

Project Report

Group 1

Francisco Castel-Branco
Faculty of Sciences of the University of Lisbon
Portugal
fc53533@alunos.fc.ul.pt

Jakub Makselon
Faculty of Sciences of the University of Lisbon
Poland
fc52341@alunos.fc.ul.pt

Francisco Medeiros
Faculty of Sciences of the University of Lisbon
Portugal
fc47848@alunos.fc.ul.pt

Miguel Lourenço
Faculty of Sciences of the University of Lisbon
Portugal
fc47836@alunos.fc.ul.pt

ABSTRACT

In this paper we provide an insight into how the project came to be, how it was developed and how well it went. The project that we proposed ourselves to do was a game for the visually impaired. This idea came from the the given topics for the project in combination with our interest in developing a small game application.

KEYWORDS

Game, Accessibility, Visually Impaired, Firebase

1 INTRODUCTION

The topic of our project is *Accessibility*, with this topic in mind we devised a game app with a focus for the *visually impaired*, however the game app is also playable by a multitude of other people whether they have sensory impairments or not. The main aim of the project was to provide a good user experience by having the user play through a predetermined main story which was comprised of little bits of story that serve as a way to fit mini-games which are the main interactive aspect of the game app. At the project proposal, we were given the idea to make it so that users could also be able to create their own dynamic stories apart from the main one in order to increase replay-ability as well as give more value to the project itself. In the next sections we talk about other games that were designed with the visually impaired in mind and the implementation details of our own project.

2 RELATED WORK

We did not explore very deeply scientific research about games for the visually impaired we read some parts of the ones that were provided to us in class for the paper presentations however this could have gone better. The exploration came mostly from searching games for the visually impaired online either videos or sites that provided such games, we also watched blind people play said video games, this gave us some clues about what they expected to play (albeit not very scientific) and how they played.

3 APPROACH/CONCEPT

As stated above our project is a game app for the visually impaired, the main concept that we had to make the game usable for our target audience is obviously sound and touch (vibration). For this game to be usable for people with visual impairments the feedback

they receive from the app needs to be different from what normal apps that use, *toasts* and other visual cues. Our approach is based on sound and vibration of the phone to transmit information to the user, the sound comes from the *talkback* tool provided by the *android accessibility suite*, this tool allows the phone to "read" the text that appears on the screen and transmit the information read to the user, this is of critical importance in our approach, without this it is not possible to navigate the game if you have visual impairments. The other aspect is touch, mainly vibration, our intent is that through vibration the user knows what is happening, it gives contextual clues about how the game is progressing, specifically in mini-games.

Now, for our concept, the game is basically a succession of accessible mini-games with tidbits of story in between, our reasoning was that people that play on their phones don't do it for a very long so we wanted the story that they play through to be doable in a reasonably small amount of time whilst providing a good experience. The mini-games would provide the interactive part and the story would complement the experience, these mini-games would then be accessible without having to go through the main story, this increases replay-ability if a user wants to play a mini-game quickly and not having to go through the main story.

4 SYSTEM/IMPLEMENTATION

The entirety of the app was done in pure android, what this means is, we used no extra frameworks only java+android, initially we discussed the possibility of doing the app in unity however since none of us was familiar with the technology we decided that using pure java+android was a better idea.

4.1 Accessibility Design

we needed to be wary about the design of the app, menus, mini-games, etc., due to the fact that visually impaired people would not be able to use this app as well as a person that are not visually impaired, this meant that we needed a way to integrate the *accessibility suite* in android to facilitate navigation. AS explained below in finer detail, the menus make use of the whole screen and the entire menu is *slidable*, this means that to change menus the user doesn't require knowledge of "where" to move to the next menu, this integrates easily with the *accessibility suite* in android, the chapter creation feature however is a little bit trickier to navigate

for the blind however the intent of this feature is for somebody else to create chapters and stories.

4.2 Story

Knowing that the game was going to be made with accessibility in mind, we knew that we couldn't rely heavily on visual effects or focus on the aspects of a normal action-reaction mobile game. After some thought we decided that our game would be story-driven, maybe and RPG (Role Playing Game) with multiple paths that you could choose from and where you would control a party that would save the world or doom it depending on the player's actions, but to keep our project simple we decided to make an action-game with a fixed and single path with some witty storytelling to lighten up the mood and mini-games between some scenes so it could actually be declared a game. We discussed alternative universes, either being steam-punk, medieval or sci-fi themed where the player was a famous spy/infiltrator with a mission to carry out. We decided to stay with the "real world" universe because it was easier to imagine and due to the difficulty it is to describe every single room and object with enough precision to make the player feel immersed by an alternative universe with completely different looking aesthetics. After some iterations and discussions we settled on the story: You, a great infiltrator/spy who would need to infiltrate a corporate building and steal a seemingly normal briefcase that holds a great secret with great value to your contractor.

4.3 Scrapped ideas

As said in the above subsection we had the idea to make an RPG where the player controlled a party of friends/adventurers who could travel between locations and randomly encounter enemies in those travels. The party members could be named and would have their own unique class, having it's ups and downs. In the middle of their travels they would find the "main quest" and pushed to complete it. They would level up as any normal RPG but with the twist that the weapons and skills would improve automatically with them, simplifying the process and making it simpler for a mobile game. The combat was simple, each party member had it's own identifying sound and unique abilities (a maximum of four). In combat they could *ATTACK*, *PROTECT*, *ITEM FLEE*. The player would also have an item bag where he could store bought or won through combat items and use them as they pleased. They would have access to a shop owner where they would exchange currency for more goods that could be used. All of this would be navigable and intuitive for visually impaired users. It would take advantage of sound, vibration, swiping and tapping. The idea was scrapped almost immediately after the first group meeting due to it's levels of complexity to accomplish a minimal playable state within the established time limits. There was also another mini-game that would revolve around the user solving a labyrinth by dragging the finger on the android and receiving cues to be able to solve it, however making the labyrinth itself proved more difficult then initially expected and thus it was also scrapped.

4.4 Menu

Menu is the first activity presented to the user. It provides access to the main story, settings and leader boards. It is based on *PageViewer*,

which stores list of *Fragments* and enables smooth translation between them, by dragging the screen horizontally. Menu is divided into three levels of depth. First level gives access to chapter selection menu, settings and leader boards. Second level is chapter selection, where we can choose to play different chapters or mini-games of specific chapter. Third level is mini-games selection within the chapter. To translate between levels, instead of launching separate activities, we decided to swap *Fragments* list on the *PageViewer* and store parent level on stack.

4.5 Chapters

We split our game into chapters, which are short quest lines of our game. Each chapter has a list of scenes. Scene consist of dialog presented to the player, background image and possibly mini-game that will be launched when player proceeds to the next scene. We created *ChapterActivity*, that loads scenes from the database, and presents them to the user. We store all the chapters and their scenes in the Roam database.

4.6 Mini-Games

Mini-games are short activities/puzzles, that player is to encounter during chapter. Mini-game can be any kind of *Activity* class. The only restriction is, that it should implement difficulty scalability and return the score. When player reaches the point of the story, where mini game appears, mini-game is invoked as *startActivityForResult* and the score is collected afterwards. Mini-games can be accessed directly from the menu during chapter selection. Since the app targets the visually impaired most of the mini-games make use of the sensors provided by android (with the exception of *LockPicking* which uses sound and vibration), the sensors were very simple to add to the app which made development faster. The sensors besides being easy to use and provided quick feedback also proved to be very enjoyable in a player kind of way, they were simple and provided a good user experience this made the mini-games interesting albeit simple according to some users. Below we will explain each mini-game individually.

4.6.1 Escape Mini-game. This game is very simple, it is based on *shake*, we use acceleration sensors to determine the *shake* degree, this degree is used to determine not only how many points the player gains by not crossing the lower *shake* bound but also the winning condition, if we shake (above the lower bound) the android for a predetermined amount of time (difficulty might change this) we win, pretty simple, but adds good interaction and is a good change of pace from simply tilting the phone. This game would take the time taken to complete calculate the score.

4.6.2 Dodge Lasers Mini-game. This game is a little bit more complicated on the implementation part, because we require two sensors to determine the tilt direction of the android, the accelerometer and the magnetometer. The main objective of this game is to follow instructions that appear on the screen, left, up, right, down, and tilt the android in the desired way, there are ten instructions in the sequence that is required to be completed without making mistakes (changes with difficulty) otherwise you go back to the beginning of the sequence. This game is simple in concept however the implementation was a little bit trickier. To increase difficulty

we could decrease the time between giving command and check the tilt levels. This game would take the difficulty level to calculate the score.

4.6.3 Lockpicking Mini-game. This game starts with a random n -value pattern (to be defined by the difficulty chosen) where the player needs to discover that pattern through trial and error, having with every mistake go back to the starting position, with the idea that he broke the lockpick he was using. The only possible movement the player can do is to swipe left or right. This mini-game uses sound to identify the correct and incorrect swipe movement. This game would take the time taken to complete and the difficulty level to calculate the score.

4.6.4 FindSpot Mini-game. This game can be reduced to the idea of a black screen with a randomly hidden button in it. The player needs to search for it using it's touch. If the player finds the button the device vibrates indicating that the button was found. To increase the difficulty we decided that the button size could be changed. This game would take the time taken to complete and the difficulty level to calculate the score.

4.7 Chapter Creator

Since our story has well defined structure, we decided to create chapter creator. Players can create their own chapters, adding new scenes, which is creating dialogues, choosing background and picking mini games, that they encounter during regular play-through. After custom chapter is created, it is stored in the database and it is accessible from the chapter selection menu, same as main story chapters. Users can edit or delete chapters they have previously created.

4.8 Database

In order to run the game smoothly with all of our features working, we implemented two different databases. Even if the database implementation was for the last checkpoint, we've developed a score system early on which is stored remotely on a Firebase Realtime Database as a result of the mini-games completion on our chapters. Then, following the teacher's suggestion, we've developed a chapter creation system (described in section 4.7) and it would be stored locally using the Room Database Framework.

4.8.1 Firebase. In the remote database implementation, we simply store the email, display name (which are obtained through the Google authentication mechanism), an array of emails (the follower's list) and the player's score. As this database is Document Oriented, the key for each user's object is it's email's MD5 signature. We use the email's on that array in order to compare with all users that we get from the database and filtering them for the Follower's leaderboard.

4.8.2 Room Database. For the custom chapters, we've decided on the Room database because we needed to get our objects synchronously, as the previous implementation of the Firebase proved to be too difficult to work with. With this approach, we could also give the players the possibility of playing their own chapters without an internet connection.

We've used two database tables: one for the Chapter's title and description and the other for its dialogues (its content and background information) and mini-games (which mini-game should we start at a given time). As the database stored the entries in its correct order (iterating through a list) and as we query it, it returns the ordered objects, we didn't need to include a special variable to keep track of the order.

5 FUTURE WORK

We could definitely do more with this idea, there are a lot of ideas that were scrapped due to timing concerns, however there were some suggestions in the open day by some users that we could do. If we had more time to develop our game, or simply focus on it in the next months we could improve the following:

- Read more research on the subject of visually impaired technology interaction
- Make a more intricate story
- Add more mini-games
- Save the created chapters into *Firebase*
- Improve mini-game design by adding more to do.
- Create different approaches for how mini-games work.
- Improve layout and image quality
- Add audio files narrating the story

6 EVALUATION

6.1 What went well

Since the beginning our group was very united and full of ideas on how to make the game. It was dynamic and had little prior programming skills for Android, making the process of learning easier to those who didn't have it. Although we had an ERASMUS colleague, communicating with him was easy and we all had a general idea what we wanted, making the game design process less conflicting.

On the open day we got some positive reviews as well as negative reviews, as expected, focusing on the good parts for now, people enjoyed the app and thought the idea was interesting, the mini-games were fun and the story was brief but good. Having the story smaller brought positive results, users were able to finish the app as a proof of concept. We also got some good remarks on the chapter creation feature which worked well enough to allow users to create and play their own stories.

6.2 What could have gone better

Since the beginning we expected to make a small game that would take around 15 minutes to complete. Little did we know that making accessible mini-games would be so hard, so we only made four of them to accompany the story. Also we had a colleague that had an android phone who had problems installing the APK, making it hard for him to create new stuff for the application; the solution was for him to focus on writing the main-story, making the images, marketing and Q&A. Some negative remarks from the open day were as stated above that the *accessibility suite* was difficult to use for people that were not used to it, this is because the *accessibility suite* changed the way gestures and clicking worked on android, this caused a lot of confusion (we somewhat expected it) simply

due to the fact that the gestures and motions users were used to in their phones was simply not present, some other remarks said that the mini-games were not very "smarts" and were too simple, however designing such games would prove to be very difficult and time consuming (however it is an interesting point to look into in future work on the app).

6.3 Wrap-Up

All in all we believe that the app we created served its purpose, the project progressed smoothly (despite some difficulties), and we think that it was, overall, a nice experience for the users and for us.