

Team 8

Levi Clark, Sam Engert, Shafeen Housein, Blaine Edmondson, Weiting Wei

Mobile Unity Game (unnamed for now)

Project Synopsis:

A mobile game created using Unity. Its primary purpose is entertainment.

Project Description:

The project is being undertaken to develop skills involving mobile and game development. This project provides the opportunity to provide entertainment for the user while generating revenue through infrequent ads. The end result will be a project published to the Google Play Store, and the Apple App Store.

Project Milestones:

Semester 1

- Game Design – Nov. 1
- UI flow design – Nov. 15
- Unity/Blender Research and Training – Dec 15

Semester 2

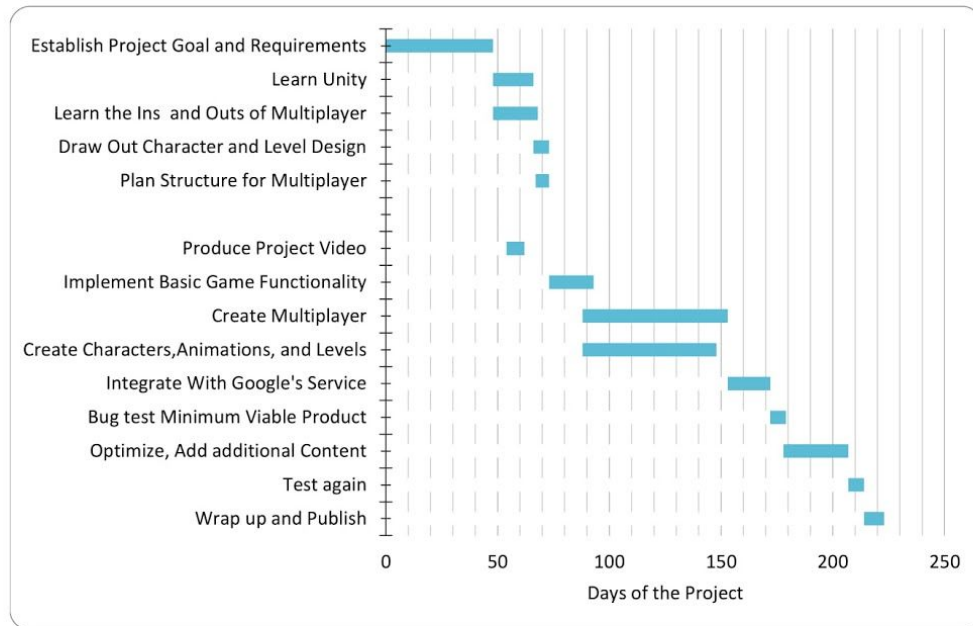
- Implement core game loop (ex. App opens, starts, and finishes) – March 1
- UI Implementation – April 1
- Game Asset Creation – March 1
- Implement Play-state mechanics – April 1
- Implement outside API's (accounts, scores, etc) – April 15

Project Budget:

- Apple App Store Account -\$ 99/yr
- Unity/Blender training for team members

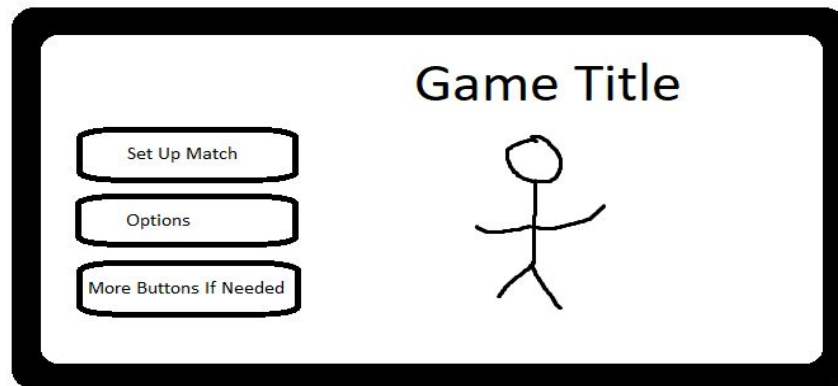
Work Plan:

- Asset Creation – Levi, Sam
- Game Design – Entire team
- Core Game Loop – Sam
- UI Implementation – Weiting, Blaine
- Play game mechanics – Entire team
- Implement outside API's – Shafeen, Sam, Levi
- Publishing/Building out to mobile – Entire team



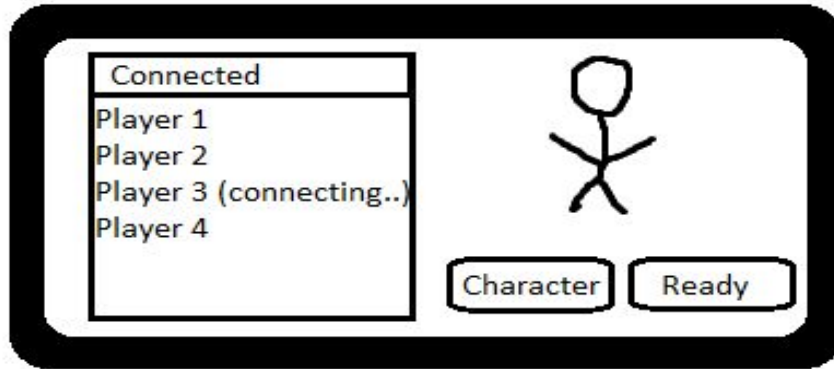
Preliminary Project Design

Our primary focus will be to run the client version of the game on android devices. Each client will connect to an authoritative server which will synchronize all connected clients to the server's current state. Upon running the client application, users will be shown the main menu screen.

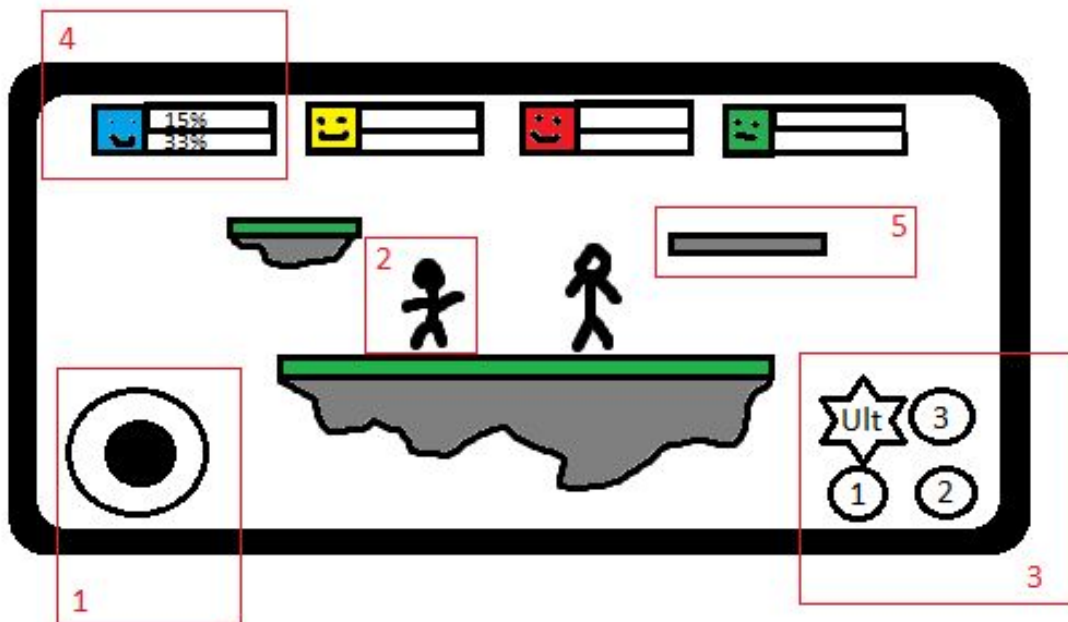


On this screen they are not yet connected to a server. Selecting the "Set Up Match" button will take the user to the next screen where connections are initialized to set up a

match with up to four players. The "Options" button will handle options such as rendering quality and sound options, as well as account options such as username.



The "match-making" menu will show available lobbies and initialize connections for each client connecting to the server. Each player's username will be displayed along with their connection status. Here each user will be able to select the character they want to play for the current match before hitting the ready button. Once all players hit the ready button, they will each be taken to the game scene.



1. GUI touch-based joystick, used to control the user's character (2)

- a. Moving the joystick up allows the character to jump, some platforms (5) can be jumped through if it is thin enough.
- b. Holding the joystick left or right makes the character (2) run in the corresponding direction.
- c. Flicking the joystick down will make the character (2) fall through a platform (5) if it is thin enough. Holding the joystick down will keep the character (2) in a crouching state until released. *Possible crouch walking

2. The player character. This is the user's avatar that will be given actions and be used to fight other avatar's.

- a. We plan to allot enough time to design 4-5 unique characters to start with.
- b. Each character will have 3 common abilities listed as 1, 2, and 3 above.
- c. Each character will have 1 ultimate ability and 1 passive ability. The ultimate ability can be used once by spending 100% energy displayed in the Player Unit Frame (4). This energy is built up by successfully landing common abilities.
- d. A player character will lose a life when it has been knocked out of the map.

3. Ability/attack GUI buttons

- a. The 3 smaller buttons represent the more commonly used abilities for the character. Some of these may enhance movement, have a cooldown, or even a negative trade-off depending on the ability.
- b. An "ultimate" ability will become available for use after the avatar has successfully landed enough common abilities against another player. Progress towards unlocking an ultimate could be displayed as a percentage in the Player Frame (4), or alternatively, as an opacity fill on the "ultimate" button itself.

4. Player unit frame, a unique one is displayed for each player in the match.

- a. The face of the character the player is currently playing will be displayed where the smile is in the frame.
- b. The percentage bar in the top of the frame represents the damage that player has taken. This percentage is multiplied by the forces of other player's attacks. It is reset to 0% when the player has died.
- c. The bottom percentage meter is the energy built up towards being able to use an "ultimate" ability. Once at 100%, the player is allowed to empty the meter to use an "ultimate" ability.

5. Map platforms

- a. Some will be thin enough for a character (2) to pass through by either jumping or crouching.
- b. The main goal is to knock other players off of these platforms, while staying on them yourself.

We plan on building the bulk of the client and server software with C# using UnityEngine. All game assets will be created using Blender, GIMP, and Audacity. For task management and version control we will be using Jira and GitHub. We will need a database for storing user information such as usernames and will be deciding on either MongoDB or MySQL. We will also

need to use a multiplayer framework in Unity (Photon/Forge) unless we decide to implement our own.

Ethical Issues

- Since we may decide to publish this project to the app store, it will be closed source. If we decide to release it, we will publish it on github under MIT's open source license.
- We may request users' personal information like email addresses and password in order to identify users, and will be responsible for keeping that information safe.
- We may use ranking system. If so, we need to publish users' user names, but users can choose to keep their username privately.
- This game will have some mild violence and thus the appropriate content warning will be displayed on the store page.

Intellectual Issues

- The game engine we are using is Unity. Therefore we must adhere to all of Unity's Terms of Service here : <https://unity3d.com/legal/terms-of-service>. In summary we must create our own assets if we intend on selling the game unless if otherwise stated it is for free use. We must also pay Unity a certain amount of money if our game produces enough profit and for any services that we use from Unity (e.g multiplayer cloud services).
- We will also be utilizing android and if we do release this to the app store are subject to the Terms of service that google has available here: https://play.google.com/about/developer-content-policy/#!?modal_active=none. In summary we will not impersonate any other developers, copy content from any other developer, and not attempt to use our app for any malicious purpose. We agree to pay any and all fees that may be incurred such as paying for a developer's license on the app store.
- Github will be used for Version Control. If we decide to move on with the project we will then switch our github account to a paid version for developers.
- We will be using Blender to generate assets and developing our models. Blender is free to use and published under GNU. And thus we will adhere to its rules. Blender's license is here: <https://www.blender.org/about/license/> and it gives developers the freedom to modify its code, sell whatever they create, and sell their modifications as their own product.

Change Log