Friday the 13th Coding Project Final Report Summary
by Alberto Ramirez, Aqsa Arif, Lucy Harck

Friday the 13th is a first person 3D horror survival game created to test a player's decision making skills in stressful situations. The game combines a story narrative and puzzle solving in order to complete game objectives. Gameplay implements the traveling salesman problem where players are forced to make decisions that will produce the most efficient outcome, in this case survival. Puzzles in the game are solved through use of items. The player will find items throughout the game of which they can only carry a limited amount and so they will need to be wise about choosing which items they will carry. The danger aspect of the game is a monster that will threaten the player's life at every turn.

The game story revolves around a police officer named Jason who will be working on solving a missing person's case. The officer is initially lured into a dark forest and unable to leave. He will need to find clues in order to solve the mystery plaguing the forest all the while evading the monster that lurks at his heels. Jason can pick up items he finds lying about, but can only carry a limited amount. He has to make decisions on which items to carry and which to drop along the way, keeping in mind that only specific items will help him solve specific puzzles. Solving these puzzles will enable Jason to navigate the maze-like forest. Although Jason can hinder the monster, he cannot cannot kill it and so he has to be efficient navigating through the forest or risk the possibility of death.

Friday the 13th is intended for PC gamers who are 17 years or older and have a working PC that can handle 3D game graphics. The game is targeted towards an audience who prefer the horror/survival genre and aims to provide entertainment while also challenging the decision making skills of the players. The players can only emerge victorious if they manage to outsmart the monster and make it to the end, along the way the players will come to know why the monster haunts the forest and what actually happened to the missing person.

For the first release implementation of the coding project, a first person controller was set up with camera view. We gave the player the ability to pick up items with a key and update the knapsack UI counter. Additionally, the game's monster was set up and was given a predetermined path to walk on. The player damage was based on proximity that would reduce the player's health displayed on the health UI bar when nearing the monster.

The second release included two new maps, the tutorial and first level map. Introduction of scene transition was set up as level transitions and upgrades were made to the inventory UI. The inventory system was given the equip and drop button features. The second release also included a main menu scene. The main menu was given new game/load game/exit game buttons.

For the final release, improvement was made on the inventory UI. The player was able to pick up items and see it reflected in their knapsack UI. Players were able to see the item in the inventory UI with its name, sprite, and equip/drop functions. Every item was set up as a singleton using a scriptable object class that had its own set of attributes: name, weight, sprite, and gameobject. The knapsack class kept track of these items through a list of item objects. As for improvements made on the maps, the main map was given more items, a maze layout was introduced for the forest, and a dead body was added to act as the missing person. This was followed by the tutorial stage that was given more assets and given a police station layout with a kitchen, office, and a general area. Improvements on user interface was to give players a working model of choosing a new game, save game with slots, load game with saved slots, and an exit game option.

As a group when we worked on the 3D game, common problems we came across were lag and low frame rate due to our assets being too graphically demanding. We did not test the game on different platforms besides what was available to us, so we are still unaware of possible compatibility issues that may arise. Additionally, as a group we found using version control more efficient than using git hub mainly due to game size restrictions. The only issues we had with version control was the initial unity version compatibility and data transmission exceeding a few gigabytes, these transmissions would take upto an hour.

Game testing involved component testing of game items, monster path, player health, map boundaries, and boundary testing. There wasn't any specific testing needed for quality control or acceptance range. Hence, testing mainly focused on reproducing the same results with an accuracy up to 95% of the time. Game items were tested for position on the map, valid player pickup, valid sprite display in the knapsack alongside weight display, item drop, item no longer being visible in knapsack, and item position upon drop based on player position. Player health was mainly tested by having the player model die over and over again by approaching the monster. Map restriction mainly focused on making sure the player model doesn't fall through the floor, pass through walls, or fall off the map. Lastly, boundary testing was done by making sure game functions did not exceed set max values.

Finally, features that could have been improved on in our final model of the game. The game needed a working script for item merging/crafting. Additionally, it needed game prompts to guide the player based on the storyline and dialog trees for decision based dialogues. Some functionality features that could have been improved on were visuals by giving a base mesh for the player model as a human in a police officer outfit with facial expressions and movement mechanics. The introduction of NPCs to add more to the storyline narrative and adding audio cues triggered by item proximity or when the monster is nearing. Additional improvements on functionality would be adding the item inspect feature to look close up at inventory items for puzzle clues. Ideally, a final release of the game would have a working tutorial with game prompts on how to play the game, upon completing a level a new map would be generated with a random maze and predefined puzzle with items scattered at random. When the game is completed, there would be a credits scene with a narrative in complement to the storyline.

Overall, working with a gaming engine came with its fair share of problems. Development issues ranged from basic script errors to outdated APIs not working on newer versions. Asset materials had display and rendering problems as well that took a generous amount of time to resolve. However, through discussion and referring to online tutorials we were able to problem solve effectively as a team and kept in regular communication about development progress. Given our small amount of experience with Unity, we quickly learned how to use the tool and took a creative approach to the project, making it our own since it was a different approach to Group 15's Fall 2019 Report in terms of environment and style.