# Proposal

The game will be a microgame collection for mobile that utilises multiple sensors for varied and unique gameplay opportunities. Players will use touch, motion, microphone and proximity input to play through a large number of very short games, usually taking around five seconds each. The player has a certain number of lives which are lost upon failing games, when they’re all one it’s game over.

Ideally this project will act as a cumulation of all the techniques I have learnt throughout the course. It will act mostly as a portfolio piece so that I have more diverse projects to show off.

The target number of games is 80. They can be played in three modes: endless, which cycles through all games of a set difficulty until the player either quits or runs out of lives; challenge, which features increasing difficulties; and practice mode, where a specified game can be played as much as the player wants. Playing the game will earn a currency which can be used to purchase games and difficulties for practice mode as well as some items to change up the gameplay.

The game will be made in Unity, using GitHub for source control and Trello for project management. It will be released for free on Android on the Google Play Store as well as Itch.io. This release version will be of a high level of polish with no known bugs.

Research would include getting the various sensors to work, especially the microphone and proximity sensor as I have had no prior experience with those. The rest of the project will be utilising the game development skills I have learnt throughout the course to create a wide range of games spanning many gameplay styles.

# Progress so far

## Research

As I already have a lot of experience in developing with Unity, I can easily come up with solutions myself so there isn’t a necessary need for very much additional research. However, I will have to research into the development and input techniques that I will be using throughout the project. I have looked into the various topics I need, but no formal research has been conducted so far. The input techniques I will have to learn more about are various forms of motion detection, microphone input and making use of the proximity sensor.

I will be looking at WarioWare and Mario Party minigames to see what similar games have done and use them as inspiration to create some of my own. I will take note of the various methods that they use to create a framework of rules that I will follow to develop my games.

## Prototype

Most of my time on the project so far has been spent on the prototype. At the time writing, I have designed 78 games, finished programming 13 games, using 8 unique input methods. Progress on creating these games has admittedly been slow, but I am developing them in such a way that I can reuse assets and scripts in order to eventually be able to drag-and-drop features to make games, so the development time for each game will decrease throughout this project.

# Project specification

Final project will include:

* Minimum of 50 microgames (aim for 80)
* Varied use of four different mobile sensors
  + Touch (tap, multitap, hold, swipe, drag)
  + Motion (accelerometer, gyroscope, orientation)
  + Exotic sensors (Microphone, proximity)
* Different gameplay modes
  + Practice mode
  + Challenge mode
  + Endless mode
* In-game store (using in-game currency)
* Settings menu
  + Allowed orientations
  + Allowed input methods
  + Accessibility features
  + Clear data
  + Language settings
* Released on Google Play Store

# Potential solutions

Problem: A game does not end up being fun  
Solutions: Remove the game, change the game, leave it.

I believe that the best option here would be to make enough small tweaks to make the minigame be as fun as it can be. Failing that, I would leave the game as it is because I wouldn’t want to remove content in a project based on making a lot of content efficiently.

# Tools and techniques

|  |  |
| --- | --- |
| Tool (choice in bold) | Use |
| **Unity**, Unreal | Game engine |
| **Visual Studio** | IDE |
| **Photoshop**, Krita | 2D art assets |
| **3DS Max**, Blender | 3D art assets |
| **Github**, GitKraken, GitLab | Source control |
| **Trello**, Hacknplan | Project management |

I have decided on all of these pieces of software for the same reason: I have years of experience with each of them which will allow me to produce work as efficiently as I am able, which is very important for large-scale projects such as this one. I have also previously looked into alternatives numerous times, none of which provide enough benefits to justify learning new software. In short, I know the chosen tools significantly more than others and the years of experience have taught me that I work very well with them, which maximises the efficiency in this project.

# Methodology

The project briefly started back in May where a prototype of the game loading system was created. I had planned on working on the project throughout the summer, but it was difficult to motivate myself for that.

I will be working on as many games as I can manage without burning myself out, aiming for at least one game per day. I don’t have any specific plans for which order games will be implemented as I will be creating whichever game I have the most motivation to work on, to ensure that I work as efficiently as possible.

Fortunately, the project plan is very open, so I can consider each game before implementing it and perform tweaks whenever necessary. Progress will be re-evaluated every week during meetings with the supervisor, where we discuss how the project should continue to move forward and readjust goals and features to be more realistic and viable.

# Management

I am using Trello to keep track of the project, where I have a list of all games which have checkboxes to measure their progress. To do this, I am tracking whether each game is functionally complete, complete art assets, complete sound assets, is polished with clear win or loss events and supports the game’s multiple difficulty levels. These conditions will be tested before being marked as complete.

My primary metric is how many games with mostly unique systems have been implemented to a high degree of polish. I have decided on this because my goal is to showcase a wide variety of different gameplay styles and input methods, so while simply having a large number of games is impressive on its own, the game would be much more impressive and enjoyable if each of those games is unique in its own way. Tangentially related, my second most important metric is the entertainment value as I would like the game to be enjoyed by others, so if I or testers don’t find something fun, it should be changed. I have no plans or interest in monetisation, so the game’s market appeal is not something that I will be considering.

# Resources

The project does not require any equipment that is not already available to me or additional resources beyond people who would be testing the game. To have other people involved with the project, I would need to go through the ethics board for permission.