Teaching Embedded Systems by Constructing an Escape Room

Marc Pfeifer
pfeiferm@informatik.uni-freiburg.de
Chair for Computer Architecture
University of Freiburg
Germany

Benjamin Völker
voelkerb@informatik.uni-freiburg.de
Chair for Computer Architecture
University of Freiburg
Germany

Sebastian Böttcher
sebastian.boettcher@uniklinik-freiburg.de
Ubiquitous Computing
University of Siegen
Germany

Sven Köhler
koehlers@informatik.uni-freiburg.de
Computer Networks and Telematics
University of Freiburg
Germany

Philipp M. Scholl
pscholl@informatik.uni-freiburg.de
Chair for Computer Architecture
University of Freiburg
Germany

ABSTRACT

Embedded systems are the foundation of many of today's consumer products and industrial applications – and they are increasingly connected. To teach this topic we created a course with the overarching goal of designing and constructing an automated escape room. This provided motivation and opportunity for students to learn the engineering and soft skills required for building networked embedded systems. The game was open for faculty members and friends of the students after the course concluded. By dividing the building process into multiple tasks, such as individual puzzles, the presented concept encourages inter- and intra-group work, including conceptualizing, designing and developing reliable, connected embedded systems. In this paper we first present the motivation, context, and pedagogical approach of the course, then describe the course structure and conclude with experiences from constructing an escape room with multiple groups of students.

CCS CONCEPTS

• Computer systems organization → Embedded systems; **• Social and professional topics → Computing education**; **• Human-centered computing → Interaction design**.

KEYWORDS

Embedded Systems; Internet of Things; Lab Course; Escape Room

ACM Reference Format:
Marc Pfeifer, Benjamin Völker, Sebastian Böttcher, Sven Köhler, and Philipp M. Scholl. 2021. Teaching Embedded Systems by Constructing an Escape Room. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432485

SIGCSE '21, March 13–20, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432485

1 INTRODUCTION

Projects play a significant role in computer science education. They teach not only technical skills, but also the soft skills required to succeed when working in teams [7]. Team- and project-based learning is common in Ubiquitous Computing which is focused on networked, embedded systems. However, courses in this domain are particularly challenging due to its interdisciplinary nature and the heterogeneous knowledge and interests of the students [5, 8, 11]. Courses aim to counter these challenges with 'real-world' projects over the duration of several semesters to teach multiple phases of software projects [15], or by connecting students with stakeholders from the local community [7] to increase their motivation with meaningful work. Unique opportunities like building prototypes for rooftop greenhouses [12] provide a common theme for a course in which students can also teach each other. Internet of Things (IoT) courses with individual projects that can be chosen freely cater directly to the interests of the students [1]. Building collaborative games which can be played by larger groups are a way to particularly highlight the challenges of networked, embedded systems and motivate students in a playful manner [4, 9]. Such project-based learning approaches [14] are aligned with engineering practices and therefore believed to be one of the most effective engineering teaching principles. Combined with team-based learning [10], these approaches further provide opportunities for developing the students' soft skills.

However, such courses often lack an overarching goal which combines the work of all students into a common, larger achievement or technical system. This common goal further improves the motivation of students, and teaches soft skills beyond what can be gathered from a small and sharply-focused group. One example is the Formula Student [19] challenge. Students of various faculties participate in a competition of self-built race cars. Inspired by this and with three years of project-based teaching experience of over 120 students, we decided to design the *Ubiquitous Computing Lab* course. The common goal of this course is to build a fully automated escape room with networked, embedded systems.

An escape room is essentially a game where players are "locked" inside a room and have to find their way out by solving different puzzles in a given time [16]. The rooms and puzzles are themed, e.g. as a medieval dungeon or an abandoned space station. The degree of automation ranges from purely analog, e.g. finding codes for pad locks, to complete automation, in which all puzzles are digitally connected and controllable. Recreational escape rooms are a popular activity and increasingly discovered by educators [3, 6, 20] to boost the motivation among students in school and higher education. In contrast to utilizing an escape room which aims to playfully teach, we present a course in which the actual construction of the escape room is the lesson.

The remainder of the paper is structured as follows: Section 2 describes the context, objectives, and structure of the course as well as the story of the escape room. After this in Section 3 we present hardware and software that is vital for a successful execution, and we describe some of the developed components to showcase the students' lessons learned. Finally, our observations of the learning effects as well as the students' evaluation results and assessment are discussed in Section 4.

2 COURSE DESIGN

The *Ubiquitous Computing Lab* course was designed for students of technical postgraduate programs, including students from programs in embedded systems, computer science or electrical engineering. The course covered a 14 week semester and students were awarded with six credit points at successful completion, which roughly amounts to 180 hours of work. We followed a constructivist approach, hence we encouraged students to learn by exploration [2]. During such explorations the students *"create their own knowledge frameworks into which they incorporate their learning"*, and the hope is *"that in being more learner-centric by taking into account the different characteristics of individual students it [the constructivist instruction; note from the authors] can successfully engage a wider range of learners"* [21]. Following the constructivist approach, the escape room and the related challenges did mimic the working environment of an engineer, which includes teamwork and the development of reliable prototypes in a given time. We, the instructors, took the role of advisors which are consulted by the students.

The students were responsible for transforming an almost empty room into a smart escape room from scratch. The room was outfitted only with basic infrastructure (e.g. power, network connection). It was the students task to create additional infrastructure in the room (e.g. automatic doors, controllable lighting, decorations), the puzzles and an operations control center. Each of these components of the escape room was designated as a student project, and projects were assigned to groups of 2–3 students. Groups worked on at least one project each, but could also be assigned to two or more smaller projects. All components were required to be remotely controllable through a (wireless) network connection. While each project was mostly self-contained, they needed to be continuously integrated in the overall story following predefined network communication protocols and the requirements of other projects. Hence, not only intra-group but also inter-group teamwork was necessary, resulting in a good imitation of a typical engineer's working environment.

The learning goals of the presented lab can be summarized as follows: (1) Building reliable IoT devices as a team from scratch. (2) Implementing complex human-machine interactions with networked and embedded devices. (3) Integrating sensor and actor devices used in industrial applications.

2.1 Course Sequence

Prior to the first meeting with the students, we picked an on-campus location for the escape room and prepared the rooms with the necessary basic infrastructure, i.e. power and network connectivity. A location that consists of two or three connected rooms allows to split the game's story into multiple stages. Based on that we came up with a theme for the escape room and created a rough story line which served as a starting point for the discussion with students. We then derived reasonable sub-topics from the story which were chosen by the groups.

During the semester, weekly mandatory two-hour meetings for all students were scheduled. The main purpose of these meetings was to synchronize all projects, discuss current problems, schedule a plan for the next week and test the escape room. These meetings were divided into four phases:

Organization: The organization phase was limited to the first week. Its main purpose was to explain the course structure and the story line of the escape room. We provided an initial view of the topics, but deliberately kept details of the story open. Our goal was to develop the projects in collaboration with the students, to make sure that the students 'owned' their part of the game. Hence we only split the game into several levels with a rough description, e.g. group #1 - named the operator group - should take care of the infrastructure required to operate the escape room from the outside. Group #4 should develop two automatic doors and should think of a way to ensure that all players are located in the room before it gets "locked". Group #6 was tasked with building a 'research prototype' to be stolen and so forth. The way of solving the tasks was left up to the students. The organization phase was also used to assign students to groups and clarify remaining organizational issues.

Initial Design: During the second and third week we refined the story together with the students. We asked the students to present refined ideas for their part of the game, including puzzles and other components. These project ideas were heavily discussed and improved in plenum, leading to the final story of the room with all puzzles that the players needed to solve during the escape game. In contrast to other lab courses where projects often exhibit a high degree of innovation, we employed the concept of *Stability over Innovation.* As robust and stable components are crucial for a flawless game experience and therefore also for the success of the lab, we decided to sacrifice the degree of innovation for improved stability. This means that the projects and single components were kept at a manageable complexity level and enough time was reserved for testing. Students usually lack experience and tend to overestimate their productivity. Therefore, the task of the instructors was to evaluate the practicability (and probable stability) of the ideas based on the instructor-assessed student's knowledge and the instructor's experience from prior lab courses.

Continuous Integration: Beginning in the fourth week, we started to perform weekly integration runs, implementing the concept of stability over innovation into the week-to-week progress of the course. Each meeting was started with a brief verbal presentation of each group. The groups explained what they had done during the last week, what problems arose and what they are planning for

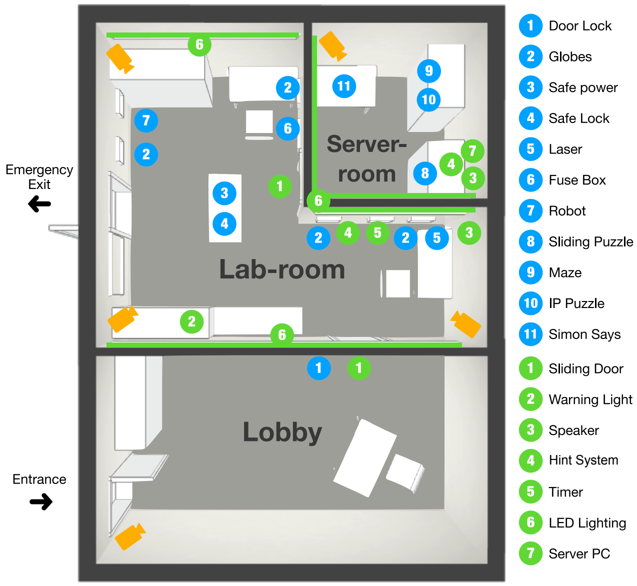

Figure 1: The floor plan of the escape room. Puzzles are marked with blue dots, other components are marked with green dots. The cameras allow for remote monitoring.

the next week. Problems and further questions were then discussed in plenum.

We then conducted a real-life test of all puzzles and other components in the escape room, i.e. we played the game: each week two members – either students from the course or invited guests – went into the room, tried to solve the puzzles in their respective implementation state and reported their experience to the whole course. In the early stages, puzzles were patched together using breadboards, cardboard and tape, and rarely worked as intended. Often a group member responsible for the puzzle had to explain what was supposed to happen. The first test runs already highlighted sporadic malfunctions, design issues or user behavior which the designer had not thought about previously. As the projects evolved towards their final form, the purpose of the integration runs shifted more and more from pure bug discovery to an optimization of the interaction with the players. Especially in the final weeks, the integration runs helped to debug the network communication and to improve the overall game experience. We think that the weekly test runs encouraged the students to work throughout the semester rather than having to finish everything during the last week, particularly since all projects depended on each other. Even in the early stages, the integration runs helped to achieve visible progress across groups.

We set two milestones towards the end of the course. Four weeks before the final presentation, the hardware of all puzzles and other components had to be finished and everything had to be installed in the room. Two weeks later the software also had to be at the final stage. Minor changes were still necessary after the deadlines. Thus, the last two weeks were reserved for further tests, debugging and optimizations of the components and gameplay.

Final Presentation: The meeting in the final week was designed as a closing event. During the whole day friends, colleagues and faculty members were invited to play the escape room. In the evening the students presented their work on site in an open demo session and answered questions. In order to receive their final grade all groups had to hand in a project documentation (consisting of a GIT *readme* file) a few weeks later. Due to the overwhelming interest to play, we scheduled three additional days during which the room could be played, and had students from the course handle additional sessions.

2.2 The Escape Room

The pilot-run of the lab course was conducted in the winter semester 2019/2020 at the University of Freiburg with 20 students. The floor plan of the room can be found in Figure 1. We made use of two connected offices with a combined area of about 35 m^2 and separated a third smaller room using dry-walls. Walls were covered in wood to protect the existing surface. We distributed multiple power outlets along the top of the rooms to power the individual puzzles, and installed a central server computer and a WiFi interface. The escape room theme was a high-tech computer laboratory from which the players had to steal a secret 'research prototype'. It was introduced to players using the following mission description:

> "You are a team of special agents charged with stealing a highly classified software prototype from a secret laboratory at the University of Freiburg. We know that the lab is secured with a state-of-the-art smart security system called STASIS (STate-of-the-art AsSIStant). STASIS is an AI controlling the room's infrastructure and interacts with the inhabitants. You will need to outwit STASIS to access the prototype, which threatens to destroy the world."

Together with the students we decided on details of the story, of the puzzles as well as the players' steps through the escape room (see Figure 1). Group #1 had to create the operations control center which connects and coordinates the components of all other groups. Group #2 was in charge of building additional infrastructure like smart lighting and the voice-interface of STASIS (including a voice-activated hint system). Group #3 built a "self-destructing" mission briefing gadget for the players. Besides also creating two puzzles, group #4 built two automatic sliding doors which were remotely controllable. These doors were essential, as now a hybrid puzzle structure [20] could be achieved by combining sequential and path based puzzles to solve the meta puzzle: 'open next door'. Combined with puzzles built by the remaining groups the following walk-through of the game was created.

After the mission briefing the game starts in the lobby room which functions as a gateway so that the players can solve the first puzzle in an isolated environment. In order to enter the second room, which is the main laboratory, a key-code has to be found using a Morse code hidden in a portable music player. To light up the main laboratory all players need to hold multiple switches concurrently. This also triggers closing the door to the lobby, which traps the players. The secret research prototype is locked in a safe (which is only activated once the players are trapped) that the players need to open by powering it up and subsequently finding the lock's combination. The AI then has to be tricked with a fake prototype, as it is able to recognize when the prototype is removed from the safe. In parallel, different puzzles like opening a locked fuse box with a (mock) high-energy laser and cooperatively navigating a robot to push a button, need to be solved to progress to the next level. In the server room the prototype has to be uploaded by solving a sliding picture, and a maze puzzle. Afterwards the AI has to be deactivated in order to reopen the main door which can be achieved through solving a modified *Simon Says* puzzle. If all puzzles are completed within a one-hour time-limit, the players will be informed that they beat the game. An in-depth explanation of each puzzle would go beyond the scope of this paper. However, a selection of puzzles are explained in more detail in Section 3.

3 REALIZATION

We decided for the following shared hardware and software principles as they benefit successful work on the project and approximate a professional engineering environment. During the pilot-run these principles proved useful.

We chose a microcontroller family that has built-in WiFi functionality and is supported by the open source *Arduino* development platform. Arduino has a large community and can be used by students who only have basic programming skills as well as more experienced students. A common software platform also simplifies intra- and inter-group collaboration, as code can be exchanged. The choice was limited to those microcontrollers so that students are able to support their peers, and to bundle the support required from instructors. Arduino allows to draw advantages from a large set of compatible libraries for many sensors, actuators, or protocols such as MQTT. Therefore, the focus of the students could be directed to finding creative solutions or developing new puzzles, rather than on how to implement standard protocols or drivers for common hardware. One example is the over-the-air update, which requires only minimal modifications to a user's program. This is especially helpful since microcontrollers can be wirelessly re-programmed in their final locations, which can be hard to reach once installed, e.g. because they are mounted in the wall or in an enclosed component.

As development of individual puzzles and other components is an ongoing process, we decided to require each group to maintain their own git repository (see [18]). This helped us to track the week-to-week progress of each group, and identify projects with single contributors. Beyond version management, issue tracking also helped in intra- and inter-group work management. One group managed their whole project by tracking tickets on github, and groups with interdependencies communicated through the platform, for example if incompatibilities or problems with a puzzle were found.

As pointed out in Section 2.1 we chose stability over innovation, hence we decided for a classic client-server communication model. Each puzzle acts as a client which directly communicates with a central server. All communication is relayed via this server, which simplifies the identification of errors and allows to log all messages at a central location. For simplicity, stability and robustness we opted for WiFi as the only physical network interface. We chose

Figure 2: The operator interface of the escape room. The players' progress in the room is shown, levels can be skipped, and components in the room can be remotely controlled.

MQTT [17] as the network protocol, which provides a standardized publish-subscribe messaging system for networked, embedded devices. The central MQTT broker retains the current state of the escape room, and puzzles which reset for any reason will then get the current state any time they reconnect to the broker. We additionally defined the following hard requirements for each component.

Hardware Requirements:

- WiFi-enabled microcontroller, ESP8266 or ESP32
- Robustness to withstand frequent use by players
- No batteries (easier maintenance)

Functional Requirements:

- All communication must be based on MQTT
- Each puzzle must be skip- and reset-able
- Each puzzle must work out-of-the-box after power-on
- Each component must support over-the-air updates
- Each puzzle must provide a set of hints (via the AI)

3.1 Students' Puzzles

As noted before, the operator interface, the puzzles and all other components such as the smart lighting and automatic doors were designed and implemented by the students from scratch. All 18 developed components are visualized by markers in Figure 1. To provide a feeling of the complexity of the projects we show five of the student's prototypes in Figure 3 and describe three of them in more detail.

#1 Operator Interface: One group of students developed a web application which allows spectators and an operator to monitor and control the escape room (see Figure 2). The operator can intervene in the game, communicate with the players and provide hints. The state of all puzzles (*unsolved*, *solved*, or *skipped*) is displayed, and can be altered if necessary. The doors and lighting can be controlled manually, even while the game is running. This was particularly helpful during test runs, when incomplete puzzles needed to be skipped or reset. This group had a central role for the whole project, since they were responsible for the component which coordinates all other components and as such had to coordinate with the whole

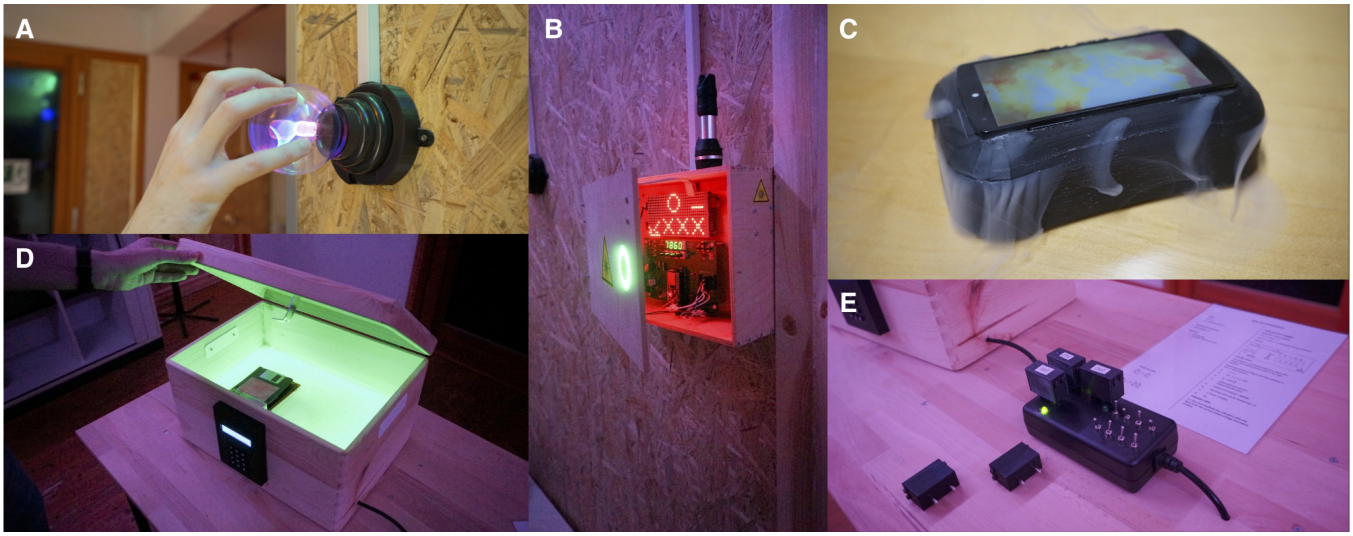

Figure 3: Examples of the students' puzzles. — A: One globe of the globes puzzle. — B: Fuse-box containing the door control mechanism. — C: Mission briefing device containing a smoke generator. — D: Safe with the secret prototype inside. — E: Safe power activation.

course to decide on the common communication protocol and the game's sequence.

#2 The First Door: Typical escape rooms start with an operator who locks the players inside one or multiple rooms. Since our goal was to automate the escape room as much as possible, we required a mechanism to make sure that all players are inside. The group with this task developed a *dead-man-switch*-like puzzle which requires each player to touch a plasma globe as shown in Figure 3.A. When all globes are touched simultaneously, i.e. all players are in the second room, the door closes automatically and the players are locked. Touch is detected by monitoring the energy consumption of off-the-shelf plasma globes. This project showcases the communication complexity between the puzzle and the student groups. The touching state of all globes needs to be transmitted to the central component, which then commandeers the door to close as long as the globes stay touched. Once the door is closed, the game needs to progress to the next level and the players can release the globes. A single developer can easily implement this interaction, however due to the division of projects this task alone required more than five students to coordinate themselves. Hence it is a good example of learning about coordination in a real project group.

#3 High-Energy Laser: Entering the third stage of the escape room requires to unlock the door to the "server room". Access to the door's control mechanism (Figure 3.B) is granted after "melting" a lock with the (mock) *high-energy laser*. A casing for an off-the-shelf laser pointer was 3D printed to fix the laser to one of the walls. Players have to redirect the laser with multiple mirrors to a phototransistor. To prevent that any other light source can be used to solve the puzzle (which was a "hack" applied by a player during one of the integration runs), the laser pointer is pulsed with a certain frequency. Only if this frequency is detected for a certain period of time the puzzle is solved and the door opens. The student group showed particular creative engineering skills as they built all the casings, while also figuring out how to best indicate that the laser is pointing at the right spot without frustrating the players. This example also highlights that design issues were revealed early on during the weekly integration runs.

4 OUTCOMES

The course was generally considered a success by students, instructors and players. While there were still some minor problems with some of the puzzles, the room was fully playable by external visitors and all parts of the rooms communicated with a central operating unit. Towards the end of the course, we specifically tried to avoid substantial last minute changes from any of the groups, and a complete run of the room was possible even a week early.

Some students regularly requested access to the room and the technical facilities outside of normal course hours, suggesting a high motivation to finalize their individual projects. This motivation was accelerated by the fact that most of the students wanted to invite their friends to test the escape room during the final presentation. Some groups were even able to go beyond their initial project specifications and added supplementary functionality. The escape room was opened for a total of four days, and was successfully played by over 50 members of the faculty, and by family members and friends of the students.

During a follow-up, we asked the students to evaluate the course and give feedback in an anonymous questionnaire (see [18]). The questionnaire was answered by 15 out of the 20 course participants (75 %). It included questions to gauge intrinsic motivation [13], and all questions were answered on a seven-point Likert scale. In the *interest and enjoyment* sub-scale, which is considered a self-report measure of intrinsic motivation, a mean score of 6.5 ± 0.4 out of 7 was reported. This suggests a high intrinsic motivation of the students which is reflected by the observations reported above, and our impressions. The weekly progress reports and integration runs were well accepted. Students agreed that the integration runs were useful (6.4 ± 1.1), and that the weekly progress reports were a motivation to engage with other students (5.6 ± 1.2). The theme of an escape room as a course topic was generally considered positive. Students also reported a higher motivation to engage in the course because of its unique theme (5.6±1.3), and they felt like they learned more than in other courses (5.5 ± 1.5). All students agreed that they can apply their acquired knowledge in other tasks in the future (6.1 ± 0.8), reported that they improved their skills in scientific problem solving (5.5 ± 1.4), and that the course work load was manageable (5.7 ± 1.2). Regarding the learning goals the students confirmed that the course contributed to their understanding of embedded systems (6.4 ± 0.7) and that they have learned a lot in the course (6.1 ± 0.9).

Inter- and intra-teamwork was particularly liked by the students and, as one student puts it, "doing something on a small group and then integrating it into a bigger system" was found to be enjoyable. Also that "compared to other courses it was really nice to have something in the end for other people to enjoy", which means that the projects are not just presented at the end of the course, but can be used to entertain a larger audience. This common goal for the whole course creates a learning effect which is unattainable with individual projects, and teaches the skills to contribute to large projects.

As a fair students' assessment can be hard for project-based courses, all instructors graded each group based on different aspects: communication inside the group and with other groups, competence, robustness of their components, and quality of documentation and final presentation. We considered good communication between groups essential for a working escape room, as many components of the escape room will have to interact. As we established the concept of stability over innovation the robustness of components was also a major requirement. The overall grade of each group was derived as the mean of all sub-grades from all instructors.

There are also some limitations to this course concept. A comparably large budget of 3200 € was available, but it should be noted that more than half of the budget was invested in preparing the rooms and buying hardware kits, which can be reused for each new iteration of the course. We aimed for robust puzzles and a solid, future-proof room infrastructure. Expenses could however be drastically reduced if the quality of appearance is sacrificed, rooms are used as they are, and puzzles are constructed from reusable parts only. Furthermore, the tutoring effort and time investment can be considerably higher than in other courses. The rooms need to be properly prepared for the students to be able to realize their projects. For example, a camera and microphone setup should be installed early on, as during the integration runs only a small amount of students will be able to enter the room, whereas the rest of the students need to be able to follow the progress and identify problems remotely. If this work cannot be done by one of the student groups, it may need to be implemented by the instructors. Once installed, this basic infrastructure can be reused for multiple semesters. Moreover, the inter-dependability of the individual components of the room to form a coherent and unified system requires that all student groups can adequately solve their problems and develop their projects. If this is not the case for just one group, or if students unexpectedly leave the course during the semester, it may be necessary for instructors to step in and take over parts of the projects in order to ensure a successful completion for the whole course.

Scaling the course to accommodate more students presents a challenge, since a reasonable students-per-supervisor ratio should be maintained and the room size sets a physical limitation. For example we could have accommodated a maximum of 30 students with a room size of 35 m^2 and six instructors.

5 CONCLUSION

We have described the *Ubiquitous Computing Lab* course where the students constructed an automated escape room. It was met with great enthusiasm. Student feedback indicates that they learned a lot and believe that they will benefit from the gained knowledge in the future. Even though this was the first time the course was offered, it concluded with great results. The system engineered by the students was stable enough to open the room for guests and members of the faculty. This gave the students the opportunity to invite friends to play the room. The escape room was demonstrated to the dean's office and was considered for advertising the university to high school students. Those plans, however, were put on halt due to the ongoing pandemic. Compared to project-based courses in which students have to complete individual group projects, a common goal for the whole course also teaches soft skills required for contributing to large projects and increases student motivation. Building such a technical system in a student course is a considerable risk. We were pleasantly surprised by the high motivation of our students and how well weekly integration runs worked for realizing the escape room. It remains to say that while considerable effort was put into this lab course, we think that it is worthwhile as it is a unique opportunity for students to gain practical skills.

6 ACKNOWLEDGMENTS

We would like to thank all the students who eagerly participated in the Ubiquitous Computing Lab course during the winter semester 2019/2020 at the University of Freiburg. We would also like to especially thank the technical faculty management at the University of Freiburg for giving us the opportunity to convert former office spaces to the escape room we presented here. Finally, we thank Prof. Dr. Bernd Becker for the support during the creation and execution of this new lab course concept.

REFERENCES

[1] Farha N. Ali. 2018. Experiences in Teaching the Internet of Things Courses. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 378–383. https://doi.org/10.1145/3159450.3159549

[2] Mordechai Ben-Ari. 1998. Constructivism in Computer Science Education. *SIGCSE Bull.* 30, 1 (March 1998), 257–261. https://doi.org/10.1145/274790.274308

[3] Carlos Borrego, Cristina Fernández, Ian Blanes, and Sergi Robles. 2017. Room escape at class: Escape games activities to facilitate the motivation and learning in computer science. *Journal of Technology and Science Education* 7, 2 (2017), 162–171. https://doi.org/10.3926/jotse.247

[4] Gaetano Borriello, Carl Hartung, Bruce Hemingway, Karl Koscher, and Brian Mayton. 2008. Multi-Player Soccer and Wireless Embedded Systems. *SIGCSE Bull.* 40, 1 (March 2008), 82–86. https://doi.org/10.1145/1352322.1352165

[5] Barry Burd, Lecia Barker, Félix Armando Fermín Pérez, Ingrid Russell, Bill Siever, Liviana Tudor, Michael McCarthy, and Ian Pollock. 2018. The Internet of Things in

Undergraduate Computer and Information Science Education: Exploring Curricula and Pedagogy. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018 Companion)*. Association for Computing Machinery, New York, NY, USA, 200–216. https://doi.org/10.1145/3293881.3295784

[6] Samantha Clarke, Sylvester Arnab, Helen Keegan, Luca Morini, and Oliver Wood. 2016. EscapED: Adapting Live-Action, Interactive Games to Support Higher Education Teaching and Learning Practices. In *5th International Conference on Games and Learning Alliance (LNCS, Vol. 10056)*. Springer, 144–153. https://doi.org/10.1007/978-3-319-50182-6_13

[7] Janet Davis and Samuel A. Rebelsky. 2019. Developing Soft and Technical Skills Through Multi-Semester, Remotely Mentored, Community-Service Projects. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 29–35. https://doi.org/10.1145/3287324.3287508

[8] Audrey Girouard, Andrew L Kun, Anne Roudaut, and Orit Shaer. 2018. Pervasive computing education. *IEEE Pervasive Computing* 17, 4 (2018), 9–12.

[9] Victoria Hong and Elana Reiser. 2020. Make Your Escape: An Active Learning Approach. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 1410. https://doi.org/10.1145/3328778.3372557

[10] Michael S. Kirkpatrick. 2017. Student Perspectives of Team-Based Learning in a CS Course: Summary of Qualitative Findings. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 327–332. https://doi.org/10.1145/3017680.3017699

[11] Andrew L. Kun. 2017. Textbooks for Pervasive Computing Training and Education. *IEEE Pervasive Computing* 16, 3 (2017), 59–61. https://doi.org/10.1109/mprv.2017.2940951

[12] Hanna Mäenpää, Samu Varjonen, Arto Hellas, Sasu Tarkoma, and Tomi Männistö. 2017. Assessing IOT Projects in University Education: A Framework for Problem-Based Learning. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track* (Buenos Aires, Argentina) *(ICSE-SEET '17)*. IEEE Press, 37–46. https://doi.org/10.1109/ICSE-SEET.2017.6

[13] Edward McAuley, Terry Duncan, and Vance V. Tammen. 1989. Psychometric Properties of the Intrinsic Motivation Inventory in a Competitive Sport Setting: A Confirmatory Factor Analysis. *Research Quarterly for Exercise and Sport* 60, 1 (1989), 48–58. https://doi.org/10.1080/02701367.1989.10607413

[14] Julie E Mills and David F Treagust. 2003. Engineering education – Is problem-based or project-based learning the answer. *Australasian journal of engineering education* 3, 2 (2003), 2–16.

[15] Christian Murphy, Swapneel Sheth, and Sydney Morton. 2017. A Two-Course Sequence of Real Projects for Real Customers. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 417–422. https://doi.org/10.1145/3017680.3017742

[16] Scott Nicholson. 2015. Peeking behind the locked door: A survey of escape room facilities. http://scottnicholson.com/pubs/erfacwhite.pdf

[17] OASIS. 2019. MQTT Version 5.0. https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.pdf

[18] Marc Pfeifer, Benjamin Völker, Sebastian Böttcher, Sven Köhler, and Philipp M. Scholl. 2020. UbiLab - Escape Room. Retrieved November 19, 2020 from https://github.com/ubilab-escape/integration

[19] Iris Talmi, Orit Hazzan, and Reuven Katz. 2018. Intrinsic Motivation and 21st-Century Skills in an Undergraduate Engineering Project: The Formula Student Project. *Higher Education Studies* 8, 4 (2018), 46. https://doi.org/10.5539/hes.v8n4p46

[20] Alice Veldkamp, Liesbeth van de Grint, Marie-Christine Knippels, and Wouter van Joolingen. 2020. Escape Education: A Systematic Review on Escape Rooms in Education. Preprints 2020, 2020030182. https://doi.org/10.20944/preprints202003.0182.v1 preprints.org.

[21] Tom Wulf. 2005. Constructivist Approaches for Teaching Computer Programming. In *Proceedings of the 6th Conference on Information Technology Education* (Newark, NJ, USA) *(SIGITE '05)*. Association for Computing Machinery, New York, NY, USA, 245–248. https://doi.org/10.1145/1095714.1095771