**Project G**

Kale A. Wicks

Southern Methodist University

**Introduction**

*Project G* is the tentative name of a 2.5d fighting game coded in C# and made completely within Unity. The game is meant to be an installation that others can play during the time of the virtual showcase. *Project G* is important for two key reasons. The first is the societal value it produces as a game during the era of covid, allowing others to relax and have a temporary escape. The second is as a portfolio building project. *Project G* is a piece I intend to carry into the future, building on it further and further throughout my career to be a project on par with something that could be pitched in the business sector of the gaming industry. *Project G* will allow me to learn and problem solve for creative projects using Unity and C#. The tangible outcomes of this semester project, will be a functional framework as the first steps towards making a polished fighting game.

**Context and Related Work**

Fighting games could be a classical and established genre within the video game community. Examples of “classic” fighting game titles are games like *Street Fighter*, *Tekken*, or the *Marvel vs Capcom* franchise. Over time as technology has evolved, these games have advanced and been polished by upgrading their visuals and playability, resulting in the birth of new and more unique fighting games such as *Fantasy Strike* (Fantasy Strike, 2019). *Fantasy Strike* is a standard fighting game on the surface, two characters fight until one of their health bars reaches zero, however the systems in which it does that are what make the game unique. *Fantasy Strike* implements a point-based health bar system, breaking the standard bar at the top of the screen into segments based on each character - bulkier characters having more segments, and characters with much higher-than-average damage having less. This balances the game while also giving each character a unique feel, something I seem to emulate in *Project G*. On the technical side of things speaking to how the game is coded I would like to emulate a few people. As I have never made my own fighting game, I have done research on how different people have sought to make their own. One such example is Jabrils who made a fighting game based around the idea of moving in response to your opponent like a game of chess (Jabrils, 2020). Jabrils also coded his game in Unity, allowing for me to have a tangible and understandable workflow that I could adapt to my game’s needs. Another person I seek to emulate is Steven, a YouTuber who documents his journey making games and learning the different ways to code and create them. As I experimented with the earlier concepts of *Project G’s* movement, I found that Steven used Unity to understand and learn 3D game development (Steven, 2020). Adapting the basis of Steven’s 3D first person movement, as well as overcoming the nuances of C# and Unity became the core for *Project G*’s movement. However, due to Steven’s focus on creating his own game with a style different from mine, I would need to look elsewhere for more detailed guidance. This search would eventually lead me to the YouTubers Sorcerer (Sorcerer, 2020) and Brackeys (Brackeys, 2019). Both channels provide in-depth coding tutorials for movement and character animation in Unity. With all of these resources, I sought to teach myself as much as I could in order to create a functioning framework for what could be a career making project for myself.

The criteria I have determined to find success in this work for the semester are as follows:

**25% -** Character movement. (Player 1 can move left and right across the screen. They should also collide and be unable to move through Player 2’s model. Both model’s should not be out of shot and should stop advancing once they reach the ends of the screen. The walking animation can still play, but they won’t go any further if they’re on the edges of the screen.)

**25% -** Health Bars. (The game should have a working health bar system that updates with each hit on it’s respective character.)

**20% -** Character actions. (The player should be able to perform an action of punching or kicking)

**10% -** The game should end upon one or both player’s health bars reaching 0.

**10% -** The health bars should be points based like the *Fantasy Strike* system.

**5% -** Code should be clear and concise and work regardless of which way the player is facing.

**5% -** Players should be able to pause the game.

**Documentation**

In order to document the separate parts within *Project G*, at the end of its completion I will review and analyze the code’s strengths and weakness. I will distribute this to others who may want to look over it or have a better understanding on the game’s framework by creating a document with corresponding sections to each scripts. As this will be on the more technical side of things, I will not have it in the initial gather.town installation, however I will provide it within the github download files. In order to display *Project G* in gather.town I will screenshare a demo of the game explaining how it works and how to play it. Players will not need any special controllers for the game as I will base its initial movement commands for keyboard in Unity.

**Works Cited**

Fantasy Strike. (2019, June 27). *Fantasy Strike — Gameplay Trailer* [Video]. YouTube. <https://www.youtube.com/watch?v=0MCka4Z_fNQ>

Jabrils. (2020, December 29). *Fighting Games are boring, so I made one that makes you feel like a god* [Video]. YouTube. <https://www.youtube.com/watch?v=hpF2CuksR64>

Brackeys. (2019, October 27). *FIRST PERSON MOVEMENT in Unity - FPS Controller*

[Video]. Youtube. <https://www.youtube.com/watch?v=_QajrabyTJc>

Steven. (2020, May 23). *Attempting to Learn 3D Game Development in One Week* [Video]. YouTube. <https://www.youtube.com/watch?v=1mwNrMD0Lm0>

Technologies, U. (n.d.). CharacterController.Move. Retrieved February 07, 2021, from <https://docs.unity3d.com/ScriptReference/CharacterController.Move.html>

Sorcerer. (2020, July 18). *How to move objects using WASD in Unity 2020* [Video]. YouTube. <https://www.youtube.com/watch?v=KiVL5rfVdTw>