, turn left, turn right”* — from the classroom floor;

2. When applying only the **time-consistency** method, during the second detection procedure — *“start center, walk left, walk right”* — from the classroom floor;

3. When the **morphological operations** were applied, with 5×5 **pixels** size element, during the third detection procedure — *“start left, walk right”* — from the classroom floor;

The first two errors corresponded to the detection of TopCodes which were not present in the classroom — TopCode 77 and TopCode 99 respectively —; the last error was the detection of a wrong answer for a TopCode in the classroom — TopCode 58, answer “B” instead of “C”.

Those decoding errors resulted from the partial occlusions present during several decoding cycles, right at the beginning of the detection cycle. The way the time-consistency was implemented (see section 4.1.1 for details), makes it particularly sensitive to a very static scenery of partial occlusions, when detected at the beginning of the scan cycle, when the detection threshold is still low (3 consecutive frames). The detection robustness of time-consistency mechanism can be increased, setting a higher value for the initial detection threshold. However, that will affect the overall user experience, slowing down the detection and decoding of any TopCode.

During 18 scanning rounds, 471 TopCodes were decoded, 468 correct (99.36%), and 3 wrong (0.64%). There were 1242 possible decodings (9 scanning rounds of 66 TopCodes + 9 scanning rounds of 72 TopCodes), which indicates only 37.92% of the possible TopCodes were detected, either correctly or not. With all those results, we concluded the following:

- The considered classroom setup is challenging for the paperclickers usage: some students are positioned too far from the teacher, and they are very close to each other, favoring partial occlusions. Even though, using only the time-consistency and the recommended scanning procedure, the detection rate was 74.24% (from the stage) and 73.61% (from the floor).
- A4-sized answering cards should be used on a similar classroom setup: applying only the time-consistency and the recommended scanning procedure, all A4-sized answering cards were detected.
- One hypothesis for the decoding errors is the fact the scanning rounds were all taken from the classroom floor, and the higher angled camera baseline was causing more partial occlusions, since the students rows increased in height.
- The decoding performance is acceptable, but there is still room to improve the solution robustness.

4.3 Ensuring codes detection is the major usability challenge

Considering the constraints on the paperclickers usage scenario, and the work done during its development, we concluded the major usability challenge is making sure the answering codes are reliably detected: it is the central feature of an image processing based CRS, and the usage scenario dynamic characteristics, makes that especially challenging.

The user satisfaction — and arguably the effective solution usage — will depend on finding the balance between additional validation cycles (for instance to avoid decoding errors due to partial occlusion or hairline effects) and the performance cost they imply. An unreasonably slow solution can similarly affect the overall experience, as well as the failure to detect some answering cards. We tried to privilege the reliability of the answers readings, reducing as much as possible the probability of having wrong decoding results.

Although TopCodes solution had been selected due to its speed and robustness, we had to face unexpected challenges when we transported it to the CRS usage scenario. Those were exactly related to its detection and decoding robustness and speed, when collecting the students' answers inside a classroom.

The experimental results provided quantitative data to guide the design decisions during the paperclickers enhancements. The obtained information forced a compromise solution between detection robustness and user experience: we dropped some development approaches, which do increase the overall detection robustness to answering cards corruption, to preserve the final solution experience; however, we were able to determine usage conditions which greatly minimizes the detection issues.

With the increasing processing power and camera resolution reaching the smartphone mass market, the paperclickers solution can — and should — be improved, both regarding the detection robustness and usability. Additional detection an decoding experiments need to be performed on real-world scenarios, considering the selected target audience of Brazilian public high schools. Its environmental conditions will certainly vary — for example, it is reasonable to suppose, the theater setup (with elevating students rows) will not be the most common one. We believe the experiments reproduction mechanisms created, along with the all experiments materials release, enables future work extending paperclickers research.

5 Effective deployment of paperclickers and Peer Instruction

Through the literature review, we recognized that the human factor is determinant when deploying a new technological pedagogical tool, hence that should be further considered in paperclickers research. In this chapter we start from the initial paperclickers goal of reducing the adoption barrier represented by the infrastructural cost and complexity, to discuss in Section 5.1 why that is not enough to achieve effective delivery of our CRS solution, since we have not tackled the adoption psychological barriers.

In Section 5.2, we detail the inclusion of usage guidelines inside the application, as the initial approach to reduce the usage barrier for the technology itself. Then, mostly inspired by Unified Theory of Acceptance and Use of Technology (2.4.2), in Section 5.3, we devise our strategy of creating training materials, using the format of self-contained video tutorials, to introduce both the tool and the methodology. We also suggest a Peer Instruction class material creation guideline (Section 5.3.3) to increase the confidence and reduce the effort expectancy of teachers. We conclude (Section 5.4) indicating how we contributed to the original paperclickers work, towards the effective delivery of a new technological pedagogical tool.

5.1 The approach for an effective classroom response system

We propose that for a technological pedagogical tool be effectively used by a target audience, it has to be meet the following criteria, taken from several well-known dimensions:

1. It has to provoke learning gains.
2. It has to be cost-effective.
3. It has to be usable and easy to learn.

The initial paperclickers conception aimed to cover all those aspects, employing different approaches and perspectives. To achieve learning gains, paperclickers — a feedback assisting tool — was associated with Peer Instruction, a proven active learning methodology which can be facilitated by a CRS. The cost-effectiveness was indeed the main innovative aspect of paperclickers, since it relied on image processing techniques to compose a CRS through the usage of a single smartphone by the teacher and a set of paper answering cards for the students. No additional hardware or connectivity infrastructure is required, besides the access to a reasonable quality printer. The present work advances Bindá [2015] research in the usability aspect of paperclickers, through the user tests analysis and further development, described in Chapters 3 and 4.

However, as identified in the literature, those aspects do not suffice for the effective deployment of a new technological pedagogical tool: paperclickers has to overcome other deployment barriers existing on the target environment, especially the ones related to psychological aspects of the people involved, both teachers and students. Inspired by models like Four in Balance and also the TPACK (Section 2.4), we recognize no pedagogical gains can be achieved only through the development of a simple tool: methodological changes are required. A tool can facilitate a new working methodology, but the teachers have to adhere to that new method, changing the way they teach.

We extend the paperclickers effectiveness discussion, including those additional environmental and psychological dimensions; we employed the literature review methodology to explore works related to technology adoption and deployment, to propose the creation of training material, with the focus of enabling the teachers and instructors to use both paperclickers and Peer Instruction. Although we recognize they do not control all aspects related to the existence of a new technological pedagogical tool in a given environment, we understand they are the fundamental actors requesting, suggesting, supporting and, indeed, using those tools in any pedagogical setup.

Supported by the literature, we hypothesized training materials could motivate and increase the teachers' confidence and comfort to use paperclickers and also to prepare their classes employing the Peer Instruction methodology. Psychological barriers to adopting novelties inside the classroom are directly related to the comfort and confidence the teachers feel, and that is true not only for a new technological tool but also for a new teaching methodology. As studied in the literature, those barriers are linked both to very tangible knowledge — as the given technology usage competence, or the new methodology process mastery — but they are also associated with personal motivation to use the methodology or tool, or particular expectancy on the performance gains or associated efforts on doing so.

In order to tackle, at least to some extent, all those barriers, we planned and started to create training material for applying paperclickers, associating its usage with a proven pedagogical methodology (Peer Instruction) as a way of reducing the psychological adoption barriers. We also embedded usage instructions in paperclickers, lowering the knowledge required to its first usage. We aimed at increasing the teachers confidence to effectively use the tool and implement the Peer Instruction methodology, in order to obtain learning gains.

5.2 Addition of onboarding and instructional overlays

In order to reduce the technological barrier for the tool adoption, we considered *onboarding techniques* — welcoming the users for the solution — and *instructional overlays* — usage hints presented as overlays on the application screens, also known as *coach marks*. Both practices are aligned with the current industry standard of smartphone applications. Nielsen Norman Group [Harley, 2014] indicates instructional overlays and coach marks can be *"helpful to the user to get a nudge in the right direction"*, but they need to be designed for *"optimal scannability"*: they should be short, focused on fewer items or features, contrast with the regular UI and be visual as much as possible. Nielsen Norman group also discuss the *progressive disclosure* of application features [Nielsen, 2006] as a method to help users, showing first the most important functionality to make *"applications easier to learn and less error-prone"*.

Google provided some onboarding[1] and *feature discovery*[2] guidance for applications on their Android mobile phones platform, which resonate similar perspectives. For the onboarding, they suggest three different models:

- **Self Select** — when the initial application state is to guide the users to customize their experience, frequently performing the required setup.
- **Top User Benefits** — a very common model of exposing right at the beginning the application value proposition, commonly implemented as an *autoplay carousel* showing the top functionality of the application.
- **Quickstart** — guide the user straight to the application's most engaging features.

The *feature discovery* assimilates the *progressive onboarding* principle, which is also an industry standard of presenting new features to the users at relevant moments. Hence, when using such approach, it is crucial to define those right moments — when those features are needed and can be better assimilated —, as well as the proper amount and frequency of that presentation.

Based on those recommendations, we embed in paperclickers an instructional path aiming to deliver the most critical information for the tool usage, but in the less intrusive way as possible. The diagram in Figure 33 depicts this flow: we used the following rationale to include a combination of the current industry standard practices:

[1] https://material.io/guidelines/growth-communications/onboarding.html
[2] https://material.io/guidelines/growth-communications/feature-discovery.html#feature-discovery-design

1. Adding an onboarding step for the first usage of paperclickers, emphasizing the tool benefits and the main required steps to start using — e.g., print the answering cards —; as usual, this step can be skipped by the savvy or impatient user.

2. Including an optional guided usage to offer education for the user, aligned with the *progressive onboarding* technique: if the user activates this option, the instructional overlays are presented for the initial usage of some key features.

3. Use the *feature discovery* technique, as part of the optional guided use, to present instructional overlays including hints on how to access key features.

Despite the understanding that a good user interface should be intuitive enough to be self-explainable, we preferred to include that additional instructional information since:

- The industry standard is to provide that kind of information, and being aligned with that might follow the regular user mindset.

- The selected instructional information is crucial to the overall solution usage, and includes elements or tasks outside the smartphone application usage — like the students' answering cards print or the answers' log sharing.

Our approach here was to risk annoying some savvy users on their first paperclickers execution, providing some additional information, while trying to reduce the usage difficulties and adoption barrier, according to our behavior expectation of the targeted users.

5.3 Designing a training material for effective deployment

While designing a training material on the paperclickers solution and Peer Instruction methodology, our primary goal was to provide the required competencies and also the motivation to use the methodology and tool combination, to achieve the effective use of an active learning methodology, facilitated by our CRS solution. We planned the production of video tutorials, depicting and enhancing the essential aspects of both the tool and methodology. We chose video tutorials medium as they are now a ubiquitous method of providing training and information, they can convey satisfactory knowledge delivery without much supervision and can also reach people without too much formalism or infrastructure, even in developing countries: Google announced Brazil is the second country in numbers of YouTube watched hours [3].

[3] https://www.tudocelular.com/android/noticias/n90175/google-for-brasil-numeros-youtube-waze.html

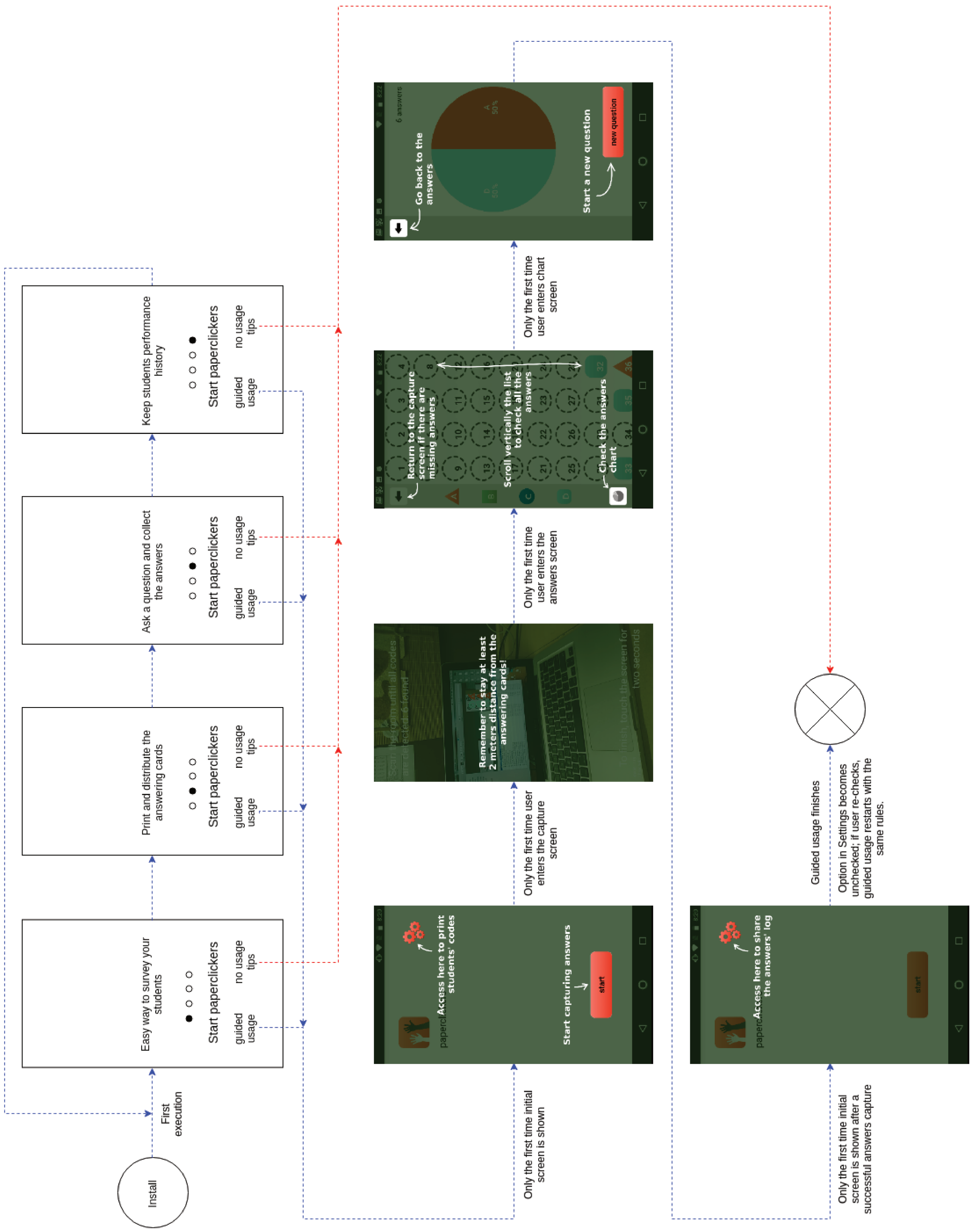

Figure 33 – Paperclickers onboarding and overlay behavior, included to provide focused and timely information on how to use the major features.

According to Guo et al. [2014], careful video production can enhance the viewers' engagement, and hence, increase the effectiveness of the information transfer. In this sense, we contribute to the paperclickers effectiveness providing an initial set of scripts for the video tutorials, establishing a basis for the training material design.

We fixed our target audience to propose training content which better suits their needs: we selected *high school classrooms of Brazilian public schools*, especially in science and mathematics fields. We believe a technological pedagogical tool as paperclickers could foster a relevant social impact on that public, particularly when backed by a sound and flexible active learning methodology as Peer Instruction. Overall evaluation of Brazilian 15-year-old students on those areas indicates performance below the world average, according to OECD [2016]; also, school evasion is greater after elementary school years, as shown in Figure 34.

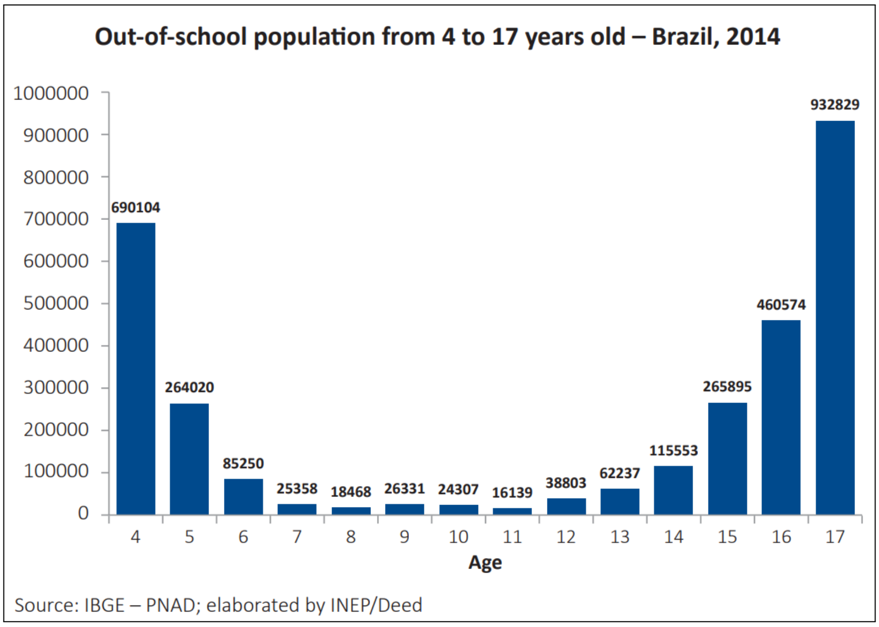

Figure 34 – Out-of-school rates in Brazil are concentrated in preschool and high school ages. Reproduced from INEP [2016].

Working with video tutorials for the same audience was the choice for another project developed within our research group, aiming to develop computer programming skills on high-school students [Celeri et al., 2017]. We designed video tutorials as pedagogical tools for both students and teachers, to support not only the classroom activities, but also the overall course organization.

Guo et al. [2014] have evaluated factors influencing student engagement in MOOC (Massive Open Online Courses), which are the best example of video-based learning, and produced a set of recommendations about how the video production affects engagement. We based our video tutorials production on those recommendations, especially on the following items:

- **Shorter videos are more engaging**: median engagement time is at most 6 minutes; videos up to 9 minutes still presents a high completion rate.
- **Preference for “personalized” videos**: informal settings, with direct eye contact and giving the impression of personal talk seems to be more engaging, when

compared to high-value production settings.

- **“Khan-Style” tutorials are more engaging**: handwriting to solve problems or sketching over slides produce greater engagement.
- **Pre-production matters**: investment in pre-production results in more engaging videos, even if the chosen style is to record a live classroom lecture.
- **Students engage differently with lecture and tutorial videos**: lecture videos should provide a good first-time watching experience, while video tutorials should support re-watching and skimming, since they are used as reference material.

Considering the recommendations above and our defined target audience, we will guide the videos design process by the following principles:

1. **Use simple and direct language**, privileging the information accessibility, in order to avoid an additional barrier related to the information clarity and comprehension.
2. **Produce short videos**, no longer than 9 minutes, targeting the 6-minute threshold indicated by the research; the required information shall be conveyed in small chunks.
3. **Explore personalized setups for methodology training**, looking for engagement on learning about Peer Instruction with videos showing real experiences.

5.3.1 Paperclickers usage training material

The first aspect we wanted to support through the training material is to provide information on how to use the paperclickers tool: we designed a set of video tutorials to guide the teachers through all steps of the tool usage, from initial setup until students' performance history recall. With that material, we intended to actuate on the following concerns of the analyzed models and frameworks:

- To answer some questions of MIT framework, especially on the **Competence**, **Role** and **Classroom Management** sub-areas of **Teachers** concerns.
- Actuate within the **Content and application** and **infrastructure** axes of Four in Balance model.
- Deal with the **effort expectancy** and **facilitating conditions** constructs, directly related to using the tool on the UTAUT framework.
- Improve the **Technology Knowledge (TK)** of TPACK model.

The goal is to address the training needed to use the new technology, as well as to make all the physical and environmental arrangements paperclickers usage requires. We designed instructional videos exploring each step the research team considered crucial for paperclickers proper usage, presenting the required application flow as well as the external arrangements.

Following the guidelines for video production (Section 2.4), we split the information into series of short video tutorials, facilitating the reference. Our final goal is to release the videos in a video sharing platform like YouTube, and refer them within the paperclickers tool distribution.

Tables 13, 14, 15, 16, 17, 18 and 19 present all the seven scripts created for the training videos about how to use paperclickers application. The original scripts version were written in Brazilian Portuguese (reproduced in appendix A), since the videos will be created in that language, according to the target audience.

Paperclickers training video 1 – Presentation and basic usage	
Video description	*Text script*
Title: “What is paperclickers?”	
Smartphone in front, showing scanning screen; classroom visible in background, with students using the answering cards	Paperclickers is a cost-effective solution for you to quickly collect your students’ responses in the classroom: you ask a question and request your students to use the coded cards to present the answers; then you use paperclickers on your Android phone to capture them.
Using the same scene structure, finalize scan, present answers’ screen and then goes to chart screen	This makes it easier to have a dynamic class with more participation. And as the answers get recorded, you can use them to prepare for the next class, knowing how the students previous performance was, and even controlling who was in class that day.
Title: “How does it work?”	
Animation showing application usage: teacher presents the question; students think about their own answers; students choose their answers rotating the answers’ cards — close showing corresponding answer in the card’s back; teacher captures the answers; teacher checks the result	You ask a multiple choice question, with up to 4 possible answers; students choose their answers by turning the card to the corresponding orientation and presenting the card; you collect and record the answers with the application on your cell phone and you know the opinion of the class right away, without having to keep counting their raised arms. With the answers you will know if you need to work more on the subject with your students or if you can move on.

Table 13 – Scripts for the first training video, presenting paperclickers.

Paperclickers training video 2 – Installation and initial execution	
Video description	*Text script*
Title: “Paperclickers installation and initial execution”	
Smartphone screen showing: “Google Play store” access; start installing paperclickers	It’s very simple to start using paperclickers; the first thing to do is to install the application. To not spend the credits on your phone, use a WiFi network; enter the Android application store and search for the paperclickers application, and ask to install it.
Finished installation; user requests execution; first execution presenting onboarding screens	Once installed, start paperclickers and be guided on your first use: initial screens will present the main features and the first steps to start using.
User enters “Settings” menu and selects “Number of students” option; number of students definition dialog opens	The first thing to do is to set the number of students you will be working with: this can be done within the *Settings* option. Choosing this number is important as it will define which answering cards will be valid. Our suggestion is that each of your students have their own answering card, because then you will always know which response a student gave for each question you have already asked.
	… This script continues in the next page

Continuation – Paperclickers training video 2 – Installation and initial execution

Video description	*Text script*
Animation showing 30 grouped icons identifying each student of a class; after four new icons appears, representing the additional answering cards; the recommended total of 34 answering cards number appears	If you work with paperclickers with only one class, enter the number of students in that class, adding 4 or 5 as a reserve; for example: if your class has 30 students, enter 34 in the application. This way you will have some additional answering cards in case someone loses or forgets her own card.
Animation showing the limit of 99 answering cards in the same class	Paperclickers supports up to 99 students in the same class; unfortunately, its usage is not suitable for larger classes.
Thumbnail for the referred video	But you can use paperclickers for multiple classes, even if the total number of students exceeds 99. If you are going to use paperclickers for more than one class, see the video *Using paperclickers in several classes* for more information.
Animation showing “paperclickers” name and a thumbs up indication; thumbnail to all the other remaining videos are presented	Everything may seem a bit complicated, but do not be alarmed: in the next videos we will explain in detail how to print the codes and also how to use the responses record to follow the evolution of your students.

Table 14 – Scripts for the second training video, focusing the installation and initial usage of paperclickers solution.

Paperclickers training video 3 – Printing students’ answering cards

Video description	*Text script*
Title: “How to print the students’ answering cards”	
Show the finalization of the number of students definition; return to “Settings” main menu; goes until "PRINT STUDENTS’ CODES" option	Once you have defined the number of students using paperclickers, the next step is to print the answering cards and distribute to them.
Show an A5-sized answering card perfectly printed and another one, in the same size, with printing imperfections; animation showing that card with printing issues should not be used	To print the answering cards you will need to have access to an inkjet or laser printer that has enough ink — or the toner — in order to have a good printing result.
Someone holding an answering card A5-sized, showing the TopCode in its front, and the answering options and holding marks in the back; vertically split the screen in two, simultaneously showing the front and the back of the same answering card; the cars is rotated to indicate each one of the four answers, both the front and the back image rotates accordingly; for each selected answer, shows a text indicating “Selected answer <A|B|C|D>”	Answering cards need to be printed in two-sided, because the front has the identification code, in the circles, and the back indicates which answer option the student wants to display. It will be easier if you have access to a printer that automatically prints two-sided.
Presentation of the 3 different answering cards sizes; they appear stacked, with the bigger (A4) below and the smaller (A6) on top, all aligned by the left lower corner; the A6-sized card slides until the right side of the screen; the A5-sized cards slides to the right, staying between the A4 and A6 cards — from left to right, the cards are ordered A4, A5 and A6 sizes; a text shows under each card: “A4 ~= 11 m; A5 ~= 8 m; A6 ~= 5 m”	You can also set the size to print the answering cards; this size can range from 1 to 4 codes per page. With larger codes you will be able to use paperclickers in larger classrooms: an A4 full-page size answering card is well detectable in a room where you stay up to 11 meters away from your students; a half-page sized code — which corresponds to A5 page size — is visible in a room where you are up to 8 meters away from your students; a quarter-page sized code — which corresponds to an A6 page size — is only visible in a room where you are up to 5 meters from your students.
Shows the selection of “Number of codes per page” option in paperclickers “Settings”, opening the dialog with the options “1 per page”, “2 per page” and “4 per page”	We recommend that you use half-page sized codes. using the *2 code per page* print option, because they support a good class size and are easier for students to manipulate and keep in good conditions.

... This script continues in the next page

Continuation – Paperclickers training video 3 – Printing students' answering cards	
Video description	*Text script*
Shows the selection of "Page format" option in paperclickers "Settings", opening the dialog with the options "A4" and "US letter"	Another parameter that you can change for printing is the sheet size that will be used: you can choose the standard A4 size sheet (210 x 297 mm) or the US letter size sheet (216 x 279 mm).
Someone holding an A5-sized answering card to show an answer, printed in 120 gsm and 75 gsm paper thickness, showing the last one folds over its own weight	One last detail for printing is about the type of paper to be used: the ideal is to use thicker paper for printing the answering cards, as that makes even easier to use the cards; a 120 gsm paper should be enough.
Shows the printing default options inside "Settings" and then the "Print or share codes" option selection; shows the referred video thumbnail	If your printer does two-sided, and you are printing 2 codes per page on an A4 size sheet, you can keep the default values for the printing options. If the printer you are using does not support two-sided printing, check the video *Manual two-sided print of your answering cards* for specific instructions.
Shows the paperclickers popup message informing the generation of the ".pdf" file; shows the popup slider with the sharing options available	Having made these settings, select the option *Print or share codes*: the application will generate a ".pdf" file with the answering cards, according to the settings you have defined. With the default options, the "paperclickers_topCodes.pdf" file will be generated, which you must then send to print.
Shows the "Gmail" sharing application selection; opens the app in the email composition screen; fills the destination address and the subject as "Answering cards for printing"; sends the email	The simplest way to do this is to select the email to send the generated file to yourself; to do this, enter your email address as the destination and send. It is important to remember that you need to be with your phone with WiFi or data connection.
Windows screen of the desktop connected to the printer; access "Gmail" page to read the received email; select the ".pdf" attachment and request to print	Using a computer connected to the printer, open your email and select the message you just sent yourself and ask to print the attached file.
Printer printing the answering cards; printed answering cards being hold to show the printed front and back; a page with two codes being cut in half with a scissors	Once the codes have been printed, cut and distribute them to the students.
	Ideally each student should have their own answering card and always keep it — this will save you time in your classes and ensure that each student always uses the same code, which will make it easier for you to keep up with the evolution of each one of them individually; you can even use paperclickers as a way of recording students presence, doing the roll call of each class through a simple question.
Students list alphabetically ordered; animation showing the first name has the answering card number 1 and so on	One suggestion is to distribute the sequence of answering cards in alphabetical order of your students; then you will know that the first student in the alphabetically ordered list will have the code number 1 and so on.
Show some examples of answering cards in bad conditions — torn up, crumpled, folded and dirty	But to always left the answering cards with the students, they must be careful to keep them in good conditions: very dirty or crumpled answering cards may be hard to paperclickers detect; your students must also always remember to have their answering cards with them, so they can use in class.
Desktop computer with opened browser; entering the URL of the ".pdf" files in paperclickers github area where all the TopCodes ".pdf" files are available	Alternatively, you can directly download the ".pdf" files with all the 99 answering cards from the paperclickers project web page: access the "https://github.com/learningtitans/paperclickers/tree/master/topcodes/en-US" address for the files.
Shows the paperclickers github area browsing, until finding the referred file; select the file to download	You will have to choose the ".pdf" file corresponding to your answering cards printing options; for example, if you want to print the answering cards at half-page size, on an A4 sheet using a two-sided enabled printer, choose the "paperclickers_topCodes_2pp_A4.pdf" file.

Table 15 – Script for the paperclickers training video about how to print the students' answering cards.

Paperclickers training video 4 – Manual two-sided print of your answering cards	
Video description	*Text script*
Title: "Manual two-sided print of your answering cards"	
	If the printer you are going to use does not have the two-sided printing capability, you will need to manually control to have your answering cards printed in both sides.
Shows the selection of "Two-sided" printing option in paperclickers "Settings", opening the dialog and selecting "I have to do two-sided manually" option	To do so, select the *Two-sided* settings option and mark *I have to do two-sided manually.*
Selects the option "Print or share codes" in "Settings"; present the two popup messages indicating the generation of the ".pdf" files	Now choose *Print or share codes* and check that two ".pdf" files are generated, one containing the front of the answering cards and the other containing the back side.
Selects the "Gmail" application for sharing; shows the application in the email composition screen; enter the destination address; enter the subject "Answering cards for printing"; send the email	Send these files to print, selecting, for example, the email option: indicate the *Gmail* application to send the 2 generated files to yourself; to do this, enter your email address as the destination and send.
Windows screen of the desktop with the printer; access "Gmail" page to read the received email; select the first ".pdf" attachment and request to print	Using a computer connected to the printer, open your email and select the message you just sent; request first to print the attached file containing the front of the response codes; that will be the file named "paperclickers_topCodes_recto.pdf"
	Wait for the end of the front-side printed answering cards. Once the printing is complete, take the printed sheets front and reinsert them into the printer to print the back of the answering cards.
Person manipulating the printed front side of the answering cards, in doubt about how to reinsert the sheets into the printer: printed side upwards or downwards? Insert the beginning or the end of the sheet first? Animation showing the doubt	At this point you need attention: each printer has the right orientation to define the printing side of the sheets. You will need to know the orientation of your printer to properly reinsert the sheets.
Shows the icon — common in all the printers — indicating the printing side of the sheets in the printer; animation associating the icon with the correct printing side of the sheet: icon shaded side indicates the side to be printed; animation to indicate the correct insertion side of a sheet with the answering cards front side already printed	There should be an icon in your printer to indicate which side of the sheet will be printed: the shaded side is the side to be printed; in this icon, it is indicated that the side to be printed is the side of the sheet that is downwards inside the printer drawer.
Person still manipulating the front-side printed card, now in doubt about which orientation to insert: Insert the beginning of the end of the sheet first? Animation showing the doubt	Now that you already know the correct printing side, you only need to know the correct orientation of the sheet: will the printing start from the top or from the bottom of the page?
Shows a white sheet; hand write "paperclickers" at the beginning of the page; shows the icon indicating the correct printing side and insert the page properly, with the written side to be printed and the top orientation first	To define this, one way is to do a quick test: write something on the top of a paper; place the paper inside the printer so that it prints on the same side you wrote on it.
Shows printing the back of a answering card; shows the result with the correct orientation, with an animation showing the handwriting orientation is the same of the answering card code number; shows another result with the wrong orientation, with the animation emphasizing the orientation difference between the handwriting and the answering card code number	Now, do a test print of an answering card back side and see how the print came out: if it was in the same orientation that you wrote, then the way you put the paper is the right one; if the orientation is inverted, the correct way to load the paper is the reverse — remember you still have to respect the printing sheet side.

... This script continues in the next page

Continuation – Paperclickers training video 4 – Manual two-sided print of your answering cards	
Video description	*Text script*
Windows screen of the desktop with the received email; select the second ".pdf" attachment and request to print	Now that you know the side and orientation for printing the back side of the answering cards, put the already-printed sheets in the printer in the correct side and orientation and then print the back side of the answering cards, which is the attachment named "paperclickers_topCodes_verso.pdf ".
Printer printing the back of the answering cards; printed answering cards being hold to show the printed front and back; a page with two codes being cut with a scissors	Once the codes have been printed, cut and distribute them to the students.
	Ideally each student should have their own answering card and always keep it — this will save you time in your classes and ensure that each student always uses the same code, which will make it easier for you to keep up with the evolution of each one of them individually; you can even use paperclickers as a way of recording students presence, doing the roll call of each class through a simple question.
Students list alphabetically ordered; animation showing the first name has the answering card number 1 and so on	One suggestion is to distribute the sequence of answering cards in alphabetical order of your students; then you will know that the first student in the alphabetically ordered list will have the code number 1 and so on.
Show some examples of answering cards in bad conditions — torn up, crumpled, folded and dirty	But to always left the answering cards with the students, they must be careful to keep them in good conditions: very dirty or crumpled answering cards may be hard to paperclickers detect; your students must also always remember to have their answering cards with them, so they can use in class.
Desktop computer with opened browser; entering the URL of the ".pdf" files in paperclickers github area where all the TopCodes ".pdf" files are available	Alternatively, you can directly download the ".pdf" files with all the 99 answering cards from the paperclickers project web page: access the "https://github.com/learningtitans/paperclickers/tree/master/topcodes/en-US" address for the files.
Shows the paperclickers github area browsing, until finding the referred file; select the file to download	You will have to choose the ".pdf" files corresponding to your answering cards printing options; for example, if you want to print the answering cards using half-page size, on an A4 sheet, since your printer only do manual two-sided printing, choose the "paperclickers_topCodes_rectoOnly_2pp_A4.pdf" and "paperclickers_topCodes_versoOnly_2pp_A4.pdf" files.

Table 16 – Script for the paperclickers training video with alternate instructions about how to manually do two-sided printing.

Paperclickers training video 5 – Sharing and using the answers log	
Video description	*Text script*
Title: "Sharing and using the answers log"	
Animation showing the paperclickers screen sequence from scanning, detailed answers and chart screens; after that, animation represents the creation of a new answers log record; repeat the same sequence during the text	Every time you use paperclickers with your students, the captured responses are logged internally. You can use this record to track the evolution of your students, class to class, question to question. Hence, it is important you always know which student is using which response code: this log keeps the option answered by each answering card, for each question you asked, identified by the date and time, and sequential number for reference.
	... This script continues in the next page

Continuation – Paperclickers training video 5 – Sharing and using the answers log	
Video description	*Text script*
Animation showing date and time of first question; scanning of the 30 answers; answers recording; restarts animation, now with date and time of the second question	For example: if on February 6, 2018, at 9:15 a.m. you asked a question for your class when there were 30 students present, and used paperclickers to capture their answers, you will have in the log each of the 30 answers given, identified by their answering cards code — in this case, the codes will be from 1 to 30. If on that same day, at 10:00 a.m. you asked another question also using paperclickers to collect the answers, you will have a new log of the new 30 answers given. That way, by consulting those answers records, you can recall later the students' performance, once you know which student is using which answering card code.
Shows a handwritten class planning, with the indication of 2 questions to be made to the class	To have a complete and meaningful record, you will need to have a separate control to recall which were the questions you asked, or at least the subject matter of each class.
Enters paperclickers "Settings"; selects the "Share answers log" option; selects "Gmail" application; at the new email composition screen, enters the destination address and fills the subject with "Class A answers log"; sends the email	To use the answers log, you need to share it from the paperclickers application; to do this, go to the *Settings* option and choose *Share answers log*. Once again, the simplest way is to email that record to yourself: choose, for example, the "Gmail" application, fill in the destination address with your own email address and enter a meaningful subject like *Class A answers log* and send the email, which will have as attachment the "paperclickers_AnswersLog.csv" file.
Shows a windows desktop acessing "Gmail"; opens the received email and save the "paperclickers_AnswersLog.csv" attached file in "Documents" folder	To read the shared answers log, open your email on a computer and access the message you just sent; save the attached file locally on your computer.
Shows "Microsoft Excel" main screen; shows "LibreOffice Calc" main screen	That file is a standard text file, but in a format that allows it to be opened by spreadsheet software, such as *Excel*, from Microsoft's Office package, or *Calc*, in the LibreOffice package — you you will need to use a computer with one of them installed to be able to easily handle the answers log.
Using "Windows Explorer", opens the "Documents" folder; selects the "paperclickers_AnswersLog.csv" file with the mouse and open using double click; starts "Microsoft Excel" until the answers log appears	Using a computer where you have, for instance, *Microsoft Excel* installed, simply double-click on the downloaded file and it will be opened as a spreadsheet; in that spreadsheet, each line will correspond to a question you asked using paperclickers to capture the answers of your students.
Using "Excel" with the answers log opened, selects the "SEQ" column; then selects the "TIMESTAMP" column; then selects the columns starting from the fourth onward, containing the individual answers	The first column, identified as "SEQ", is the question number you made in the same application session; the second column, "TIMESTAMP", indicates the date and time you captured the corresponding answers; starting from the fourth column you will have each of the responses detected for the answering cards codes 1 to 99; empty cells will indicate the absence of response — most probably those students were not in that class.
Using "Excel" with the answers log opened, selects the "QUESTION" column	The third column of the answers log, the one named "QUESTION", corresponds to a usage option disabled by default in "paperclickers", but you might want to activate to choose entering a short text to identify each question you ask for your students, when using paperclickers to collect the answers; that will allow you to record in the answers log a short text which will help you recall the question you had asked, facilitating your later usage of the answers log.
Enters the paperclickers "Settings" option; selects the "Enter questions text for log?" option; sets "Yes" on the dialog	To enable that text entry option, go to *Settings* on paperclickers and answer *Yes* to the *Enter questions text for log?* option.
Shows paperclickers initial screen; selects "start" button; shows "Question for the class" screen; enters text "Global warming causes"	That way, whenever you ask a question for your students, a screen will appear for you to write something that helps you to identify the *Question for the class* you will be giving; the text you type on that screen will be saved under the *QUESTION* column inside the answers log.

... This script continues in the next page

Continuation – Paperclickers training video 5 – Sharing and using the answers log	
Video description	*Text script*
Opens the answers log on “Excel”, showing now the same “QUESTION” previously entered	Check how that makes it much easier for you to use the answers log to analyze your class’s performance.
Animation showing the scanning of students answers and the creation of an entry on the answers log; shows then the answers log entry with the initial 30 answers, from codes 1 to 30, indicating the missing answers correspond to absent students	Depending on your usage of paperclickers, it can be a tool to help you control your students presence on your classes: if on every class you ask a question and use paperclickers to capture the answers, you will have inside the answers log the identification of all the answers given, which will correspond to the record of which students were present in that class. For example: if you have 30 students in your class and for a given question you have only 27 answers recorded in the answers log, answering cards codes with missing answers probably indicates those students — the owners of those answering card code — were not present in that class.
Shows the paperclickers scanning screen, with students using their answering cards; shows the thumbnail of the referred video	However, to use paperclickers as students presence record tool, it is very important that you always make sure all responses have been captured during the scanning process. See the *How to effectively capture the students’ answers* video for instructions on how to ensure effective capture of students answers using paperclickers.

Table 17 – Script for the paperclickers training video about how to access and use the answers log to follow the students performance.

Paperclickers training video 6 – Using paperclickers in several classes	
Video description	*Text script*
Title: “Using paperclickers in several classes”	
Animation showing 3 different groups of students, representing 3 different classes, and the doubt about how to distribute the answering cards	If you want to use paperclickers with multiple classes, you can choose how you will distribute the answering cards to all your students; depending on the total number of students, there will be two possibilities.
	If you have a total of up to 99 students considering all of your classes, you can assign a unique answering card to each one of your students. This might be interesting, if you want to always have an easy way to analyze the performance of all your students at the same time, regardless of their classes, once the answers log stores the answers from all answering cards codes from 1 to 99, for each question.
Animation of 28 grouped icons, representing the students of the first class; after another 26 grouped icons appear, identifying the second class; underneath each group shows the text “Answering cards 1 to 28” and “Answering cards 29 to 54”, respectively	So if you use paperclickers with two classes, one with 28 students and another with 26, you can distribute the answering cards codes from 1 to 28 for the first class and 29 to 54 for the second class.
Shows the answering log; selects the columns representing the answers of the answering cards codes 1 until 28; then selects the columns corresponding to the answers of codes 29 until 54	Therefore, in the answer log, you will know that questions with answers for answering cards codes 1 through 28 are from the first class, and 29 to 54 are from the second class.
Students list, alphabetically ordered; animation indicating the first name corresponds to the answering card code 29 and so on	You will need to know that for the second class, the answering cards codes begin at 29, in order to know which answering card code is with which student, using the students’ list alphabetically ordered.

... This script continues in the next page

Continuation – Paperclickers training video 6 – Using paperclickers in several classes	
Video description	*Text script*
Shows again the animation of 28 grouped icons, representing the students of the first class; after another 26 grouped icons appear, identifying the second class; after a group of 4 new icons appears, representing the additional answering cards	To use this form of code assignment for multiple classes, you should set the total number of students, inside the paperclickers *Settings*, to be the sum of the number of students in both classes; you could include additional 4 or 5 answering cards for eventualities, such loss of cards or special presence in your classes. In the example, the total number of students to define — corresponding to the total is number answering cards — would be 58.
Animation with the answering cards being printed, creating a single stack; after the stack is partially separated, creating a second stack; text appears underneath each stack showing "Class 1 – Answering cards codes 1 to 28" and "Class 2 – Answering cards codes 29 to 54"	Once the answering cards are printed, you must distribute them according to the assignment to each class; in the example, the students in the first class receive the answering cards codes from 1 to 28, and the students in the second group receive the answering cards codes from 29 to 54.
Shows the referred videos thumbnails	Watch again the *Installation and initial execution* and the *Printing students' answering cards* videos if you have questions about how to set the number of students or how to print the codes.
Shows again the animation of 28 grouped icons, representing the students of the first class; after, another 26 grouped icons appear, identifying the second class; underneath each group shows the text "Answering cards 1 to 28" and "Answering cards 1 to 26", respectively	Another way to distribute the answering cards among your classes is to consider that each class always starts with the answering card code 1. Thus, in the same example above, the first class would use the answering cards codes from 1 to 28 and the second the answering cards codes from 1 to 26. In that way, it is always easy to associate a student from each class with her answering card code, using students list alphabetically ordered.
Shows the answering log with some entries with answers for answering cards codes 1 to 28 and other entries with answers to answering cards codes 1 to 26; animation identifying each entry either with "Class 1" or "Class 2", depending on the "TIMESTAMP" column value	However, it will require extra care to use the answering log, in order to know which class the entries corresponds to: for that you will need to use the *TIMESTAMP* column to correctly identify which class you asked the registered question. This way of assigning response codes is recommended in the case you have multiple classes and the total number of students exceeds the 99 paperclickers answering cards.
Shows again the animation of 28 grouped icons, representing the students of the first class; after another 26 grouped icons appear, identifying the second class; underneath each group shows the text "28 students" and "26 students"; after, another 4 grouped icons appears, representing the additional answering cards; shows the total of "32 students"	To assign the answering cards in this way, you must set the number of students for paperclickers to be the total number of your largest class; in the previous example, it should be 28 students, which may be supplemented by 4 or 5 additional answering cards to cover losses or special participation in your classes. Therefore, you must set 32 as the total of students to be handled by the paperclickers.
Shows the referred video thumbnail	Watch again the *Installation and initial execution* video if there are any questions on how to do define the number of students in paperclickers.
Shows an animation of the answering cards being printed and creating a single stack; after, shows underneath the stack the text "Class 1 – Answering cards codes 1 to 28"; then repeats the animation of the answering cards being printed, now creating a second stack; after writes underneath the second stack the text "Class 2 – Answering cards codes 1 to 26"	Once you have set the total number of students, you should print the answering cards sets several times, since you should distribute sequences of similar answering cards codes for each class.
Shows the referred video thumbnail	Review the video *Printing students' answering cards* if you have questions about how to print them, and repeat the final steps of printing until you have completed the required answering cards sets. In the example, you should print 2 times the generated answering cards set.

Table 18 – Script for the training video on how to use paperclickers on several classes.

Paperclickers training video 7 – How to effectively capture the students' answers	
Video description	*Text script*
Title: "How to effectively capture the students' answers"	
	When you capture your students' responses using paperclickers, there are a few things you can do to improve the answering cards recognition, making the whole process faster and more reliable.
Shows students in the classroom, presenting their answering cards; take on problematic situations, like answering cards overlapping each other, answering cards partially occluded by other students bodies, students holding the answering cards partially covering their front; show answering card back and then its front	The first tip is to ask students to present their answering cards making sure they are well visible, holding them in the proper area, indicated on the back of the answering cards, to avoid covering the area of the response code printed on the front — the code is the black circles.
Paperclickers scanning screen, with the smartphone too close (1 meter distance) an A5-sized answering card with folding marks, showing the answering card is not properly recognized; the smartphone gets farther from the answering card and then the paperclickers properly recognizes it	Whenever you start capturing the answers, do not get too close to the students; the best performance is from a distance of at least 2 meters between the camera of your phone and the student's answering cards, considering the codes printed in the half-page size — corresponding to the 2-per-page impression. If you are at a smaller distance, any defect or imperfection on the answering card — for example a folding marks or dirt — will make it harder to recognize.
Presentation of the 3 different answering cards sizes; all three starts stacked, with the bigger (A4) below and the smaller (A6) on top, all aligned by the left lower corner; the A6-sized card slides until the right side of the screen; the A5-sized cards slides to the right, staying between the A4 and A6 cards — from left to right the cards are ordered A4, A5 and A6 sizes; a text shows under each card: "A4 ~= 11 m; A5 ~= 8 m; A6 ~= 5 m"	On the other hand, also remember that there is a maximum distance limit for the recognition of the answering cards, and that distance varies according to size you have printed them. If your room is very large, and you normally stays at more than 10 meters away from the student farther away, you should print response codes in the size of 1 per page for good detection performance.
Shows the paperclickers scanning screen, with the camera facing the answering cards in a parallel position, where the TopCodes circles are almost perfect; shows an animation indicating that is the correct usage; shows once again the paperclickers scanning screen, now with the camera capturing the answering cards from an inclined position, where the TopCodes appears like an ellipse; shows an animation indicating that is the incorrect usage	When capturing the students answers, always try to stay right in front of the students, facing the camera directly towards the answering cards — you should see the codes as circles, not ellipses; move around the front of the room, preventing to capture the answering cards from their side.
Starts paperclickers answers scan from initial screen, showing the students still selecting their answers, starting to hold up their answering cards; animation showing that is the incorrect procedure; shows paperclickers scanning screen, partially capturing the students answers but interrupting, turning the camera down, with the scanning screen active; animation indicating that is an interruption on the scanning process; another animation showing that is the incorrect procedure	Finally, during each capture session, avoid spending too much time on the paperclickers scanning screen, with your phone's camera turned on: that will make the answering cards recognition slower, and will also quickly exhaust your phone battery. Enter the capture screen only after all your students have already raised their answering cards; once on the scanning screen, try to detect all response codes without interruption — for instance, to answer some last-minute questions.
Paperclickers scanning screen, partially captured the students answers; halts the scanning and goes to the detailed answers screen, showing missing answers; shows an animation indicating that is an interruption; returns to the scanning screen and capture the missing answers; animation indicating the scanning process is resumed; finishes the scanning process and shows the detailed answers screen, with all the answers	If you need to stop the answers capture process by any reason, split the capture process in more than one step, going to the detailed answers screen to handle the interruption; there you can check the partial answers capture result, and you can return to the scanning screen to complete capturing the missing codes: paperclickers will recognize you want to resume the capture process if you return from the detailed answers screen, without starting a new question.

Table 19 – Script for the paperclickers training video on how to effectively capture the students' answers.

5.3.2 Peer Instruction basic training videos

Since we also intend to provide the minimum knowledge for the teachers to apply Peer Instruction in their classes effectively, we considered a second set of training videos specifically focusing the methodology. Those videos would address the following aspects of the deployment of a technological pedagogical tool:

- To answer questions of MIT framework related to the **Comfort** and **Openness to Change** sub-areas of **Teachers** component, and the **Pedagogy** sub-area of **Learning** component, providing details on how to employ a proven pedagogical methodology associated with paperclickers.
- To deal with the **expertise** and also on the **vision** axes of the Four in Balance model, since we try to make the teachers knowledgeable about the use of paperclickers with a proven pedagogical methodology, which implies the effort to use active learning on their classes.
- To deal with the **performance expectancy** and the **effort expectancy** constructs of the UTAUT, aiming to enable the teachers to be confident about how to efficiently applying Peer Instruction with paperclickers help.
- To improve **Pedagogical Knowledge (PK)** and **Technological Pedagogical Knowledge (TPK)** of TPACK model, once we provide information about a pedagogical methodology associated with a technological tool.

Peer Instruction is a simple teaching practice, which aims to stimulate the students to make sense of the information they received in a traditional lecture. It allows the students to actively practice reasoning, speaking and sustaining their opinions based on the content information they received. The Peer Instruction dynamics, as presented on Section 2.3, can be implemented in different levels on a traditional lecture format: teachers can choose to ask a single question or plan an entire class or even course using PI.

All those PI aspects, which made it a perfect candidate for our effort of efficiently delivering a CRS, need to be presented and clarified on the training material. Therefore, the videos for training teachers for that methodology usage along with paperclickers should:

1. **Motivate the teachers to use Peer Instruction**, presenting its rationale and how it can effectively promote learning gains, working on the teachers' motivation — according to Koh and Divaharan [2011], it is the first aspect to be tackled when introducing new teaching methods and tools.

2. **Present the steps of Peer Instruction activity**, preferably using a real setup scenario, emphasizing the goal of each step in the methodology.
3. **Talk about the common doubts and difficulties on using Peer Instruction**, in order to reduce the psychological barrier related to the **performance expectancy** concerns.
4. **Suggest how to transition from a traditional lecture to the Peer Instruction usage**, providing guidance on how to create the class material, one of the most important aspects of using PI.

From the accumulated experience of using PI, Crouch et al. [2007] indicate several difficulties and doubts which appear when transitioning to the methodology; warning the paperclickers users about the following points would be a way of dealing with the insecurity of leaving the comfort zone of the traditional lectures:

- **The classes should be prepared to cover less content**, since the focus would not be information transference, but making the students engage with the content actively. The question posing and the peers discussion takes time, but according to Crouch et al. [2007], *"...students develop complex reasoning skills most effectively when actively engaged with the material they are studying..."* and *"...cooperative activities are an excellent way to engage students effectively"*.
- **The classes are more dynamic**, and that leads to the instructor having to improvise more often [Crouch et al., 2007], due to the additional participation added by PI.
- **Skepticism from students**, once they also feel the change in the class dynamics. Properly motivating the students is also a work to be done, and should include talking about the reasons for teaching in that new way.

Up to the present research point, we have been only able to create a list of the videos we would like to have for this specific training about PI, without starting to write their scripts. Since we have defined the initial target audience, those videos should be specifically designed to approach their experiences and needs; ideally, to better tailor the training material, teachers of the target audience should be consulted, as well as teachers with PI usage expertise.

Table 20 presents the videos we have considered to create for Peer Instruction training, indicating their main subjects and rationale. That list needs to be refined and developed in future work, and the suggested videos might be split to comply with the short video requirement. The videos will point and introduce a specific guideline provided along the training material — discussed on the next section (5.3.3) —, focused on the step-by-step tips about how to create a Peer Instruction class material, starting from traditional lecture material. We believe that would be the most common scenario for the teachers of our target audience.

Peer Instruction with paperclickers training video 1 – What is Peer Instruction and why it works

This first video on the Peer Instruction with paperclickers training series will have the goal of presenting the teaching method, clarifying its basic structure of a question based teaching methodology, as well as its major differential: foster the students active engagement in the class through the peer discussion step, when they have the opportunity to expose and to defend their point of views, stimulated by the questions posed by the teacher.

This video major goal is to motivate the teachers to learn how to use Peer Instruction in their classes, trying to convince them about the methodology effectiveness. Eric Mazur's "Confessions of a Converted Lecturer" talk [a] — when the PI creator expresses some of the ideas also captured in Mazur [1997] — can be a good inspiration for this video script, since his goal is exactly motivating for PI usage.

Peer Instruction with paperclickers training video 2 – How to use Peer Instruction in your class

The second video on this training series will depict the Peer Instruction process, briefly describing each step, as presented in Section 2.3; this video should also add some rationale for each of those steps and some best practices — for instance, the threshold for repeating the process on the same topic — as discussed by Vickrey et al. [2015] and Crouch and Mazur [2001].

Ideally, this video should present a real PI use case, considering the target audience of Brazilian public high school teachers, especially of STEM subjects.

Peer Instruction with paperclickers training video 3 – Why choosing the right questions improves the learning gains with Peer Instruction

In this third video, we will emphasize the importance of creating proper questions to use during a PI class, since they need to explore the concepts on the presented subject, aiming at common students misunderstandings, being also able to foster the discussion among the students, extracting the most from the peer discussion step.

This video should clarify that the question choice would directly affect the students learning gains, especially on the peer discussion step, but also through the quality of the feedback exchanged among teachers and students: too simple questions, not linked to the main concepts worked in the class, might not provide meaningful feedback regarding students understanding and content absorption; too hard or long questions might also miss the point, since the students might not be able to work on them during the methodology timeframe.

Hence, this video should present some basic guidelines on how to create those questions, stating the main requirements they should meet, and pointing to a more detailed document, which would suggest a working guideline to transform a conventional lecture class material into PI. This video should use the experience gathered on PI usage [Crouch and Mazur, 2001] and also on other question based methodologies [Beatty, 2005].

Peer Instruction with paperclickers training video 4 – Common concerns when moving to Peer Instruction

The final video envisioned for this training series should address the common doubts and difficulties on implementing Peer Instruction classes, as identified by the current experience — once again on works like Crouch and Mazur [2001]. The idea is to briefly discuss each one of those issues, recognizing they represent real concerns which can affect the PI implementation, but they can be handled, allowing the teachers to gain confidence and become comfortable on using PI on their classes.

This video should work issues like the students' resistance to new teaching methods, the coverage of a smaller syllabus during the classes, the specific difficulties on preparing the PI classes creating the right conceptual questions.

Table 20 – List of suggested videos to be created in order to provide training on how to employ Peer Instruction in the classes, using paperclickers as a facilitating tool

[a] https://www.youtube.com/watch?v=8UJRNRdgyvE&feature=youtu.be

5.3.3 Peer Instruction material creation guidelines

Probably the greater barrier to implementing Peer Instruction is the creation of the questions to be used in the process: posing questions are central to Peer Instruction, and choosing the right ones for the specific content can be challenging, especially aiming to explore misconceptions or provoke discussion among the students. Beatty [2005] proposed what he called the *question-driven instruction* methodology, which shares with Peer Instruction the same dependency on the type of the questions used: *"...the fundamental rule is to ask question that cannot be answered without exercising the desired habits of mind and to avoid excess baggage that might distract students from the need to exercise them"*. In fact, Vickrey et al. [2015] meta-analysis pointed out students benefited more when conceptual questions were applied in PI.

The type of questions also directly affects the students' participation in the discussions: an effective PI question should leave room for disagreement, to foster the crucial part of the peer discussion. Crouch et al. [2007] indicated the following general criteria a good PI question — named **ConcepTest** by the methodology creator — should meet:

- The question should focus on a single important concept; ideally, it should correspond to a typical student difficulty — uncovering a misconception, or verifying the proper understanding of an important concept, should be the best result of a PI round;
- It should also require thought, not just plugging numbers into equations, or simple memorization;
- There should be plausible incorrect answers, in order to foster discussion and students exploring their reasoning when trying to convince their peers;
- The question should be unambiguously worded, since the main focus should be the concept covered;
- And finally, the question should be neither too easy nor too difficult — being too easy would defeat its previous propositions; being too hard, they would not fit into the expected class dynamics.

We proposed a document containing guidelines for the creation of a Peer Instruction class, starting from a regular lecture class material. With that, we believe we can reduce the related psychological adoption barriers, since the teachers would work over a material they are already familiar with, and could make the transition gradually, choosing to apply PI only for selected topics they deemed appropriate and feel comfortable.

We based an initial version of those guidelines on the PI creators experience [Crouch and Mazur, 2001; Crouch et al., 2007], and also in the Vickrey et al. [2015] extensive meta-analysis of PI implementation research. We also included some hints from the *question-driven instruction*, as devised by Beatty [2005], and some information on how to create effective multiple-choice questions — Medeiros [1975] and Brame [2013] —, since that is the most straightforward question model for PI when using CRS.

The complete guideline was created in Brazilian Portuguese language (reproduced on appendix B), once again considering our initial target audience; it should be used along with the video tutorial about the work to transform into PI, a regular lecture class material (see Section 5.3.2). The guideline is organized in three different sessions, addressing:

1. **General information on how to start structuring the PI class material**: In this session there are general recommendations like: think about the concepts to be covered; devise questions to explore those concepts using common students misconceptions and difficulties; always seek for balance between difficulty and easiness; use the traditional lecture class material as starting point; plan to cover fewer content, considering you can apply PI to some of them.
2. **Suggestions on different techniques to create the questions**: This second session would address recommendations for the questions' structure: always remember the question goal of motivating discussion; write clear questions, focusing on the selected concept; seek for questions requiring interpretation of representations; use restrictions on the question to focus the attention; create questions enabling multiple solutions and ask for the best; use questions requiring only the creation of a solution strategy, not the complete final solution.
3. **Suggestions for creating effective multiple-choice questions**: This final session would include recommendations on how to create the multiple-choice answers for a given question: start with the correct answer, once again seeking for clarity; whenever possible, use the wrong alternatives to explore common misconceptions or difficulties, using your previous classes experience; avoid obviously wrong answers; avoid clues on the correct answer, keeping language uniformity among the alternatives; create mutually exclusive answer alternatives.

This guideline needs to be completed with some real case examples of how to transform traditional lectures in PI class materials, especially considering the knowledge fields and specific classes subjects of our target audience: those examples would provide a valuable starting point for the teachers to work on their own classes materials, once they would be closer and easily transposed to their specific realities.

5.4 Providing training material might increase the delivery effectiveness

New technologies alone cannot provoke real learning gains: new technological pedagogical tools are only effective when combined with proven pedagogical methodologies. We associated paperclickers with Peer Instruction, recognizing that methodology, although proven in its effectiveness and also being two decades old, is still unknown and most teachers continue to rely on traditional lectures for their classes. Supported by the literature (explored in chapter 2), we considered that a training material on PI and paperclickers would need to be included in the technology delivery package, to achieve an effective usage.

We embedded usage instructions in paperclickers mobile application aiming to decrease the knowledge required for the technology initial use. We proposed the creation of training materials to decrease the overall infrastructural barriers, but also the psychological barriers involved in successfully using paperclickers and Peer Instruction. We presented the scripts of a video sequence providing information on how to effectively use paperclickers and its main features. We also presented the basic content of a second training video sequence, aiming to clarify how to start using Peer Instruction, including a textual guideline on how to prepare PI classes, starting from traditional lectures material.

Our work still needs completion, to develop the designed training videos and guidelines. Furthermore, the effectiveness of the training material has to be validated with the target audience, verifying it can successfully promote the adoption of the technological pedagogical tool and its associated methodology. Even the adequacy of the training material choice — training videos — has to be validated with Brazilian public high school teachers, our target audience. Once again, the diversity of environmental conditions and experiences greatly increases the challenges of this following required research.

Although still incomplete, we believe the suggested training material advances the paperclickers research towards increasing the probability of its effective delivery.

6 Conclusion

Making available a low-cost classroom response system is only part of what is needed to foster active learning in disfavored communities. It is critical to address teachers' and instructors' concerns related to not only adopting the new technology but also employing new teaching methods, especially when that represents leaving behind the safety, predictability, and control of a lecture classroom setup [Beatty, 2005].

We studied the effort of looking for the effective delivery of a new technological pedagogical tool, in order to achieve the social impact intended for it — with paperclickers, we pursued broadening the adoption of Peer Instruction, an active learning methodology. We explored the technical aspects in order to enhance the overall usability, interpreting the user experiments executed in previous work [Bindá, 2015], designing and implementing the corresponding changes, also including usage guidelines within the tool itself. We investigated additional limitations, resulted from employing the answering cards detection technology. The resulting application has been released as an open-source solution, along with some of the user experiments material, both available for the public at large. All those efforts aimed at lowering any usage barrier created by the difficulties of using the technology.

We also explored other adoption barriers, documented in the literature, resulting from psychological aspects of the target audience — like the perceived usefulness, the expected benefits, the effort and support required to use it —, provoked by not only new technologies, but also the related pedagogical methodology they might imply or suggest. To address those issues, we presented the initial design of training material, focusing on a specific target audience — Mathematics and Physics teachers of public high schools in Brazil.

Throughout this work, we collected in vast literature the grounds for a multidisciplinary approach for effective delivery of a new technological pedagogical tool, providing some details on that pursuit in a research team aiming social impact on developing areas.

6.1 Some conclusions from paperclickers investigation

From the research process of paperclickers enhancements, we drew some conclusions we believe represent some practical recommendations, especially useful for teams working on technological pedagogical tools[1]. The major challenge throughout development was ensuring a robust detection and decoding of the students' cards. Image pro-

[1] This section is reproduced from Oliveira et al. [2017]

cessing for a large number of cards in the uncontrolled environment of a classroom, while targeting low-cost computational device, proved technically challenging. Although TopCodes are very robust to distortions and noise, we had to include adaptations to transpose them from their original application context (tangible programming environment) to ours (CRS).

On the usability tests, the recording of the users' interaction with the application — including their "think-aloud" comments and recommendations — was the strategy that provided the most actionable information. The unstructured interviews were also interesting, but, to our surprise, we found the structured, formal survey the least useful of the instruments — it only provided enough information to reinforce trends we had already understood — with more confidence — in the recordings and interviews. We believe that a survey has to be exceptionally well-designed to provide actionable information, while interviews and recording can be useful even for developers without a huge background in Human-Computer Interaction. In future projects, we will attempt to apply heuristic evaluation [Nielsen and Molich, 1990], before experimenting with real users — we believe that cost-effective technique would have anticipated some of the problems found in our user trials.

Relying on storyboards for design and documentation worked very well for a small team, designing a small-sized (less than 10-screen workflow) user-interaction driven application. Our research group comprised five people, partially changing throughout the project — a scenario not uncommon on academic research. We employed storyboards to elicit and document the requirements, to sketch the interaction elements, to design the navigation and dynamics of the application, etc. We also used them to image usage scenarios, which were also crucial to design the usability tests.

6.2 Next steps and future research

This work added to paperclickers research the discussion towards achieving social impact, through the deployment of a new technological pedagogical tool. We have released an improved version of paperclickers, but there still are a lot of work and studies to be completed; we list the following next steps and future research:

1. The video tutorials need to be completely designed and developed, including the contact with teachers of the target audience and with Peer Instruction usage experience.

2. The video tutorials usage needs to be analyzed: are they effective to provide basic knowledge and motivation for applying Peer Instruction using paperclickers inside Brazilian public high schools?

3. The clarity and effectiveness of the PI material creation guidelines need to be verified, on the same target audience: is it enough to guide teachers to build PI material from regular lecture class materials?

4. The final paperclickers user experience should be evaluated, also with its target audience. As mentioned throughout this work, that research will face significant challenges due to the great diversity of Brazilian public high schools, probably imposing a fractionated approach or involving multiple research teams.

5. Investigate the possibility and value of embedding into paperclickers the pedagogical methodology — for instance, analyzing how to integrate Peer Instruction process into the scanning procedure.

6. Study paperclickers research to devise a model for effective technological pedagogical tools creation and deployment.

6.2.1 The need for a research program

The probability of achieving social impact through academic research increases a lot if a research program is developed towards a specific goal. Our results in paperclickers are the combination of several studies and works developed within our research group, all regarding the same subject of creating technological pedagogical tools, which can be effectively used by the target audience.

The knowledge and research required to achieve social impact through a new technological pedagogical tool include several knowledge areas, and largely surpasses the time frame of a single academic research. As already stated by other works [Shneiderman, 2016], combining researchers from different areas and backgrounds might be the requirement to truly achieve the social impact with a technological pedagogical tool.

6.3 Towards the effective delivery of a technological pedagogical tool

A new technological pedagogical tool will only be effectively delivered, when it is successfully used by the targeted users. For that to happen, the infrastructural requirements associated with the technology employed need to be covered; in some realities, that might represent a definitive barrier. Reducing those infrastructural requirements is crucial to increase the tool usage.

However, human restrictions also impose an equally categorical barrier, which might be even harder to tackle, since they are subtle and specific for a given technology — and the associated pedagogical methodology. It is essential to gain teachers confidence and motivation to leave the comfort of their known tools and processes; the pursuit of effective technological pedagogical tool delivery should include research, most probably multidisciplinary, on how to decrease those human restrictions.

The investigation of paperclickers effective deployment not only illustrates the challenges of creating a classroom response system with the lowest adoption barriers — analyzed from both the infrastructural and psychological perspectives —, but it inspires further researches on effective technological pedagogical tools investigations.

Bibliography

Almeida, M. E. B. d. and Valente, J. A. Políticas de Tecnologia na Educação Brasileira. Technical report, Centro de Inovação para Educação Brasileira - CIEB, São Paulo, Brasil, november 2016. Last accessed on 2017-12-22. Cited 4 times in pages 16, 18, 29, and 30.

Amy, N. R. and Amy, S. R. Optical polling platform methods, apparatuses and media, 08 2015. URL `https://www.google.com/patents/US9098731`. Cited in page 22.

Archambault, L. M. and Barnett, J. H. Revisiting technological pedagogical content knowledge: Exploring the tpack framework. *Computers & Education*, 55(4):1656–1662, 2010. Cited in page 34.

Beatty, I. Transforming student learning with classroom communication systems. Technical report, EDUCAUSE Center for Applied Research, 2005. Cited 4 times in pages 20, 93, 94, and 97.

Bindá, J. M. Affordable solutions for classroom response systems. Master's thesis, School of Electrical and Computer Engineering at the University of Campinas — UNICAMP, Campinas, São Paulo, Brazil, 7 2015. Cited 8 times in pages 7, 16, 17, 38, 39, 41, 76, and 97.

Bødker, S. Third-wave hci, 10 years later—participation and sharing. *interactions*, 22(5): 24–31, 2015. Cited in page 16.

Brame, C. J. Writing Good Multiple Choice Test Questions, 2013. URL `https://cft.vanderbilt.edu/guides-sub-pages/writing-good-multiple-choice-test-questions/`. Last accessed on 2018-01-08. Cited in page 94.

Celeri, V. V., Mendes, P. A., and Valle Jr, E. Programação No Ensino Médio Com Auxílio da Plataforma App Inventor. In *XXV Congresso de Iniciação Científica da Unicamp*. Pró-Reitoria de Pesquisa, UNICAMP, 2017. developed under Programa Institucional de Bolsas de Iniciação Científica e Tecnológica (PIBIC). Cited in page 80.

Cross, A., Cutrell, E., and Thies, W. Low-cost audience polling using computer vision. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 45–54. ACM, 2012. Cited 2 times in pages 22 and 40.

Crouch, C. H. and Mazur, E. Peer instruction: Ten years of experience and results. *American journal of physics*, 69(9):970–977, 2001. Cited 5 times in pages 18, 24, 25, 93, and 94.

Crouch, C. H., Watkins, J., Fagen, A. P., and Mazur, E. Peer instruction: Engaging students one-on-one, all at once. *Research-Based Reform of University Physics*, 1(1): 40–95, 2007. Cited 2 times in pages 92 and 94.

Eagan, M. K., Stolzenberg, E. B., Berdan Lozano, J., Aragon, M. C., Suchard, M. R., and Hurtado, S. *Undergraduate teaching faculty: The 2013–2014 HERI Faculty Survey*. Higher Education Research Institute, UCLA, Los Angeles, 2014. Cited in page 17.

Felder, R. M. and Brent, R. Active learning: An introduction. *ASQ Higher Education Brief*, 2(4):1–5, 2009. Cited in page 16.

Fiala, M. Artag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 590–596. IEEE, 2005. Cited in page 22.

Gain, J. Using poll sheets and computer vision as an inexpensive alternative to clickers. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pages 60–63. ACM, 2013. Cited in page 23.

Graham, C. R. Theoretical considerations for understanding technological pedagogical content knowledge (tpack). *Computers & Education*, 57(3):1953–1960, 2011. Cited in page 34.

Guo, P. J., Kim, J., and Rubin, R. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014. Cited 2 times in pages 79 and 80.

Hao, Y. and Lee, K. S. Teaching in flipped classrooms: Exploring pre-service teachers' concerns. *Computers in Human Behavior*, 57:250–260, 2016. Cited in page 16.

Harley, A. Instructional overlays and coach marks for mobile apps, February 2014. URL `https://www.nngroup.com/articles/mobile-instructional-overlay/`. Last accessed on 2018-01-08. Cited in page 77.

Hattie, J. A. C. *Visible learning: a synthesis of meta-analyses relating to achievement*, chapter 9, pages 161–183. Routledge, 2009. Cited 2 times in pages 17 and 23.

Horn, M. Topcode: Tangible object placement codes. *Retrieved from http://users.eecs.northwestern.edu/~mhorn/topcodes*, 2012. Cited 2 times in pages 40 and 58.

Horn, M. S. and Jacob, R. J. Designing tangible programming languages for classroom use. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 159–162. ACM, 2007. Cited in page 41.

Horn, M. S., Solovey, E. T., Crouser, R. J., and Jacob, R. J. Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 975–984. ACM, 2009. Cited 2 times in pages 8 and 53.

Hunsu, N. J., Adesope, O., and Bayly, D. J. A meta-analysis of the efects of audience response systems (clicker-based technologies) on cognition and affect. *Computers & Education*, 94(March 2016):102–119, 2016. Cited 3 times in pages 20, 21, and 46.

INEP. Overview of the Brazilian Education System. Technical report, Anisio Teixeira National Institute For Educational Researsh and Studies (INEP), Brasilia, Brazil, October 2016. Cited 2 times in pages 9 and 80.

Ito, M. and Miura, M. Portable vision-based response analyzer with sheet bending recognition. In *Consumer Electronics (GCCE), 2015 IEEE 4th Global Conference on*, pages 143–144. IEEE, 2015. Cited in page 23.

Jamieson-Proctor, R., Albion, P., Finger, G., Cavanagh, R., Fitzgerald, R., Bond, T., and Grimbeek, P. Development of the ttf tpack survey instrument. *Australian Educational Computing*, 27(3):26–35, 2013. Cited 3 times in pages 10, 35, and 36.

KENNISNET. Four in balance monitor 2015. : Kennisnet, 2015. Technical report, Kennisnet, Zoetermeer, Netherlands, 2015. Cited 2 times in pages 18 and 30.

Koehler, M. J. and Mishra, P. What is technological pedagogical content knowledge. *Contemporary issues in technology and teacher education*, 9(1):60–70, 2009. Cited 4 times in pages 7, 18, 32, and 33.

Koh, J. H. and Divaharan, H. Developing pre-service teachers' technology integration expertise through the tpack-developing instructional model. *Journal of Educational Computing Research*, 44(1):35–58, 2011. Cited 4 times in pages 7, 34, 35, and 91.

Lane, D. and Atlas, R. The networked classroom. In *meeting of Computers and Psychology*, 1996. Cited in page 20.

Mazur, E. Peer instruction: getting students to think in class. In *AIP Conference Proceedings*, pages 981–988. IOP INSTITUTE OF PHYSICS PUBLISHING LTD, 1997. Cited 5 times in pages 17, 18, 23, 24, and 93.

Medeiros, E. B. *Provas objetivas: técnicas de construção*. Editora da Fundação Getulio Vargas, 1975. Cited in page 94.

Miura, M. and Nakada, T. Device-free personal response system based on fiducial markers. In *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on*, pages 87–91. IEEE, 2012. Cited in page 22.

Neto, W. J. P. Testes formais de usabilidade para a interface com o usuário projeto paperclickers. B.s. thesis, School of Electrical and Computer Engineering at the University of Campinas — UNICAMP, Campinas, São Paulo, Brazil, 6 2015. Cited 2 times in pages 17 and 38.

Neuville, S., Frenay, M., and Bourgeois, E. Task value, self-efficacy and goal orientations: Impact on self-regulated learning, choice and performance among university students. *Psychologica Belgica*, 47(1), 2007. Cited in page 35.

Nielsen, J. Progressive disclosure, December 2006. URL `https://www.nngroup.com/articles/progressive-disclosure/`. Last accessed on 2018-01-08. Cited in page 77.

Nielsen, J. and Molich, R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM, 1990. Cited 2 times in pages 48 and 98.

OECD. Brazil, Country Note - Results from Programme for International Student Assessment (PISA) 2015. Technical report, Organization for Economic Co-operation and Development (OECD), Paris, France, 2016. Last accessed on 2017-12-11. Cited in page 80.

Oliveira, E., Bindá, J., Lopes, R., and Valle, E. Paperclickers: Affordable solution for classroom response systems. *CoRR*, abs/1710.02763, 2017. Cited 5 times in pages 10, 19, 22, 41, and 97.

Osterweil, S., Shah, P., Allen, S., Groff, J., Kodidala, S. P., and Schoenfeld, I. Full report: A framework for evaluating appropriateness of educational technology use in global development programs. Technical report, The Massachusetts Institute of Technology, Cambridge, Massachusetts & The Indian Institute of Management, Ahmedabad, India, Comprehensive Initiative on Technology Evaluation at MIT, Department of Urban Studies and Planning Building, Cambridge, Massachusetts, US, 2016. Cited 5 times in pages 10, 18, 27, 28, and 29.

Ribeiro, V., Bindá, J. M., Lopes, R. R., and Valle Jr, E. Appliance para sistemas de resposta a audiência em dispositivos móveis. B.s. thesis, School of Electrical and Computer Engineering at the University of Campinas — UNICAMP, 2015. developed under Programa Institucional de Bolsas de Iniciação Científica e Tecnológica (PIBIT). Cited 2 times in pages 17 and 38.

Ries, E. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, chapter 3. Crown Publishing Group, 2011. Cited in page 45.

Shneiderman, B. *The new ABCs of research: Achieving breakthrough collaborations*. Oxford University Press, 2016. Cited in page 99.

Smith, D. J. and Valentine, T. The use and perceived effectiveness of instructional practices in two-year technical colleges. *Journal on Excellence in College Teaching*, 23(1): 133–161, 2012. Cited in page 17.

Tejada, L. M. M. Interatividade de chamada e realidade aumentada no projeto paperclickers. B.s. thesis, School of Electrical and Computer Engineering at the University of Campinas — UNICAMP, Campinas, São Paulo, Brazil, 6 2014. Cited 2 times in pages 17 and 38.

Urban-Woldron, H. Integrating technology into pre-service physics teachers' pedagogical content knowledge. *ESERA Science Education Research Series, Strand 13 - Pre-service science teacher education*, 2011. Cited 2 times in pages 34 and 35.

Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. User acceptance of information technology: Toward a unified view. *MIS quarterly*, pages 425–478, 2003. Cited 4 times in pages 7, 18, 31, and 32.

Vickrey, T., Rosploch, K., Rahmanian, R., Pilarz, M., and Stains, M. Research-based implementation of peer instruction: A literature review. *CBE Life Sciences Education*, 14(1):1–11, 2015. Cited 6 times in pages 7, 18, 25, 26, 93, and 94.

Wellner, P. D. Adaptive thresholding for the digitaldesk. Technical Report EPC-1993-110, Rank Xerox Research Centre, July 1993. Cited in page 55.

Appendix

APPENDIX A – Original version of paperclickers usage training material

The paperclickers usage training video scripts presented in section 5.3.1, were originally created in Brazilian Portuguese language, designed for the selected target audience. Tables 21, 22, 23, 24, 25, 26 and 27 present the original version here for completeness, highlighting the same information has already been presented in the referred section.

Vídeo 1 de treinamento do paperclickers – Apresentação e funcionamento básico	
Descrição do vídeo	*Texto falado*
Título: “O que é paperclickers?”	
Celular em primeiro plano, tela de captura; ao fundo a sala é parcialmente visível com os alunos usando os cartões para resposta.	O paperclickers é uma solução de baixo custo para você coletar rapidamente as respostas de seus alunos em sala de aula: você faz uma pergunta e pede para seus alunos utilizarem os cartões codificados para apresentar as respostas e você utiliza a aplicação no seu celular Android para capturá-las.
No mesmo enquadramento, finaliza captura, apresenta tela de respostas e vai para gráfico	Com isso fica mais fácil ter uma aula dinâmica, com maior participação. E como as respostas ficam registradas, você pode utilizá-las para preparar a aula seguinte, sabendo como foi o desempenho anterior, e até mesmo controlar quem esteve presente naquele dia.
Título: “Como funciona?”	
Sequência de animação mostrando o uso — apresentação da pergunta, pensamento da resposta, manipulação do cartão para escolha da resposta, captura das respostas, verificação do resultado	Você faz uma pergunta múltipla escolha, com até 4 respostas possíveis; os alunos escolhem sua resposta girando o cartão até a orientação correspondente e apresentam o cartão; você coleta e registra as respostas com a aplicação no seu celular e pronto, fica sabendo qual é a opinião da turma, sem precisar ficar contando os braços levantados. Com as respostas você saberá se você precisa trabalhar ainda o assunto com seus alunos, ou se pode seguir em frente.

Table 21 – Original script for the training video presenting paperclickers.

Vídeo 2 de treinamento do paperclickers – Instalação e execução inicial do paperclickers	
Descrição do vídeo	*Texto falado*
Título: “Instalação e execução inicial do paperclickers”	
Gravação da tela com a sequência de uso: acesso playstore, instalação	É muito simples para começar a usar o “paperclickers”; a primeira coisa a fazer é instalar a aplicação. Para não gastar os créditos de seu celular, use uma rede WiFi, entre na loja de aplicativos do Android e busque pelo aplicativo "paperclickers". Assim que encontrar, peça para ser realizada a instalação, que será bem rápida.
Apresenta instalação terminada; usuário solicita execução, aplicação inicia e apresenta a sequência de telas de onboarding	Uma vez instalado, inicie o paperclickers e seja guiado no seu primeiro uso: serão apresentadas telas iniciais explicando as principais funcionalidades do aplicativos os primeiros passos necessários para o seu uso.
	... *Este script continua na próxima página*

Continuação – Vídeo 2 de treinamento do paperclickers – Instalação e execução inicial do paperclickers	
Descrição do vídeo	*Texto falado*
Usuário entra na opção de configuração, seleciona a opção de definição de número de alunos e aparece o campo para entrar o número	A primeira coisa a fazer é configurar o número de alunos com os quais você irá trabalhar: isso pode ser feito dentro da opção configurações. A escolha desse número é importante, pois indicará para a aplicação quais os códigos de resposta serão válidos. A nossa sugestão é que cada um de seus alunos tenha o seu próprio código de resposta, pois com isso você sempre saberá qual resposta que cada aluno deu para cada uma das perguntas que você já fez.
Animação mostrando 30 ícones identificando cada aluno de uma turma; na sequência aparecem mais 4 alunos adicionais, compondo o total de 34 códigos de resposta recomendados	Se você for trabalhar com o paperclickers com apenas uma turma, entre com o número de alunos dessa turma, acrescentando uns 4 ou 5 a mais como reserva; por exemplo: se sua turma tem 30 alunos, indique 34 no aplicativo. Com isso você terá alguns códigos adicionais para o caso de alguém perder.
Animação mostrando o limite do paperclickers para 99 alunos em uma única turma	Paperclickers suporta um limite de até 99 alunos numa mesma turma; seu uso não é adequado para turmas maiores.
Miniatura como link para o vídeo referenciado	Mas é possível utilizar "paperclickers" para diversas turmas de até 99 alunos, mesmo que o total de alunos de todas elas ultrapasse 99. Se você for utilizar "paperclickers" para mais de uma turma, veja o vídeo "Utilizando paperclickers em várias turmas" para instruções específicas.
Animação do nome paperclickers e uma indicação de "curtir"; miniaturas como links para todos os outros vídeos de treinamento	Tudo pode parecer um pouco complicado, mas não se assuste: nos próximos vídeos vamos explicar em detalhes como imprimir os códigos e também como usar o registro de resposta para acompanhar a evolução de seus alunos.

Table 22 – Original script for the training video on paperclickers install and initial usage.

<table>
<tr><th colspan="2">Vídeo 3 de treinamento do paperclickers – Imprimindo os códigos de resposta dos alunos</th></tr>
<tr><td>Descrição do vídeo</td><td>Texto falado</td></tr>
<tr><td>Título: "Imprimindo códigos de resposta dos alunos"</td><td></td></tr>
<tr><td>Finalizando a definição do número de alunos; retornando à tela de configuração e deslizando o menu até a opção "IMPRIMIR CÓDIGOS DOS ALUNOS"</td><td>Depois que você definiu o número de alunos que vão usar "paperclickers", o próximo passo é imprimir os códigos de resposta para distribuí-los.</td></tr>
<tr><td>Apresentação de um cartão de resposta, tamanho A5, bem impresso e outro, mesmo tamanho, com falhas na impressão; animação indicando que o cartão com falhas não pode ser usado</td><td>Para imprimir os códigos você vai precisar ter acesso a uma impressora jato de tinta ou laser que esteja com a tinta — ou com o tonner — em boa qualidade, para que a impressão fique sem falhas.</td></tr>
<tr><td>Manipulação de um cartão de resposta tamanho A5, mostrando frente, com o código, e verso com as diversas opções de resposta; divide a tela em 2, mostrando frente e verso do mesmo cartão; aluno rotaciona o cartão para cada uma das respostas, mostrando que tanto o verso quanto a frente mudam de orientação; para cada resposta selecionada, escreve texto indicado "RESPOSTA SELECIONADA <A|B|C|D>"</td><td>Os códigos de resposta precisam ser impressos com frente e verso, já que a frente é onde está o código de identificação em círculos, e o verso tem a indicação de qual a opção de resposta o aluno quer mostrar. Assim, será mais fácil se você tiver acesso a uma impressora que imprime frente e verso automaticamente.</td></tr>
</table>

… Este script continua na próxima página

Continuação – Vídeo 3 de treinamento do paperclickers – Imprimindo os códigos de resposta dos alunos	
Descrição do vídeo	*Texto falado*
Apresentação dos 3 tamanhos de código: começam os 3 empilhados, do maior para o menor, alinhados canto inferior direito; são manipulados para a direita, ficando na ordem A4, A5 e A6; overlay indicando as distâncias máximas de captura: A4 ~= 11 m; A5 ~= 8 m; A6 ~= 5 m	Você também pode definir o tamanho dos códigos de resposta; esse tamanho poderá variar de 1 até 4 códigos por página. Com os códigos maiores você conseguirá usar “paperclickers” em salas maiores: um código do tamanho de uma página inteira é bem detectável em uma sala onde você fica até uma distância de 11 metros dos seus alunos; um código do tamanho de meia página é visível em uma sala onde você fica até 8 metros de distância dos seus alunos; já um código do tamanho de um quarto de página é apenas bem visível numa sala onde você fica até 5 metros de seus alunos.
Mostra a seleção da opção “Número de códigos por página”, onde é apresentada a janela de escolha “1 por página”, “2 por página” ou “4 por página”	Recomendamos que você utilize códigos do tamanho de meia página (usando a opção de impressão de 2 códigos por página), pois eles oferecem uma boa distância e são mais fáceis para os alunos manipularem e guardarem sem estragar.
Mostra a seleção da opção “Tamanho do papel”, onde é apresentada a janela de escolha “A4” ou “Carta”	Outro parâmetro que você pode alterar para a impressão é o tamanho de folha que será usado: você pode optar pelo padrão de folha tamanho A4 (210 x 297 mm) ou o padrão de folha tamanho Carta (216 x 279 mm).
Mão segurando pela ponta cartões no momento de resposta, tamanho A5, impressos gramatura 120 e 75, mostrando que o último dobra com seu próprio peso	Um último detalhe para realizar a impressão é sobre o tipo de papel a ser utilizado: o ideal é o uso de um papel mais grosso para a impressão dos códigos de resposta, pois assim fica mais fácil usar os cartões; um papel como o “sulfite 40” - de gramatura 120 g/ms - já é o suficiente.
Mostra as opções padrão e a seleção da opção “Imprimir ou exportar códigos”; miniatura como link para o vídeo referenciado	Se você for imprimir 2 códigos por página, numa folha tamanho A4 e usando uma impressora que faz frente e verso automaticamente, você pode manter os valores padrão para as opções de impressão. Se a impressora que você for utilizar não fizer frente e verso automaticamente, veja o vídeo “Imprimindo os códigos de resposta fazendo frente e verso manualmente” para instruções específicas.
Mostra o popup com a indicação de geração do arquivo “.pdf”; a abertura das opções de compartilhamento	Tendo feito essas configurações, selecione a opção “Imprimir ou exportar códigos”: neste momento a aplicação vai gerar um arquivo “.pdf” com os códigos de resposta, conforme as configurações que você fez. Com as opções padrão, será gerado o arquivo “paperclickers_topCodes.pdf” que você deve então mandar para impressão.
Seleção da opção “Gmail”; abertura do aplicativo na tela de envio; preenchimento do endereço de envio; entrada de um texto para assunto, “Códigos de resposta para impressão”, e envio do email	A maneira mais simples de fazer isso é selecionar o email - usando o aplicativo “Gmail” por exemplo - para enviar para você mesmo o arquivo gerado; para isso, entre com o seu endereço de email como destino e envie. É importante lembrar que você precisa estar com o seu telefone com conexão WiFi ou de dados.
Tela windows do computador conectado à impressora; acesso dentro do “Gmail” ao email enviado com o anexo dos códigos de resposta; pedido de impressão	Usando agora um computador conectado à impressora, abra o seu email e selecione a mensagem enviada e peça para imprimir o arquivo anexo.
Impressora imprimindo os códigos; códigos prontos sendo manipulados para mostrar frente e verso; uma página com 2 códigos impressos sendo cortados com tesoura	Uma vez impressos os códigos, corte-os para distribuí-los aos alunos.
	O ideal é que cada aluno tenha o seu próprio código de resposta e já fique com eles sempre - com isso você ganha tempo nas suas aulas e garante que cada aluno use sempre o mesmo código, o que vai facilitar para você acompanhar a evolução de cada um deles individualmente, podendo até mesmo utilizar paperclickers como forma de fazer o registro de presença — a chamada de cada aula.

... Este script continua na próxima página

Continuação – Vídeo 3 de treinamento do paperclickers – Imprimindo os códigos de resposta dos alunos	
Descrição do vídeo	*Texto falado*
Lista de presença em ordem alfabética, com uma animação indicando que o primeiro nome é o código 1, o segundo o 2 e assim por diante	Uma sugestão é que você distribua a sequência dos códigos de resposta seguindo a ordem alfabética dos seus alunos; assim ficará fácil saber que o primeiro aluno dessa lista ordenada alfabeticamente estará como código 1 e assim por diante.
Exemplos de códigos em más condições: rasgados, amassados, dobrados e sujos	Mas para que os alunos fiquem sempre com seus próprios códigos é preciso que eles tenham o cuidado de mantê-los sempre em boas condições: códigos muito sujos ou amassados podem ter sua identificação prejudicada; seus alunos tem também que sempre lembrar de ter com eles seus códigos de resposta, para poderem usar nas aulas.
Desktop com navegador aberto, digitando o endereço dos arquivos ".pdf" dos TopCodes no github do paperclickers contendo todas as opções disponíveis de arquivos	Alternativamente, você pode baixar os arquivos ".pdf" com todos os 99 códigos de resposta da página web do projeto paperclickers: acesse a página "https://github.com/learningtitans/paperclickers/tree/master/topcodes/pt-BR" para ter acesso aos arquivos disponíveis.
Mostra a navegação dentro da área do paperclickers no github até encontrar e selecionar o arquivo referenciado	Você terá que escolher o arquivo ".pdf" correspondente a sua opção de impressão dos códigos; por exemplo, se quiser imprimir os código no tamanho de meia página, numa folha A4 usando uma impressora que faz frente e verso automaticamente, escolha o arquivo "paperclickers_topCodes_2pp_A4.pdf".

Table 23 – Original script for the paperclickers training video about how to print the students' answering cards.

Vídeo 4 de treinamento do paperclickers – Imprimindo os códigos de resposta fazendo frente e verso manualmente	
Descrição do vídeo	*Texto falado*
Título: "Imprimindo códigos de resposta fazendo frente e verso manualmente"	
	Se a impressora que você vai utilizar não faz frente e verso automaticamente, você vai precisar controlar manualmente a impressão dos códigos de resposta.
Seleção da opção "Frente/verso" e depois da opção "Preciso imprimir frente/verso manualmente"	Para tanto, selecione a opção de configuração "Frente/verso" e indique "Preciso imprimir frente/verso manualmente".
Escolha da opção "Imprimir ou exportar códigos"; popup com a geração dos 2 arquivos ".pdf"	Escolha agora a opção "Imprimir ou exportar códigos"; veja que serão gerados agora 2 arquivos ".pdf", um contendo a frente dos códigos de resposta e o outro contendo o verso.
Seleção do aplicativo "Gmail"; abertura do aplicativo na tela de envio; preenchimento do endereço de envio; entrada de um texto para assunto, "Códigos de resposta para impressão", e envio do email	Mande então esses arquivos para a impressão, selecionando, por exemplo, o envio por email: indique a aplicação "Gmail" para enviar para você mesmo os 2 arquivos gerados; para isso, entre com o seu endereço de email como destino e envie.
Tela windows do computador conectado à impressora; acesso dentro do "Gmail" ao email enviado com os anexos dos códigos de resposta; pedido de impressão primeiramente do arquivo frente	Usando agora um computador conectado à impressora, abra o seu email e selecione a mensagem enviada; peça para imprimir primeiramente o arquivo anexo contendo a frente dos códigos de resposta; será o arquivo "paperclickers_topCodes_frente.pdf"
	Aguarde a impressão da frente de todos os códigos de resposta. Uma vez finalizada a impressão, pegue as folhas impressas somente na frente e recoloque na impressora para imprimir então o verso dos códigos.

... Este script continua na próxima página

Continuação – Vídeo 4 de treinamento do paperclickers – Imprimindo os códigos de resposta fazendo frente e verso manualmente	
Descrição do vídeo	*Texto falado*
Pessoa em dúvida de como recolocar os papéis impressos de um lado para imprimir o outro: face impressa para cima ou para baixo? qual orientação utilizar? Animação com indicação dúvida	Nesta hora é preciso atenção: cada impressora tem uma orientação certa para definir o lado e orientação de inserção das folhas a serem impressas. Você vai precisar saber qual é a orientação da sua impressora.
Mostrando ícone indicando o lado de impressão, existente em várias impressoras; associação do ícone com o lado correspondente das folhas: lado do ícone que está riscado indica o lado da folha onde será impresso; animação para indicar qual lado está correto, correspondendo ao ícone apresentado	Muitas vezes existe um desenho indicando qual o lado da folha será impresso: o lado riscado é o lado a ser impresso; neste desenho, está indicado que o lado a ser impresso é o lado da folha que está para baixo dentro da gaveta de impressão.
Pessoa agora em dúvida sobre qua orientação da folha utilizar, uma vez que já se sabe o lado de impressão: cima para baixo ou debaixo para cima? Animação para indicar a dúvida	Agora que você já sabe o lado de impressão, falta apenas saber a orientação da folha: se a impressão vai começar de cima para baixo ou ao contrário.
Apresenta folha em branco; escreve "paperclickers" no topo da folha; apresenta o ícone de indicação do lado de impressão; coloca a folha dentro da impressora com a orientação de impressão de baixo para cima	Para definir isso, uma maneira é você fazer um rápido teste: escreva na parte de cima de um dos lados do papel; coloque agora esse papel na impressora, para que seja impresso nesse mesmo lado em que você escreveu.
Mostra pedindo para imprimir uma página com o verso do código; mostra o resultado com a orientação correta, com animação indicando a orientação do escrito "paperclickers" igual à orientação do número do código de resposta; mostra agora o resultado com a orientação incorreta, com animação indicando a orientação do escrito "paperclickers" invertida com a orientação do código de resposta	Peça então para fazer uma impressão de teste e veja como a impressão saiu: se ficou na mesma orientação que você escreveu, então o jeito que você colocou o papel é o correto; se ficou invertido, o jeito correto de colocar o papel é ao contrário - sempre respeitando o lado de impressão.
Tela windows do computador conectado à impressora apresentando o email recebido; pedido de impressão do arquivo contendo o verso dos códigos de resposta	Agora que você já sabe qual o lado e orientação para a impressão do verso dos códigos, coloque corretamente na impressora as folhas já com a impressão da frente, e peça então para imprimir o verso dos códigos, que será o anexo com o nome "paperclickers_topCodes_verso.pdf ".
Impressora imprimindo o verso dos códigos; códigos prontos sendo manipulados para mostrar frente e verso; uma página com 2 códigos impressos sendo cortados com tesoura	Uma vez impressos os códigos, corte-os para distribuí-los aos alunos.
	O ideal é que cada aluno tenha o seu próprio código de resposta e já fique com eles sempre - com isso você ganha tempo nas suas aulas e garante que cada aluno use sempre o mesmo código, o que vai facilitar para você acompanhar a evolução de cada um deles individualmente, podendo até mesmo utilizar paperclickers como forma de fazer o registro de presença — a chamada de cada aula.
Lista de presença em ordem alfabética, com uma animação indicando que o primeiro nome é o código 1, o segundo o 2 e assim por diante	Uma sugestão é que você distribua a sequência dos códigos de resposta seguindo a ordem alfabética dos seus alunos; assim ficará fácil saber que o primeiro aluno dessa lista ordenada alfabeticamente estará como código 1 e assim por diante.
Exemplos de códigos em más condições: rasgados, amassados, dobrados e sujos	Mas para que os alunos fiquem sempre com seus próprios códigos é preciso que eles tenham o cuidado de mantê-los sempre em boas condições: códigos muito sujos ou amassados podem ter sua identificação prejudicada; seus alunos tem também que sempre lembrar de ter com eles seus códigos de resposta, para poderem usar nas aulas.

... Este script continua na próxima página

Continuação – Vídeo 4 de treinamento do paperclickers – Imprimindo os códigos de resposta fazendo frente e verso manualmente	
Descrição do vídeo	*Texto falado*
Desktop com navegador aberto, digitando o endereço dos arquivos ".pdf" dos TopCodes no github do paperclickers contendo todas as opções disponíveis de arquivos	Alternativamente, você pode baixar os arquivos com todos os 99 códigos de resposta da página web do projeto paperclickers: acesse a página "https://github.com/learningtitans/paperclickers/tree/master/topcodes/pt-BR" para ter acesso aos arquivos disponíveis.
Animação mostrando a navegação e seleção do arquivo referenciado	Você terá que escolher os arquivos ".pdf" correspondente a sua opção de impressão dos códigos; por exemplo, se quiser imprimir os código no tamanho de meia página, numa folha A4 usando uma impressora na qual você precisa fazer frente e verso manualmente, escolha os arquivos "paperclickers_topCodes_frenteSomente_2pp_A4.pdf" and "paperclickers_topCodes_versoSomente_2pp_A4.pdf" files.

Table 24 – Original script for the paperclickers training video with alternate instructions about how to manually do two-sided printing.

Vídeo 5 de treinamento do paperclickers – Compartilhamento e uso do registro de respostas	
Descrição do vídeo	*Texto falado*
Título: "Compartilhamento e uso do registro de respostas"	
Animação mostrando a sequência da tela de captura de respostas e avanço até a tela de respostas detalhadas; nesse ponto, animação indicando a criação de um novo registro no registro interno de respostas. Repete a sequência durante o tempo do texto, cada vez criando um novo registro	Toda vez que você utiliza o paperclickers para capturar respostas de seus alunos, as respostas capturadas são registradas internamente. Você pode utilizar esse registro para acompanhar a evolução de seus alunos, aula a aula, pergunta a pergunta. Por isso é importante que você sempre saiba qual aluno está utilizando qual código de resposta: nesse registro é gravada a opção respondida por cada um dos códigos de resposta, para cada pergunta que você fez, identificada pela data e hora, além de um número sequencial.
Animação com data e hora da primeira pergunta; captura das 30 respostas; registro das respostas; recomeça a animação agora com a data e hora da segunda pergunta	Por exemplo: se no dia 06 de fevereiro de 2018, às 9h15 da manhã você fez uma pergunta para sua classe, onde estavam presentes os 30 alunos, e utilizou paperclickers para capturar a resposta, você terá o registro de cada uma das 30 respostas dadas, identificadas pelo código de resposta — nesse caso, pelos códigos de 1 ao 30. Se nesse mesmo dia, às 10h00 você fez outra pergunta utilizando paperclickers, você terá um novo registro das 30 respostas dadas. Dessa forma você poderá, consultando esse registro de respostas, saber qual foi o desempenho dos seus alunos, uma vez que você souber qual aluno está usando qual código de resposta.
Apresenta um planejamento de aula, escrito à mão em um caderno, com a indicação do assunto e de 2 perguntas para serem feitas aos alunos	Para ter um registro completo, você vai precisar ter um controle à parte para saber quais foram as perguntas feitas, ou pelo menos o assunto tratado, em cada aula.
Entra em "Configurações"; seleciona opção "Exportar registro de respostas"; seleciona "Gmail"; na tela de nova mensagem, preenche o endereço de destino, preenche assunto com "Registro de respostas da turma A"; envia email	Para consultar o registro de respostas, você precisa compartilhá-lo para fora da aplicação "paperclickers"; para fazer isso, entre na opção de "Configurações" e escolha "Exportar registro de respostas". Novamente, a maneira mais simples é mandar esse registro por email para você mesmo: escolha, por exemplo, o aplicativo "Gmail", preencha o endereço de destino como seu endereço de email e coloque como assunto algo que ajude a identificar o material — por exemplo "Registro de respostas da turma A". Então envie o email que terá como anexo o arquivo "paperclickers_RegistroDeRespostas.csv" file.

... Este script continua na próxima página

Continuação – Vídeo 5 de treinamento do paperclickers – Compartilhamento e uso do registro de respostas	
Descrição do vídeo	*Texto falado*
Em um desktop windows, acessa o email e seleciona a mensagem recebida; grava o arquivo anexo "paperclickers_RegistroDeRespostas.csv" na pasta "documentos"	Para ler o registro de respostas compartilhado, abra seu email em um computador e acesse a mensagem que acabou de enviar; salve o arquivo anexo localmente no seu computador.
Mostra uma tela do "Excel" e depois uma tela do "LibreOffice Calc"	Esse arquivo é gravado como um arquivo texto padrão, mas num formato que permite ser aberto por softwares de planilha eletrônica, como por exemplo o "Excel", do pacote "Office" da Microsoft ou "Calc", do pacote "LibreOffice" — você vai precisar de um deles para poder manipular mais facilmente o registro de respostas.
No "windows explorer", na pasta "documentos", seleciona com o mouse o arquivo "paperclickers_RegistroDeRespostas.csv" e abre com clique duplo, iniciando o excel até aparecer a planilha com os registros de respostas	Num computador onde você tem, por exemplo, o "Excel" instalado, simplesmente faça um clique duplo do mouse sobre o nome do arquivo que ele será apresentado como uma planilha; nessa planilha, cada linha corresponderá a uma pergunta que você fez e utilizou paperclickers para capturar as respostas de seus alunos.
No "Excel" com o registro de respostas aberto, seleciona a coluna "SEQ", depois a coluna "DATA E HORA" e depois as colunas a partir da quarta, com as respostas individuais	A primeira coluna, identificada como "SEQ", é o número da questão que você fez numa mesma sessão de uso da aplicação; a segunda coluna, "DATA E HORA", indica justamente a data e hora que você capturou as respostas correspondentes; a partir da quarta coluna você terá cada uma das respostas detectadas para os códigos de 1 à 99, sendo que células vazias indicarão a ausência de resposta.
No "Excel" como registro de respostas aberto, seleciona a coluna "QUESTÃO"	A terceira coluna do registro de respostas, de nome "QUESTÃO", corresponde a uma opção de uso que vem desativada no paperclickers, mas que você pode escolher ativar: você pode escolher entrar um breve texto para identificar cada pergunta que fizer para seus alunos utilizando paperclickers; isso permitirá que você já deixe gravado, no próprio registro de respostas, um texto indicativo de qual foi a pergunta feita, facilitando a sua consulta posterior do registro.
Entra em "Configurações"; seleciona a opção "Entrar texto para registrar questões?" e escolhe "Sim"	Para ativar essa opção de entrada de texto, entre em "Configurações" e selecione a opção "Entrar texto para registrar questões?", respondendo "Sim".
Aparece primeira tela do "paperclickers"; seleciona botão "início"; aparece tela "Pergunta para turma"; digita texto "Causas do aquecimento global"	Dessa forma, sempre que você for fazer uma pergunta para seus alunos, aparecerá uma tela pedindo para você escrever algo que identifique a "Pergunta para turma" que você irá fazer; o texto que você digitar nessa tela vai ser gravado na coluna "QUESTÃO" dentro do registro de respostas.
Abrindo o registro de respostas no "Excel", agora com a mesma "QUESTÃO" digitada anteriormente	Veja como dessa forma fica bem mais fácil utilizar o registro de respostas para analisar o desempenho da sua turma.
Animação mostrando a captura de respostas e criação correspondente de uma entrada no registro de respostas; mostra o registro das primeiras 30 respostas de uma pergunta, indicando através de animação que as respostas faltantes indicam alunos que não foram na aula	Dependendo da sua dinâmica de uso do paperclickers, é possível utilizá-lo como ferramenta para o controle de presença de seus alunos: se em toda aula você fizer uma pergunta e utilizar o paperclickers para capturar as respostas, você terá no registro de respostas a identificação de todas as respostas dadas, o que vai corresponder a indicação de quais alunos estiveram presentes naquela aula. Por exemplo: se sua classe tem 30 alunos e para uma pergunta no registro de respostas você tiver apenas 27 respostas, os alunos que utilizam os códigos de resposta que estavam em branco não responderam à pergunta e provavelmente não estiveram presentes naquela aula.

… Este script continua na próxima página

Continuação – Vídeo 5 de treinamento do paperclickers – Compartilhamento e uso do registro de respostas	
Descrição do vídeo	*Texto falado*
Mostra tela de captura do paperclickers, com os alunos com as respostas ao fundo; miniatura como link para o vídeo referenciado	Mas para utilizar paperclickers como uma forma de registro de presença, é muito importante que você sempre verifique que todas as respostas foram capturadas no processo. Veja o vídeo "Dicas para a realização da captura das respostas" para instruções em como garantir a eficiência da captura das respostas pelo paperclickers.

Table 25 – Original script for the paperclickers training video about how to access and use the answers log to follow the students performance.

Vídeo 6 de treinamento do paperclickers – Utilizando paperclickers em várias turmas	
Descrição do vídeo	*Texto falado*
Título: "Utilizando paperclickers em várias turmas"	
Animação mostrando várias turmas e a dúvida sobre como distribuir os códigos de resposta	Se você quiser utilizar paperclickers com várias turmas, você poderá escolher de que forma você irá distribuir os códigos para todos os seus alunos; dependendo da quantidade total de alunos, você poderá escolher entre duas possibilidades.
	Se você tiver um total de até 99 alunos, considerando todas as suas turmas, é possível atribuir um código de resposta único para cada um de seus alunos. Isso pode ser interessante se você quiser sempre ter uma maneira fácil de analisar o desempenho de todos os seus alunos, em conjunto, lembrando que o registro de resposta armazena as respostas dos códigos de 1 a 99 para cada pergunta.
Animação mostrando 28 ícones agrupados, identificando cada aluno da primeira turma; na sequência aparecem mais 26 ícones agrupados, identificando cada aluno da segunda turma; abaixo do primeiro grupo aparece o texto "códigos 1 ao 28", abaixo do segundo grupo aparece o texto "códigos 29 ao 54"	Assim, se você for utilizar paperclickers com duas turmas, uma com 28 alunos e outra com 26, você pode distribuir os códigos de respostas de 1 ao 28 para a primeira turma e de 29 ao 54 para a segunda.
Apresenta registro de respostas; seleção das colunas de 1 ao 28; seleção das colunas de 29 ao 54, sincronizadas com o texto	Assim, no registro de respostas, você saberá que perguntas com respostas para os códigos de 1 ao 28 são da primeira turma, e de 29 ao 54 da segunda.
Lista de presença em ordem alfabética, com uma animação indicando que o primeiro nome é o código 29, o segundo o 30 e assim por diante	Você vai precisar saber que os códigos de resposta da segunda turma começam no 29, para associar os códigos de resposta dos alunos a partir da ordem alfabética da lista de chamada.
Reapresenta animação mostrando 28 ícones agrupados, identificando cada aluno da primeira turma; na sequência aparecem mais 26 ícones agrupados, identificando cada aluno da segunda turma; na sequência aparecem mais 4 ícones representando os códigos adicionais de reserva	Para utilizar essa forma de atribuição de códigos para várias turmas, você deve definir o número total de alunos, na tela de configuração de paperclickers, como sendo a soma dos alunos das duas turmas, adicionando talvez uma sobra de 4 ou 5 códigos de resposta, para eventualidades como perda ou participações especiais. No exemplo, esse número total de alunos seria 58.
Animação com códigos sendo todos impressos formando uma grande pilha; ao término da impressão essa pilha é então parcialmente separada formando uma segunda pilha, aparecendo ao final, abaixo da primeira pilha, o texto "Turma 1 – códigos 1 ao 28", e abaixo da segunda pilha, o texto "Turma 2 – códigos 29 ao 54	Uma vez impressos os códigos de resposta, você deverá distribuí-los de acordo com a atribuição a cada turma; no exemplo, os alunos da primeira turma recebem os códigos de 1 ao 28, e os alunos da segunda turma recebem os códigos de 29 ao 54.
Apresenta miniaturas como links dos 2 vídeos referenciados	Reveja os vídeos "Instalação e execução inicial do paperclickers" e "Imprimindo códigos de resposta dos alunos" se tiver dúvidas como fazer essa definição e imprimir os códigos.

... Este script continua na próxima página

Continuação – Vídeo 6 de treinamento do paperclickers – Utilizando paperclickers em várias turmas	
Descrição do vídeo	*Texto falado*
Reapresenta a animação mostrando 28 ícones agrupados, identificando cada aluno da primeira turma; na sequência aparecem mais 26 ícones agrupados, identificando cada aluno da segunda turma; abaixo do primeiro grupo aparece o texto "códigos 1 ao 28", abaixo do segundo grupo aparece o texto "códigos 1 ao 26"	Uma outra maneira de distribuir os códigos de resposta entre as suas diversas turmas é considerar que cada turma sempre começa com o código de resposta 1. Assim, no mesmo exemplo anterior, a primeira turma utilizaria os códigos de resposta de 1 ao 28 e a segunda turma os códigos de resposta de 1 ao 26. Dessa maneira, é sempre fácil associar um aluno de cada turma ao seu código de resposta, a partir da ordem alfabética da lista de chamada.
Apresenta registro de respostas, com perguntas com respostas de 1 ao 28 e respostas de 1 ao 26; animação identificando cada entrada como sendo da "turma 1" ou "turma 2" a partir do campo "DATA E HORA"	Entretanto, é preciso um cuidado adicional para o uso do registro de respostas, para saber qual turma as respostas correspondem: para isso vai ser preciso você utilizar o valor da coluna "DATA E HORA", para identificar corretamente para qual turma você fez a questão registrada. Essa forma de atribuição de códigos de resposta é recomendada para o caso de você ter várias turmas cujo total de alunos ultrapasse os 99 códigos de resposta possíveis na aplicação.
Reapresenta a animação mostrando 28 ícones agrupados, identificando cada aluno da primeira turma; na sequência aparecem mais 26 ícones agrupados, identificando cada aluno da segunda turma; abaixo do primeiro grupo aparece o texto "28 alunos", abaixo do segundo grupo aparece o texto "26 alunos"; na sequência aparecem mais 4 ícones agrupados representando os alunos adicionais; surge então o total de 32 alunos	Para utilizar essa forma de atribuição de códigos de resposta, você deve definir o número de alunos para a aplicação como sendo o número total da sua maior turma; no exemplo anterior, deveria ser 28, que pode ser acrescido de 4 ou 5 códigos de resposta adicionais para o caso de eventuais perdas ou participações especiais em aulas. Assim você deve definir o total de 32 como sendo o total de alunos a serem tratados pelo paperclickers.
Apresenta miniatura como link do vídeo referenciado	Reveja o vídeo "Instalação e execução inicial do paperclickers" caso haja dúvidas em como fazer essa definição do número de alunos.
Animação com códigos sendo impressos e empilhados numa primeira pilha, aparecendo ao final, abaixo dessa primeira pilha, o texto "Turma 1 – códigos 1 ao 28"; repete a animação, com a impressão e criação de uma segunda pilha, aparecendo ao final o texto, abaixo dessa segunda pilha, o texto "Turma 2 – códigos 1 ao 26	Uma vez definido esse total, você deverá realizar tantas impressões quantas forem o número de turmas, já que deverá distribuir sequências de códigos semelhantes para cada turma.
Apresenta miniatura como link do vídeo referenciado	Reveja o vídeo "Imprimindo códigos de resposta dos alunos" caso haja dúvidas em como fazer uma impressão, e repita os passos finais de impressão dos códigos de resposta até completar os conjuntos necessários. No exemplo, você deverá imprimir 2 vezes o conjunto de códigos gerado.

Table 26 – Original script for the training video on how to use paperclickers on several classes.

Vídeo 7 de treinamento do paperclickers – Dicas para a realização da captura das respostas	
Descrição do vídeo	*Texto falado*
Title: "Dicas para a realização da captura das respostas"	
	Quando você for capturar as respostas dos seus alunos utilizando "paperclickers", existem alguns cuidados que você pode tomar que melhorarão o reconhecimento, tornando todo o processo mais rápido e confiável.
Apresenta vários alunos em uma sala, apresentando seus códigos de resposta; foco em situações problemáticas, como códigos se sobrepondo, alunos segurando os cartões de forma a cobrir os códigos	A primeira dica é pedir para os alunos deixarem seus cartões bem visíveis, segurando-os na área indicada no verso, para evitar cobrir a área do código de resposta impresso na parte da frente — os códigos são os círculos pretos.

... Este script continua na próxima página

Continuação – Vídeo 7 de treinamento do paperclickers – Dicas para a realização da captura das respostas	
Descrição do vídeo	*Texto falado*
Apresentação da tela de captura, com o celular muito próximo de um código tamanho A5 com uma dobra, que não é reconhecido; continua na mesma cena, mas agora afastando o código até que ele seja corretamente reconhecido	Quando for realizar a captura, não fique muito próximo dos alunos; o melhor desempenho é a partir de uma distância de pelo menos 2 metros entre a câmera do seu celular e os códigos de resposta dos alunos, considerando os códigos no tamanho de meia página — correspondentes à impressão de 2 por página. Se você estiver a uma distância menor, qualquer defeito ou imperfeição no cartão — por exemplo uma dobra ou sujeira — vai dificultar o seu reconhecimento.
Apresentação dos 3 tamanhos de código: começam os 3 empilhados, do maior para o menor, alinhados canto inferior direito; são manipulados para a direita, ficando na ordem A4, A5 e A6; overlay indicando as distâncias máximas de captura: "A4 ~= 11 m; A5 ~= 8 m; A6 ~= 5 m"	Mas por outro lado, lembre-se também que existe o limite da distância máxima para o reconhecimento dos códigos de resposta, e essa distância varia com o tamanho que você escolheu para imprimir os códigos de resposta. Se sua sala for muito grande, com você ficam a mais de 10 metros de distância do aluno mais longe, você deverá imprimir os códigos de resposta no tamanho de 1 por página, para conseguir uma boa performance de detecção.
Tela de captura, mostrando a câmera sendo posicionada de forma paralela aos códigos de resposta, indicando ser a situação correta; uma segunda tomada da tela de captura, agora com a câmera sendo posicionada de maneira angulada aos códigos, tendo dificuldade de captura	Ao capturar as respostas, procure ficar sempre de frente para os alunos; movimente-se pela frente da sala, evitando que os cartões fiquem muito de lado para a câmera.
Tela de captura ativa, com os alunos ainda escolhendo as respostas e posicionando os cartões de resposta; animação indicando ser procedimento incorreto; tela de captura ativa, capturando parcialmente a tela, mas parando a captura, movendo até a câmera para baixo — animação indicando que é uma interrupção —; posterior retomada do processo de captura; animação com indicação de ser procedimento incorreto	Por fim, durante cada processo de captura, evite ficar muito tempo na tela de captura, com a câmera do celular ligada: isso vai fazer com que os códigos de resposta demorem mais para serem reconhecidos, além de gastar mais rapidamente a sua bateria. Entre na tela de captura somente depois que todos os seus alunos já levantaram os cartões de resposta; uma vez na tela de captura, tente detectar todos os códigos de resposta sem muita interrupção — por exemplo, para responder a alguma dúvida de última hora.
Tela de captura ativa, processo de captura ocorrendo ainda sem terminar; evolução para a tela de respostas — animação indicando que é uma interrupção —; posterior retorno para a tela de captura e continuação da captura até terminar; evolução para a tela de respostas, agora completa	Caso seja preciso interromper o processo de captura, faça a captura em mais de um passo, indo até a tela de respostas, onde você poderá ver o resultado captura parcial dos códigos, e retornando à tela de captura para finalizar os códigos faltantes: paperclickers vai reconhecer que você quer completar a captura, caso retorne à tela de captura sem iniciar uma nova questão.

Table 27 – Original script for the paperclickers training video on how to effectively capture the students' answers.

APPENDIX B – Original version of Peer Instruction material creation guidelines

The Peer Instruction material creation guidelines were initially designed in Brazilian Portuguese, considering the target audience of Brazilian public high school STEM teachers. The guidelines for basic content was described in section 5.3.3, and the original version is included here.

B.1 Instruções para criação de material de aula — Instrução pelos Pares e paperclickers

Trabalhar com instrução pode ser muito diferente de uma aula expositiva tradicional, tanto em termos dos resultados com seus alunos, mas também em termos do material que você precisa ter preparado para suas aulas.

Este material oferece alguns direcionamentos para a criação desse material para uma aula de instrução pelos pares, considerando que você já tem um material pronto para uma aula expositiva.

Como o objetivo da instrução pelos pares é conduzir a aula a partir de perguntas chave para os conceitos que forem apresentados, o preparo do material de aula conterá a formulação de perguntas que consigam estimular a discussão entre os alunos, proporcionando que eles construam e consolidem seu conhecimento através da defesa de suas respostas. E nesse processo será possível para você saber e validar o entendimento dos alunos a respeito dos conceitos apresentados.

O método de instrução pelos pares pode ser usado de forma gradativa: você pode conduzir pequenos trechos, cobrindo poucos conceitos — até mesmo apenas um — numa aula; dessa maneira será mais fácil para você se familiarizar com o modo de trabalho e também realizar o preparo do material para suas aulas.

Partindo então de seu material e experiência prévios das aulas expositivas, os itens a seguir compõem uma sugestão para a construção de uma aula em instruções pelos pares:

1. Liste os conceitos principais dentro do material a ser exposto em aula — as perguntas que serão utilizadas deverão explorar esses conceitos; use como referência o seu planejamento tradicional de aula expositiva; lembre-se que você poderá aplicar instrução pelos pares a apenas um subconjunto desses conceitos.

2. Planeje para cobrir poucos conceitos por aula — a aplicação da dinâmica de instrução pelos pares ocupa tempo que numa aula tradicional estaria sendo usado para apresentar novo conteúdo.

3. Pense agora em questões que possam explorar cada um desses conceitos; foque em problemas e dificuldades comuns dos alunos para o assunto em questão — utilize o histórico de outras turmas para a disciplina.

4. Considere que as questões precisam ser simples o suficiente para serem respondidas em poucos minutos, mas precisam também ser representativas o suficiente para avaliar o essencial dos conceitos.

Para facilitar na dinâmica da aula, o método de Instrução pelos Pares considera que as perguntas conceituais sejam de múltipla escolha; para o uso do paperclickers, cada pergunta pode ter até no máximo 4 opções de respostas.

Seguem abaixo um conjunto de orientações gerais para a criação de perguntas de múltipla escolha; a lista é heterogênea, com sugestões que valem para qualquer assunto ou área do conhecimento, e com outras que talvez se apliquem melhor em determinadas situações:

1. Mantenha apenas o essencial nas perguntas — apresente um problema claro, sem incluir na pergunta informações que sejam irrelevantes para o que se deseja avaliar; evite redigir a pergunta na forma de negação.

2. Sempre que possível utilize comparações e contraste — com isso você chamará a atenção para a diferença entre situações, cenários ou conceitos.

3. Avalie a possibilidade de estender os contextos de aplicação, utilizando questões já vistas ou trabalhadas, aplicando-as para novas situações; isso permitirá uma avanço gradual.

4. Utilize questões que requeiram interpretação de representações.

5. Utilize restrições para solução como forma de chamar a atenção para pontos específicos — direcione a resposta, indicando uma determinada abordagem para ser usada ou para ser evitada.

6. Faça questões que permitam a apresentação de uma forma alternativa — e mais vantajosa — de resposta.

7. Faça perguntas que busquem apenas a definição de uma estratégia de resolução.

Seguem algumas sugestões, pensando agora especificamente em como criar as alternativas de resposta:

1. Pense primeiramente na resposta correta, construindo-a para ser a mais clara possível.

2. Se possível, inclua entre as possíveis respostas, alternativas que permitam identificar diferentes dificuldades conceituais, aumentando assim a informação sobre os alunos que você obterá com as respostas.

3. Uma maneira de montar as alternativas é incluir erros comuns dos alunos, listados a partir da experiência passada no tema/conceito.

4. Inclua alternativas que sejam defensáveis, evitando as obviamente incorretas — isso irá criar a oportunidade de fomentar discussão dentro da sala de aula.

5. Evite incluir pistas para a resposta correta: as alternativas devem ser homogêneas em conteúdo (nenhuma deve ser deliberadamente simples ou simplória); devem usar linguagem similar, devem ser ordenadas alfabeticamente para evitar tendência de posições.

6. As respostas alternativas devem ser mutuamente exclusivas.