Gamification of Evacuation Drills Post Mortem

Our group was requested by the client to make an augmented reality mobile app which could track the user as they move throughout a building and relay that back to a server. The gamification part means that the player has to avoid obstacles etc. and eventually find their way to the exit. Our project owner was Mirjalol Arminov, who would be responsible for any further distribution or development of the project.

The goal was to make several working prototypes before the final one on 2022-12-12, and demonstrate our progress in front of the class. In the beginning our top priority was to get it set-up, and have some sort of an AR development environment working. When we got that working our focus shifted to the core features like the tracking system, database and obstacles. While implementing we constantly did bug testing and worked on improving the overall user experience for the end-user. In the end we ended up with a product which had some missing pieces, but which contained core functionality and we were happy with it.

The project was developed using Unity which supports both Android and IOS deployment, but during development we focused on Android as the majority of us had Android phones and it was easy to debug that way. Should the project owner decide to publish the project, the target audience would be whoever has a phone, is in one of the supported locations, and is willing to participate in the research.

What went right?

1. **The AR basics**

From the beginning, we were intimidated by the AR aspect of the project, as none of us had worked with AR before. It ended up being easy with the AR packages available for Unity. The basics of tracking, plane detection and overlapping objects was very straightforward.

2. **The aesthetics**

The project was looking a little bland from the start but we ended up with an easily changeable aesthetic which fit the core concepts of the game. The graphics have that kind of ugly-pretty charm which makes you feel nostalgic. The particle system also turned out nice and added a polished feel to the game.

3. **The tracking system**

The accuracy of the tracking was something we struggled with for a time, but we ended up with a fairly robust system which can capture very subtle movements. This is provided you have a phone powerful enough to utilize AR to its fullest, but most modern phones handle it no problem.

4. **The AGILE method**

The fact that this project was developed using an AGILE method allowed us to quickly iterate solutions and continually test/debug features as they were being implemented. A more linear waterfall model would have stretched out the project to take an unreasonable amount of time to complete.

What went wrong?

1. **Team cohesiveness and workload spread**

Within our team, the workload ended up being unbalanced in such a way that two team members did most of the work. The cohesiveness of the team could have been improved with better communication between members, and a better organizational structure within the weekly meetings.

2. **The project owner & feedback**

Our project owner shared with us the initial ideas about the project, but beyond that we had extremely little communication with him - even inside the weekly meetings. Our lecturer answered questions instead, so we treated him as the actual project owner, since we couldn't get in touch with the original project owner.

The feedback we received was at times conflicting. For example, during one meeting we discussed the fact that the app needed the player to start at a predetermined position within the building. The lecturer expressed that this needs to be dynamic and that we should let the player choose where they start in the building. One of our team members suggested adding a map on which the player would choose where they currently were in the building and the lecturer agreed. After implementing this, he went back on the idea and said that this wasn't what he expected and that the phone should just know from the start where the player is.

3. **Campus 3D model**

From the start, we were told we would get access to a 3D model of the campus. However, such a thing did not at all exist according to the faculty. It wasn't too much trouble to model it ourselves using the fire escape plans as a template, but it took some time from the project which could have been used elsewhere. We might also have been able to improve the tracking if supplied with a more detailed model of the campus, as well as support more than the first floor of the first building.

4. **Relying on the GPS**

At the start of the project, we had agreed to use GPS as the tracking method. However, this was not accurate enough to get a good fix on the player when indoors. We changed this to a purely "relative to the building"-approach, where the player chooses where they are in the building to initialize the tracking and then the AR positioning system takes care of the rest.

5. **Advanced AR**

By the time the client requested automatic starting position detention, we had run out of time to implement such a feature.

6. **The database**

The database didn't get fully completed, but given the time and resources we had available it is not unreasonable. It is easy to expand upon for future developers who may want to continue the project.

What can we learn from the experience?

This project helped us learn more about mobile development, as we had limited previous experience with that. AR was intimidating at first but we gained valuable experience within that field as well. AGILE development practices were not utilized to their full capacity within this project, and that could be improved upon for the next time - but its core pillars were helpful to keep in mind.

The experience of working for a client who was absent or unclear about their goals, while perhaps realistic, was very demotivating. This experience has perhaps hardened us for the work life experience. Working on a bigger project was fun, as we could see the progress more clearly. Setting up the foundations of the web server was very new, but was rewarding and resulted in valuable developer experience.

In the future, we could focus more on getting the amount of planning correct. Too much and we waste time, too little and it results in bad team cohesion and inefficient work spread.

Statistics

- Number of developers assigned to this project: 4
- Duration: from 2022/11/21 until 2022/12/12 (21 days)

3rd party tools and products used:

- Git
- GitHub
- Unity
- Discord
- Google docs
- Google drive
- Trello
- AR Foundation, AR Kit, AR Core
- Gimp
- Online conversion tools
- Docker
- PostgreSQL
- Django
- Gunicorn
- Nginx

Meetings:

- Around 5 or so.
- 110 - 130 hours total work