KTH Royal Institute of Technology

Walk Over Me Interactive Floor Controller Application

Developing Mobile Applications

David Ekman
Iza Grasselli
Eva Maria Veitmaa
Salim Yahya

Stockholm

2019

Table of Contents

Introduction

Walk Over Me is an interactive touch-sensitive light-up floor designed to support and inspire physical play. The WalkOverMe mobile application (WOM-app) is a remote controller that enables customising how the actual physical floor works, uploading new games to it, using it as a drawing canvas, and creating and viewing high scores.

An additional goal of the WOM-app is to teach players about programming and technology by encouraging them to design, create, and share their own games. However, due to time constraints this functionality is not present in the version developed during the course.

The current version features a mocked Bluetooth connection, the possibility to upload a selection of pre-designed games (“Draw”, “Whack-a-Mole”, “Tic-Tac-Toe”) to the floor, mock-play the game, save and view the scores in a global database using a web service, and check which game is currently uploaded to the physical floor.

Various user tests carried out during the design and development phase of the application show the need for the WOM-app controller. The app’s design and functionality have been strongly inspired by feedback gathered from continuous user tests. The end result for this course is a fun and playful app that solves an actual problem regarding the use of a physical playground product.

Concept and Market Viability

The Walk Over Me floor consists of multiple independently **programmable tiles** that are controlled by an Arduino microcontroller (see Figure 1). By default, the floor only supports one game at a time. Every time the user wishes to change the game, the floor needs to be reprogrammed.

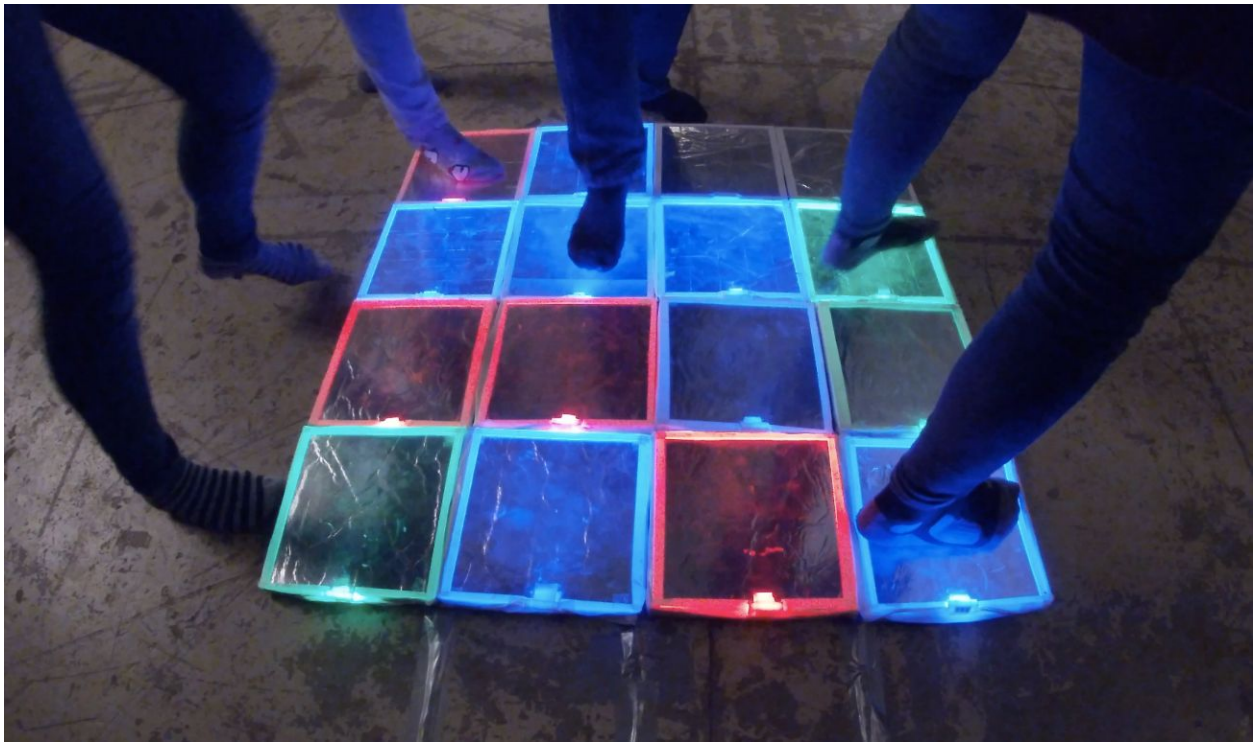

Figure 1. The physical prototype of a Walk Over Me floor consists of multiple pressure-sensitive tiles that can be independently programmed using an Arduino microcontroller.

The Walk Over Me floor is mainly targeted for parents, kindergartens, schools, and other institutions that come into close contact with children and playground management. Not all of the users in those target groups are familiar with **programming an Arduino microcontroller**, however, although kindergarten teachers are facing an increasing pressure of digitalisation [1, 2]. This means that to take full advantage of what the interactive floor has to offer, the users need some kind of an easy-to-use control device. This could be a stand-alone remote controller, a built-in control screen, or a mobile application.

To enable this kind of **remote control of the interactive floor**, the WOM-app was built. The app is mainly created so that less tech-savvy users who do not feel comfortable with programming could **change the game** that has been uploaded to the floor without having to learn how to code. Additionally, this controller application also makes it more convenient for tech-savvy people to use the Walk Over Me floor. For them, there is no need to write actual code anymore, although the app does support **uploading custom programs** to the floor.

Currently, the Walk Over Me floor only supports open ended play. This means that the touch-sensitive tiles recognise when someone steps on one, and change colour after each press. The colour is programmed to rotate through red, green, and blue and fade out after 5 seconds.

Although children have been enjoying this open ended play with no specific rules, the parents have often taken interest in which games the floor supports and how to shift between them. The

children we have tested the floor with have shown interest in **competitive gaming and saving their high scores**. This shows that there is a need for playing and creating more games, scoreboards, and a controller that enables switching between the games.

The WOM-app is not intended as a separate financially profitable stand-alone product, but an **additional supportive element for a physically tangible item**. From our preliminary user research, we have received a lot of positive feedback and willingness to buy the product. One large robotics company (FeelRobotics) was even interested in collaborating with us on turning our current prototype into an actual product that they could install outside their shared office building to "see all the boring bankers have some fun during their lunch break"[1].

Therefore, both Swedish **individuals and municipality playgrounds have shown interest in buying** the Walk Over Me floor. However, based on our preliminary interviews and observations, we strongly believe that to satisfy this market need, the floor needs to be coupled with an easy-to-use and expandable controller application.

Design Process

Our design process was inspired by the paper "Design and Implementation of a Game Interface Interaction on Smartphone" to guarantee operability, playability and appreciation [3]. We **worked iteratively**, creating paper, low-fidelity, and high-fidelity prototypes, and relied strongly on the **feedback from the users** in our test group. We started with preliminary user research, mapping out the needs of potential customers. Then we created paper prototypes which we modified based on user feedback. The low-fidelity wireframes and the web app enabled us to get additional input which we used for creating the final high-fidelity application prototype for Android operating system.

Preliminary User Research

The Walk Over Me interactive floor was presented at a **demo day** of the Physical Interaction Design and Realisation course at the Reaktorhallen at KTH ("R1") with the default open-ended game loaded on it. We observed people using the floor both alone and in groups and conducted **unstructured interviews** with visitors regarding their experience. Often visitors came to us to give feedback or ask questions themselves.

Required Functionality

One of the main questions we received was about the **rules and how the game should be played**. Since the uploaded code did not feature any specific game, but simply supported changing the colour of the tiles, there were no set rules. It had intentionally been designed to support open play and players coming up with their own games and rules. However, we

[1] Quote from the CTO of FeelRobotics, Maurice Op de Beek

gathered that grown-ups are used to having pre-set rules and guidelines meaning that there should be an option to have **rule-based games**.

Additionally, visitors often asked about various different games, such as Whack-a-Mole, Tic-Tac-Toe, or turning all the tiles the same colour. This indicated that the floor should have a **large selection of games to choose from**.

Another thing we noticed was that despite lacking rules, the visitors easily managed to **come up with their own games and interactions**, e.g. unlocking special modes or animation rewards with geometrical patterns or ordered presses. This inspired us to support the user in designing and creating games for the floor themselves using a **game creator functionality**. A parent even suggested holding game design competitions with financial rewards as these could encourage both parents and children to get more into technology and programming. Supporting users in designing their own games is also aligned with the Swedish kindergarten curriculum's requirement of developing digital skills[2]. The idea of coupling physical products with the goal of **teaching how to code** is not new, as was also demonstrated by Nilsson and Åkervall who, analogously to us, used programmable car-like toys [4].

Children engaged in a lot of **competitive games** which is why we see it fit to have scoreboards that connect users of the Walk Over Me floor all around the world. Since the floor has been designed for people of all ages, some of whom may not yet own a smartphone, we proposed a possibility to save the score under any name, independent of the user account.

The visitors also pointed out that when there is no one playing on the floor it can also be used as a **decorative element** featuring various animations. Due to this we incorporated the Draw game as one of the options. The Draw game features a digital representation of the floor that acts as a **pixelated canvas**. By tapping on the digital tiles, the user can change the colour of each individual physical tile and, thus, create their own patterns.

Controller Type

Once we had come to the conclusion that the Walk Over Me floor needs to support the user in those goals by having some kind of a controller, we conducted **follow-up interviews** regarding how that controller should look and feel like. During the follow-up interviews we used **video footage of the interaction** with the physical floor to illustrate the use case better. We did not use the actual physical prototype as it was damaged from the exposition at the Reaktorhallen and we had not managed to repair it yet. Most of the participants for our follow-up interviews had experienced the Walk Over Me floor during the exposition at R1. However, we do believe that for those who had not seen the floor firsthand, having a physical prototype present would have helped illustrate the scenario better.

From these interviews, two main concepts emerged. First was a **separate dedicated controller device** that comes along with the floor (similar to a TV remote). Alternatively, the interviewees

[2] https://www.skolverket.se/undervisning/forskolan/laroplan-for-forskolan/laroplan-lpfo-18-for-forskolan

suggested using **already existing digital smart devices**, such as mobile phones, tablets, or laptops. We decided to go with the latter due to the following reasons:

- Smart devices are common and most people we interviewed owned at least one.
- Using already existing devices is more sustainable as we avoid manufacturing new ones.
- Releasing updates for mobile applications is easier and more scalable, thus providing better user experience and almost unlimited options for expansion.
- Developing a mobile application carries a lot of costs in the initial phase (R&D), but little overhead costs in later stages.

Therefore, as a controller for the Walk Over Me floor, we decided to create a **mobile application coupled with the physical floor**.

Similarly to the app-controlled lock system presented in the paper “Arduino Based Auto Door Unlock Control System by Android Mobile through Bluetooth and Wi-Fi” [5], for our Walk Over Me floor that also uses an Arduino microcontroller, we decided to go with a **Bluetooth module** to connect to the physical floor.

Bluetooth was preferred over local WiFi or static IP-address for a variety of reasons.

Firstly, Bluetooth requires the user to **be near the floor to be able to control it**. Otherwise, someone could change the currently uploaded game from a distance and ruin the experience for the players.

Additionally, Bluetooth enables **connecting only a single device to the floor** simultaneously. On the positive side, this means that only one person is in control of the Walk Over Me floor and no one without access to the connected phone can interrupt the game while it is in progress. On the downside, the first person to connect to the Walk Over Me floor has a **monopoly** over it. However, we are hoping for people’s kind-heartedness and willingness to share.

Bluetooth also enables **showing only a specific type of objects** in the selection of possible devices to connect to. As we are surrounded by many Bluetooth devices nowadays, we see it as beneficial for filtering out only the Walk Over Me floors and making the connection process easier for the user.

It is important to note that Bluetooth has a relatively **low energy consumption**. Although Saborido et al did not find this aspect to be correlated to the application’s ratings [6], our interviews indicated that users prefer their mobile phone batteries to last longer, especially when the app is being used passively in the background. As one interviewee put it, “I do not wish to live my life connected to the wall.”

In addition, Bluetooth connection as any other external connection needs to be **secure to guarantee the privacy of the user and their data** [7, 8]. This, however, will be out of the scope

of our project in its current phase as we will be mocking the Bluetooth functionality, but definitely something to keep in mind for future developments.

Paper Prototype

The paper prototype was created to quickly gather feedback about the functionality of the proposed application. The prototype was created **modularly**. Different screen views could easily be switched by replacing pieces of paper representing individual screens (see Figure 2). This enabled **simulating how the app would respond** to the user's actions.

The paper prototype featured a top bar with the application's logo and a bottom bar with links to three different tabs. Initially, the tabs were:

- "Games" - for presenting a list of games that the user could upload to the physical floor;
- "Home" - for showing the current state of the floor, i.e. which game has been uploaded;
- "Draw" - for enabling pixelated drawing functionality.

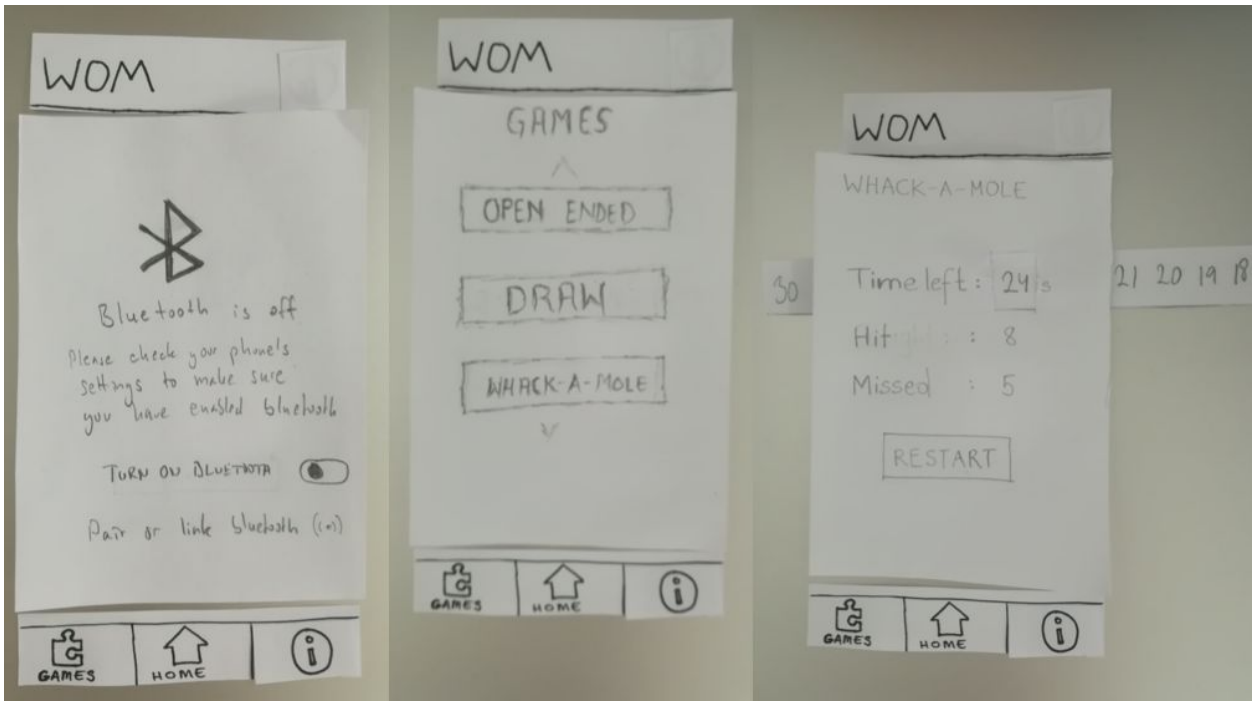

Figure 2. The screens of the paper prototype were modular and enabled us to quickly gain initial feedback.

We started with **expert reviews and testing** amongst ourselves. During the testing, the facilitator would **switch out the modular paper scenes** to illustrate the workings of the app. This worked well and even supported AB-testing with comparative screens, although the

process was slower than it would have been with a digital wireframe as the facilitator spent time on finding the correct screen corresponding to a specific button press.

As a result, we moved the “Draw” tab under the “Games” tab in order to represent a game option instead of a separate action. This was done so that when the user moves from what used to be the “Draw” tab to the “Home” tab, the page layout would change to **indicate the app’s responsiveness**. Otherwise, the “Draw” and “Home” tab screens would have looked the same which was confusing.

Based on our experience with existing applications, we decided that users should have the possibility to **explore the app even when they are not connected to the floor**. This enables them to decide whether they even want to invest in a floor, as well as to use additional (future) functionality, such as designing your own games, without having to have the actual floor nearby. This design decision was further confirmed during user testing. To enable this exploration, we chose the connection screen to be the first screen of the app but also presented the user with the bottom menu bar, which enables exploration of the app without having to connect to the floor. If a connection has not been made, the connection screen appears when the user tries to upload a game to the floor.

Wireframes

Based on the paper prototype, we created wireframes using Balsamiq (see Figure 3). Balsamiq was chosen because it is **free and allows online collaboration** of multiple project members. It also offers a number of **readily available UI elements** which we could use in our wireframes. The elements are simple and allow for simple interactions. Therefore, in our wireframe designs, we **focused on functionality rather than the looks**.

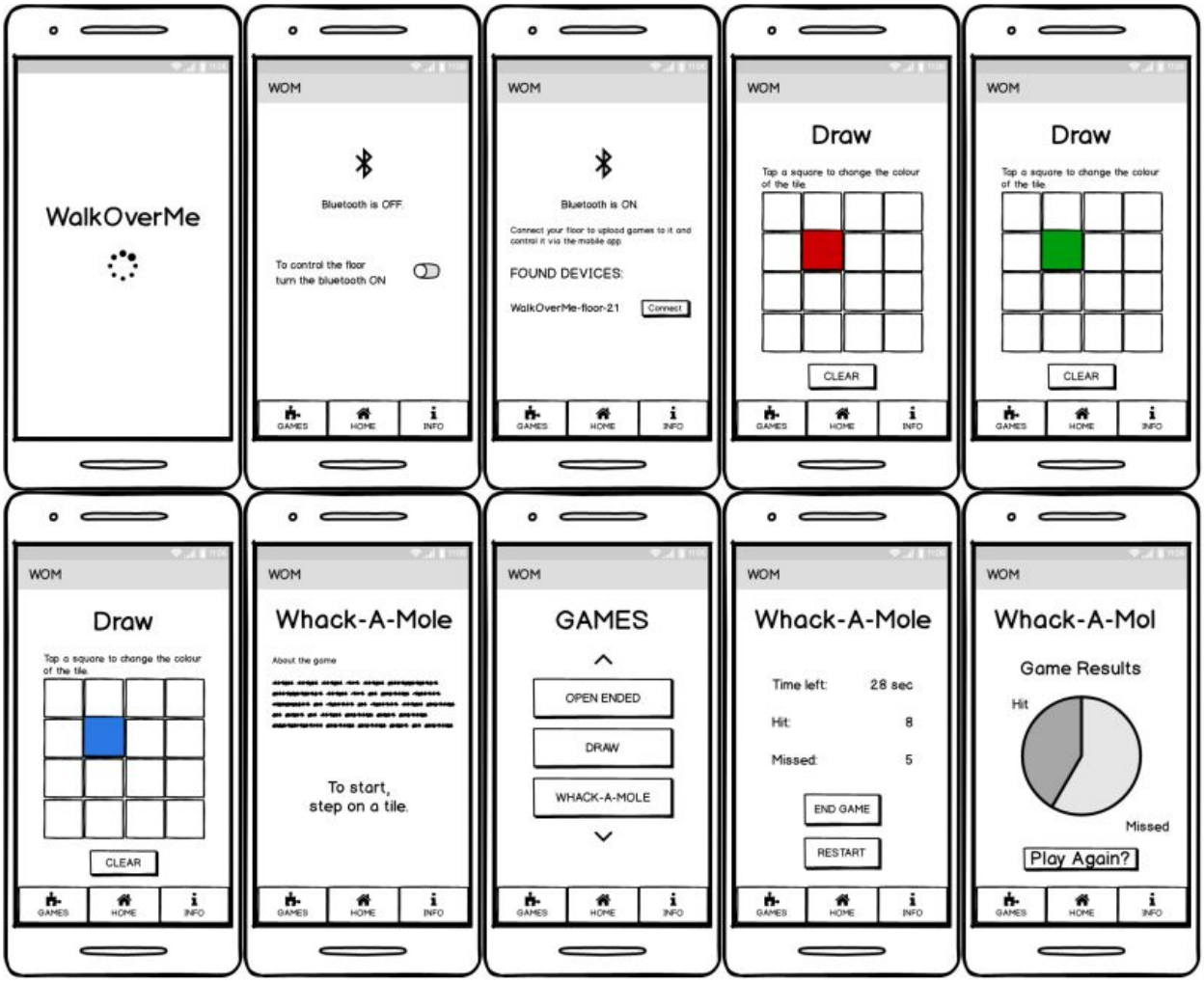

Figure 3. Wireframes of the application were created using Balsamiq and feature a variety of screens demonstrating the functionality of the WOM-app.

We employed this wireframe prototype to get additional feedback on our app and perform design iterations. The participants for our user tests were our classmates who had experienced the floor during the demo day at R1. As we had some difficulties with setting up the Balsamiq on our phones, we **demonstrated the app from the computer screen** instead. We acknowledge that this reduces the real feel of the mobile app wireframe.

Using imaginary scenarios and storytelling, we established that the user would usually prefer **putting the phone away whilst playing** a game on the physical floor. This inspired us to create a prompt that stated that a game would start when the user stepped on the floor.

However, user tests revealed that a better option would be to **start the game by pressing a button** and having a countdown timer indicated on both the floor and in the app. This enables the user to put the phone away securely.

On the other hand, the users also expected to be able to **interact with the floor without having to use their phone** at all. Therefore, it is good to also have the possibility to either start a game by interacting with the floor only or to have the floor default to an open ended no-game setup.

In the end, we decided to employ both options. The user can control the floor both by stepping on it or from the app, including restarting the game mid-way and checking the end score. In case of the latter, one participant pointed out that if a phone is connected to the floor, it would be great if the **phone showed the score in an enlarged manner** on the phone screen when a game is started. This would encourage placing the phone near the floor and using the phone as an informative score screen without having to run back and forth between the phone and the floor.

One participant was confused about our **placeholder text** for the game instructions. Therefore, in the web app prototype, we decided to use actual text as much as possible. Additionally, participants were confused about why nothing happened when specific interface elements were used. However, that was due to the fact that not all games and functionality had been implemented in the wireframe.

While one of the users pointed out that they **like how simple and easy it is to interact with the app,** another participant pointed out the **lack of actual design**, such as a background image or colours. However, at this stage, we focused on establishing how well our app supported the intended functionality.

Participants suggested various additional features, such as creating their own games, saving the drawings and sharing them with others, or creating high scores and score boards and synchronizing them with a central unit so that they could compare their score to the high scores of every other WalkOverMe floor in the world. This demonstrates that people expect the app to also have a **social component** in addition to being a remote control for the floor.

Web App

The web app prototype was done using HTML, CSS, and JavaScript. We started exploring the jQMobile and jQTouch frameworks, but we did not feel the need for their advanced functionality at this stage. The web app was created with a **mobile first principle**, but thanks to a **responsive design template**[3] it was also usable on a tablet or a desktop computer (see Figure 4). As all of those devices have Bluetooth compatibility, this means that the user can use any of them to control the floor - in case of a web app, the remote controller for the floor does not have to be a smartphone.

[3] https://startbootstrap.com/themes/freelancer/

Figure 4. The responsive design works on various screen sizes, for example, iPads (left), iPhones (middle), and desktop screens (right).

We implemented the key features of the app, such as navigating between various screens (see Figure 5), (mock) uploading a game to the floor, and showing the in-game screen for some games, namely, Open Ended, Draw, and Whack-a-Mole.

Figure 5. Various screens of the mobile app as seen on a Galaxy S5 screen.

We developed our web app iteratively, taking into account the feedback from continuous tests. We **replaced the info tab in the bottom navigation menu with scoreboards** that show three best results for all the games. The extra information about the app can now be found by tapping on the info button in the top navigation bar.

The web app was tested with the same user group as in the previous session. In general, the app was easy to use and relatively intuitive.

Implementation as a Native Android App

We have chosen Android Studio as the framework for developing the application. It is **programmed in Java**, as opposed to Kotlin, because that is the language we are more familiar with.

Android studio was chosen because it has a wide online community support and elaborate, well written documentation. At the same time, it also has some downsides. It requires a lot of

processing power and memory, especially when using an emulator. Thus, we used an external mobile device for running and testing the application in order to assure a more timely and smoother coding experience. We also used some tricks to increase the speed of compiling the code, such as disabling the plugins that our app does not use.

Without Web Services

The application was created with a **bottom navigation activity**. We opted for bottom navigation because our design does not require many screens. Consequently, this means that the whole app **functionality is based on fragments**. This is good because the reloading is only necessary for the part of the screen that changes, which adds to the responsiveness of the app.

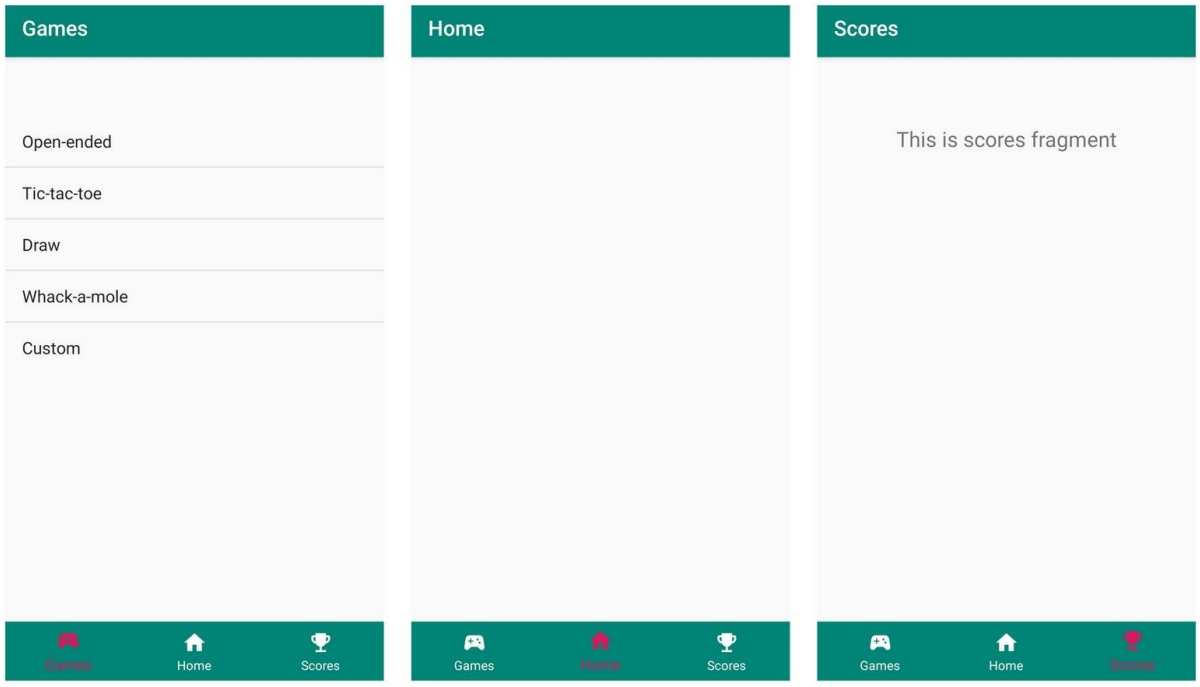

Figure 6. Bottom navigation bar enables switching between the views of "Home", "Games", and "Scores". Note that both "Home" and "Scores" views rely strongly on database connection which would be implemented in the next iteration.

The Games Fragment features a **scrollable list of games** that are available to be uploaded onto the interactive tiles. The list was implemented with a RecyclerView adapter because it is supposed to be a "more advanced and flexible version of ListView"[4], which was our initial choice. The list is responsive to clicks, which redirect the user to a respective game.

In this stage, we had **implemented two games**. The drawing game could be accessed by clicking on the Draw item in the list of games. The drawing canvas is made of 16 squares which change color on user click. The colors switch between red, green, blue and white, depending on

[4] Android Studio Guide. Create a List with RecyclerView. Retrieved from: https://developer.android.com/guide/topics/ui/layout/recyclerview

the number of clicks, which are monitored with onClickListener. The drawing canvas was declared in the layout file, and populated programmatically, using a RecyclerView combined with Grid Layout Manager.

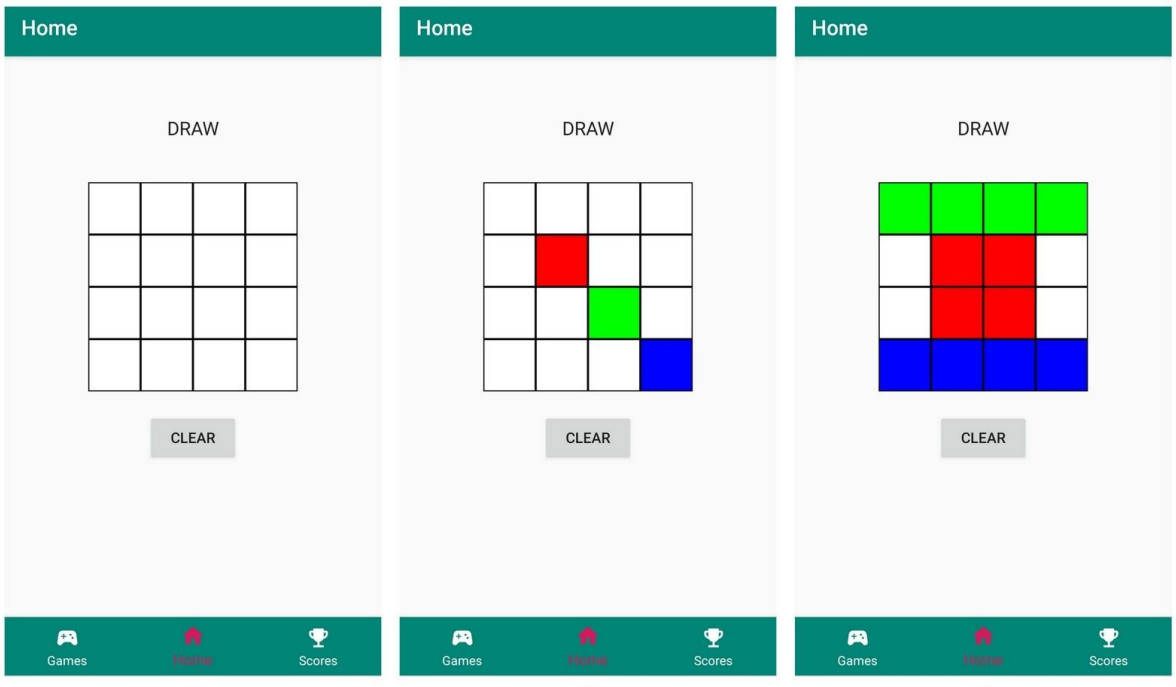

Figure 7. "Draw" is one of the possible games to be uploaded to the floor. The app creates a mapping to actual physical tiles, so that when the user changes the colour of a tile in the app, the colour of a physical tile changes correspondingly.

The second implemented game is called Whack-a-Mole. Because the game is not self explanatory, we first **present the user with an explanation**, after which they can start the game with a click on the START button. The game is set to last 30 seconds, and the user can see on the screen how many seconds are left. In the same table where the countdown timer is, the user can also see the number of "moles" they managed to hit. This was also their score. After the time runs out, a popup appears. It is implemented using AlertDialog and encourages the user to **enter their name, so it can be saved into the database together with their score**, which, once implemented, will be listed on the Scores screen. Since the connection to the Firebase was not fully working in this stage, the scores screen was planned to be implemented in the following stage.

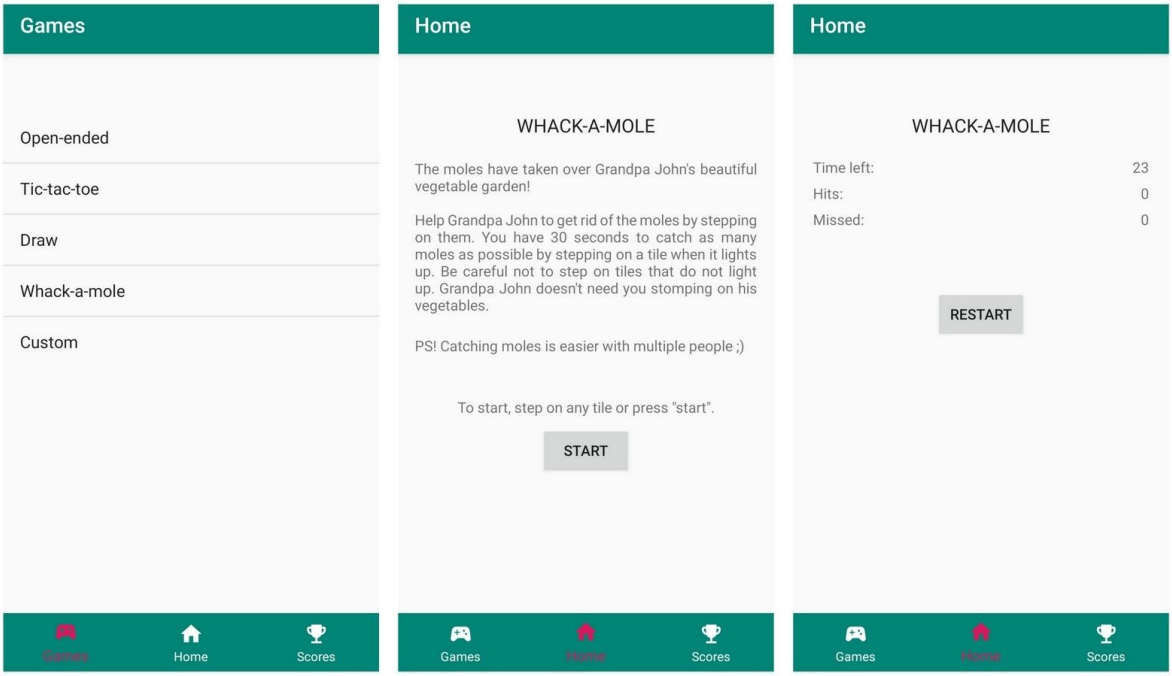

Figure 8. The user can choose which game they wish to upload to the floor from the "Games" view. Depending on the game, instructions are shown. In case of Whack-a-Mole, the app presents the game countdown and score screen after the game has been started.

Before the user loads the game on the tiles, the **Bluetooth connection has to be established**. Thus, upon a click on the game that is to be uploaded, the user is prompted to allow turning on Bluetooth and discoverability of the device. The two are required for the phone to find all available devices nearby. If the user allows these actions, a popup with found Bluetooth devices appears. It is implemented using a Dialog, and features a **ListView Adapter, which refreshes each time a new device appears**. Click on a device initiates pairing of the phone and the device to enable communication between them.

In this current stage, the actual communication with the interactive floor is not possible. The reason for this is that the physical floor itself has not been developed enough to support connecting to it via Bluetooth. Due to this and the time constraints, it is likely that **this feature of Bluetooth connectivity will not be supported in the final version** either. Instead, we plan to mock it to enable testing the functionality with the users.

User Tests

User tests were carried out with 3 other people taking this course. All of them were shown a **video** of the Walk Over Me physical floor prototype and each received an **explanation** of what the product is designed for before testing the app. This way they would have background knowledge about the Walk Over Me. The participants were presented with the following **scenario**:

> *“You have just obtained a Walk Over Me floor. You wish to explore what it has to offer via the corresponding WOM-app that you have just downloaded to your phone using the instructions on the product packaging.”*

During testing, the users pointed out that the **interface looks too plain and not playful enough** for what the app’s intended purpose is. This was something that we had planned to improve upon anyway, but the feedback gave confirmation for our suspicions.

Another thing the participants pointed out was the **lack of games**. Unfortunately, as most of the games presented in the “Games” view were not implemented, the users were unable to interact with them. However, we see these placeholders as a useful way to demonstrate that ideally there would be many games to choose from. It seemed that this aspect was also communicated to the users clearly and they were just disappointed that they could not actually try out those unimplemented games.

Of course, the user testing **lacked the feature of actually testing out the physical floor** in parallel which is something that we could do better during the next iteration. A possibility is to use a **non-functioning prototype**. The most minimal version of this would be simple sheets of paper or tape lines on the ground to indicate the tiles. Maybe this would give better feedback about how the app is to use when coupled with the physical floor.

With Web Services and Front-End Design

In the final iteration, the **user interface style was redesigned** to look more fun and engaging. We went with a light blue, white, and orange colour scheme and used rounded buttons for game selections to look more playful.

The UI styling can, of course, be improved further still, e.g. by using a premade template. Our goal was to try our hand at creating and programming a design from scratch to additionally improve our own skill set.

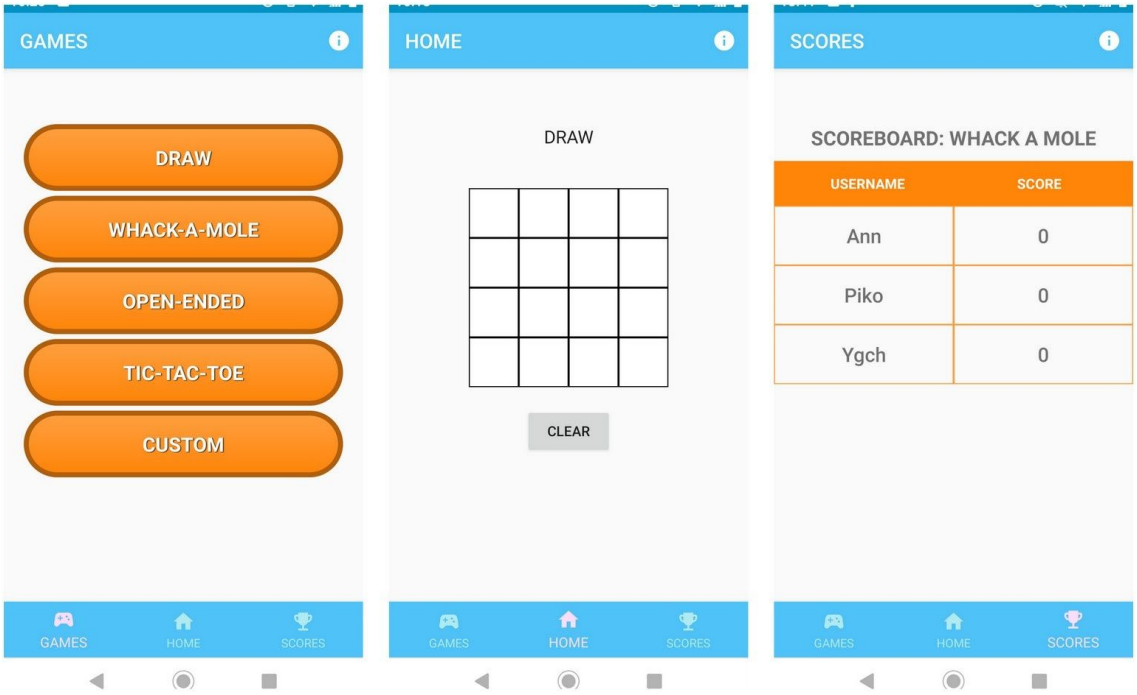

Figure 9. The app has three main menu items: "Games" featuring a list of games that can be uploaded to the floor, "Home" that shows currently uploaded game, and "Scores" featuring scoreboards of any given game.

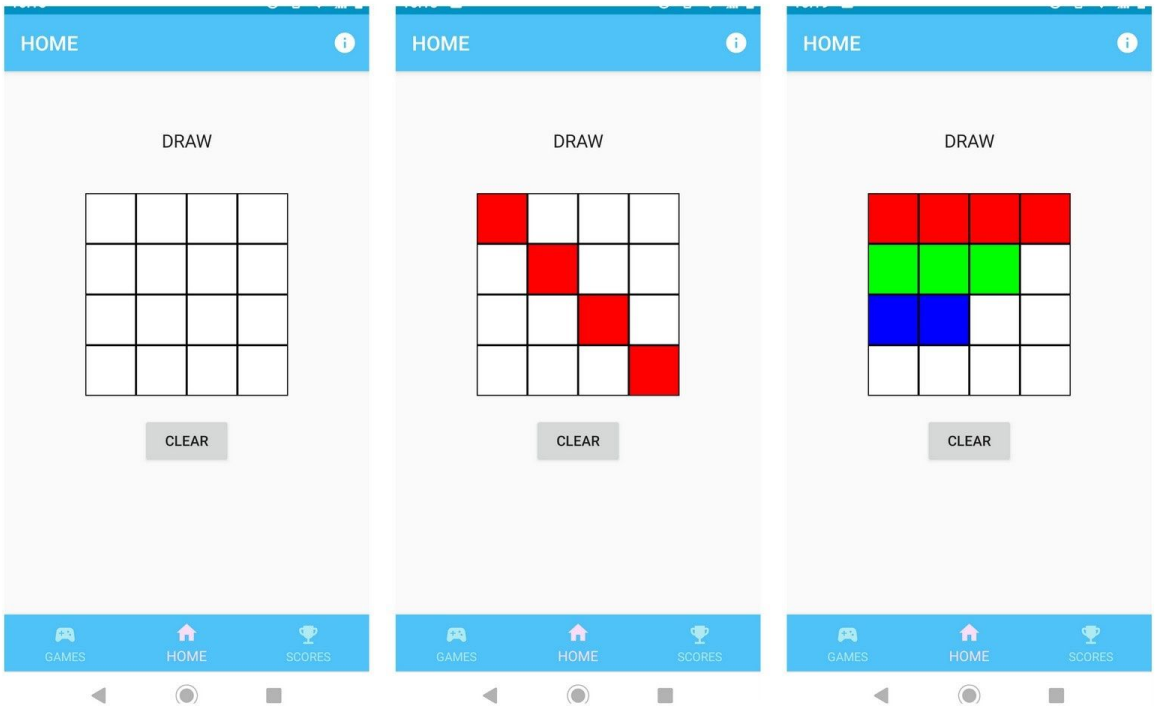

Figure 10. In the "Draw" game, corresponding tile on the physical floor lights up when a digital square is pressed.

Scoreboards

The application provides an option to **view scoreboards** and compare achievements of each game with other users. For a successful implementation of this scoreboard functionality, we decided to use Firebase Realtime Database.

Firebase Realtime Database stores information as JSON and offers real time synchronization[5] which fits the needs of our application. Especially, because it syncs across clients in real time and also retains data in times that application is offline.

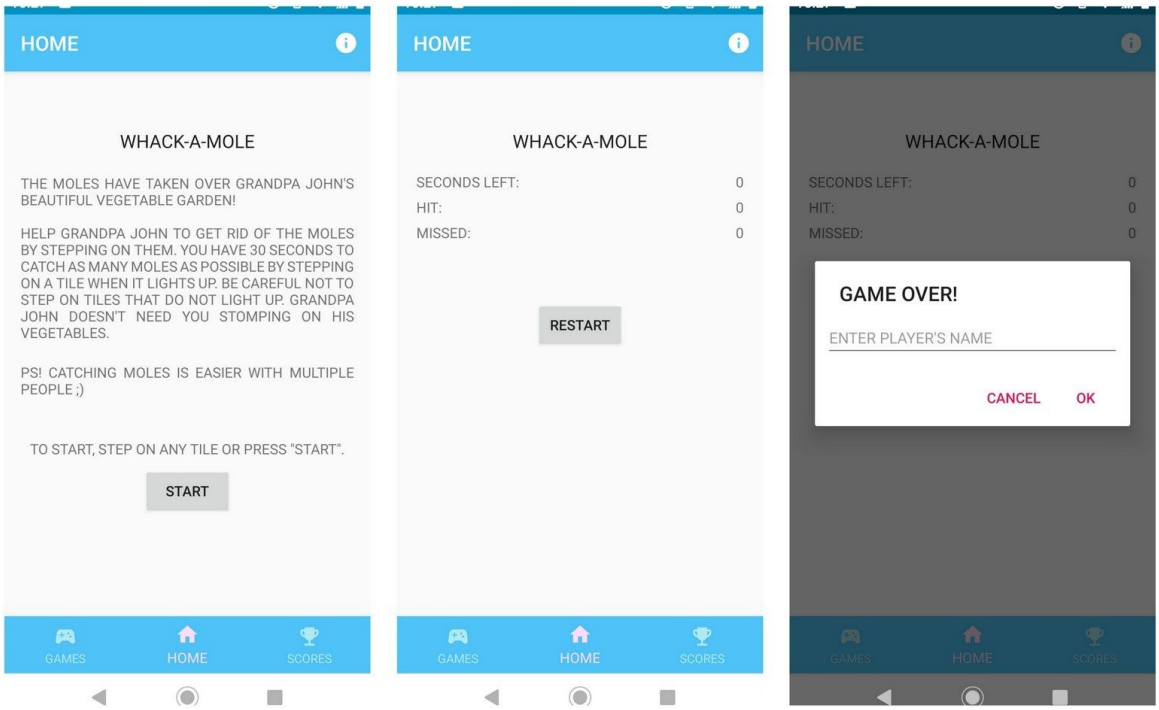

Figure 11. Some games have instructions or fun descriptions. During the game, the app shows the score. After the game, it is possible to save your score to the scoreboard.

Because our application does not require authentication, we configured the database for **public access**. We are aware that this design choice makes it possible for anyone to access it, even people who are not using the Walk Over Me app. However, we believe that this kind of risk need not require further attention at the moment. Mainly, because we assume the database can only be misused for cheating and that can also be done in other ways which can not be easily monitored, such as not following the rules when stepping on the tiles. Additionally, the player's name that is entered and visible on the scoreboard does not need to be the player's real name. It is up to the user to decide which name to provide. Scoreboard is simply a way to document

[5] Firebase Docs. Firebase Realtime Database. Retrieved from: https://firebase.google.com/docs/database

achievements and **does not contain any personal information** that would infringe on privacy unless the user themselves decides to enclose such information.

User Tests

The app was tested with three users who had previous experience with the physical Walk-Over-Me tiles and two who had not, and were thus showed a demo video of the tiles instead, to get familiarized with the concept.

The styling and colour scheme were met with great enthusiasm as opposed to previous user tests.

3 out of 5 users had a problem while uploading a game to the floor using Bluetooth. The “Games” tab of the app offers a selection of five games. When selecting a game, the user has to choose an appropriate device (Walk Over Me floor) from the list of available Bluetooth devices nearby. However, the users failed to choose any device because there was **no instruction** that would explain what they have to do. They said that the design itself does not indicate that devices can be clicked, hence, they did not know what to do just closed the window. We researched what is the best practice for solving this issue and realized that changing the popup title from “Available devices” to “Choose your WOM-floor” should be sufficient.

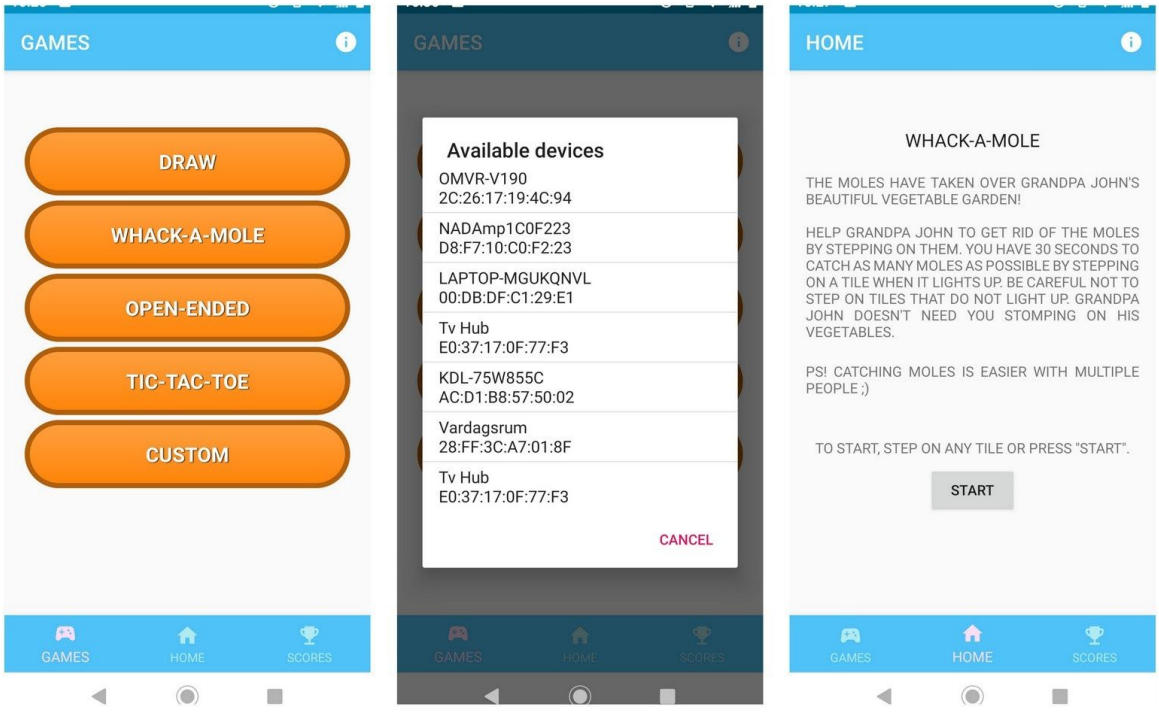

Figure 12. Uploading a game to the floor requires Bluetooth connection with the floor.

Additionally, one of the games, Whack-a-Mole, has a description that was reported as disturbing. The users either disliked the **uppercase letters** used in the description, because

they are “hard to read”, or considered the **text too long** and thus started playing without familiarizing themselves with the rules, which caused confusion during play. Our main argument for using uppercase had been that small children may be unable to read the instructions otherwise. This hypothesis will need to be researched further by letting the corresponding user group test the app.

The main conclusions from user tests were that the text designs need to be easier and more fun to read and that the user needs an even clearer indication on how to connect to the floor using Bluetooth.

Conclusion

In general, the project provided us with a great learning experience in how to design relevant and engaging mobile applications. By now, we have created a mobile application which enables the user to connect to and control a physical interactive Walk Over Me floor. The user can choose from a list of games, upload them to the floor, play with the product, save and view scores. Some of the less important functionality, e.g. Bluetooth connection, has been mocked in this current stage, but will be implemented in future iterations.

The user tests have helped us continuously improve our mobile application design to make sure we are designing for and with our users.

References

[1] Frid, Milla. “Digital kompetens i förskolan: En intervjustudie om vad fyra förskollärare beskriver som viktigt i arbetet med digital kompetens i förskolan, 2019.” http://urn.kb.se/resolve?urn=urn:nbn:se:du-30333.

[2] Eurén, Janna. Digitalisering i förskolan : Betydelsen av IKT för barns lärande och utveckling, 2018. http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-156468.

[3] Ji, Zhe, Wei-Hsin Huang, and Xincheng Zhang. “Design and Implementation of a Game Interface Interaction on Smartphone.” Edited by Mahalingam Sundhararajan, Xiao-Zhi Gao, and Hamed Vahdat Nejad. Journal of Intelligent & Fuzzy Systems 34, no. 2 (February 27, 2018): 923–31. https://doi.org/10.3233/JIFS-169386.

[4] Åkervall, Isak, and Linus Nilsson. “IKT & utomhuslek: IKT i ett platsbaserat lärande i förskolan”, 2016. http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-145480.

[5] Muthumari, M., Nitesh Kumar Sah, Rishu Raj, and Jyotikinkar Saharia. “Arduino Based Auto Door Unlock Control System by Android Mobile through Bluetooth and Wi-Fi.” In 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 1–4, 2018. https://doi.org/10.1109/ICCIC.2018.8782297.

[6] Saborido, Rubén, Giovanni Beltrame, Foutse Khomh, Enrique Alba, and Giuliano Antoniol. “Optimizing User Experience in Choosing Android Applications.” In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), 1:438–48, 2016. https://doi.org/10.1109/SANER.2016.64.

[7] Butler, Margaret. “Android: Changing the Mobile Landscape.” IEEE Pervasive Computing 10, no. 1 (January 2011): 4–7. https://doi.org/10.1109/MPRV.2011.1.

[8] Ma, Li, Lei Gu, and Jin Wang. “Research and Development of Mobile Application for Android Platform.” International Journal of Multimedia and Ubiquitous Engineering 9, no. 4 (April 30, 2014): 187–98. https://doi.org/10.14257/ijmue.2014.9.4.20.