Implementation 1

Group 2 - Vikingz

Damian Boruch

Tommy Burholt

Elliott Bryce

James Butterfield

Ren Herring

Zhenggao Zhang

Sharlotte Koren

Implementation

Libraries Used

To create our game we have decided to use libGDX. LibGDX is a cross-platform java game development framework that provided us with the necessary tools to get a simple window up and running as well as a plethora of other tools which made the development much simpler.

For our build system we went with gradle, however this was essentially imposed on us from the moment we decided to use libGDX as the project creator creates the project with gradle. Using gradle made building cleaning and packaging our project much easier than it would have been if we weren't using a build system.

LibGDX is under the Apache 2.0 licence which means that we can use the software for any purpose such as distributing and modifying the software, which also means it's perfect for this project.
https://github.com/libgdx/libgdx

Another library we used is GSON which is a Java serialization/deserialization library that converts Java Objects into JSON strings made by google with the Apache 2.0 licence.This is used for the purpose of loading assets and data into string dictionaries.
https://github.com/google/gson

Assets

All of the assets for our game have been created by our team members, everything apart from the libGDX glassy skin which was taken from https://github.com/czyzby/gdx-skins. Since all of the assets are custom we don't need to worry about any copyright laws that assets downloaded from the internet may have been under. It also gave us more creative freedom to design the assets ourselves as we could design that would better suit our game, as opposed to having to tailor the game to the assets.

Things left todo or that could have gone better

We have managed to implement all of the requirements set out in the product brief, as well as a large amount of those outlined in the client meeting. However, some of the features such as resizing the UI elements could have been made a little better.

We also planned on creating a leaderboard (UR_LEADERBOARD) which we outlined in our requirements document, however due to time constraints we were unable to implement it in our final iteration.

Also, currently the game config data is being saved to a binary file, and while it's fast and the file size is small, it would have been much more useful if it were saved to a JSON file.