**Project Description**

Zombie dash is a map-based survival game which can be played by strategical decisions. The maps are based on different puzzles which are used as mazes though which the players need to find his way out. There are multiple mazes which can be selected from the main menu. The player is supposed to find their way out within the time limit and without having the player’s health depleted to zero. Player will encounter zombies along the way with random abilities and weapon speeds. Killing these zombies will have the chance of randomly dropping a consumable to be used by the player. This project is directly focused on the gaming industry. There are many zombie games on the market, however there are no zombie games that have maze/puzzle elements to it.

**Releases throughout the semester:**

In the first release, we mostly worked on user interface which included the login menu and the options menu through which gameplay can be customized. We worked on the shop, which included different options for the player to select such as purchasing energy or weapons. We also worked on the ads which can be used to refill the energy. We also worked on the shop which had different option like clothes, guns, and ammos, but the functionality was not yet implemented in this release.

In the second release, we focused on gameplay which included start up of the game and the player was also able to play the game with the ability to use ammos. The health bar and the weapon info on the right side was also implemented in this release. The player was also able to move in any direction using the arrow keys. Also, the zombies were implemented which were basically AI controlling the moment of the zombies. Zombies would follow the player when the distance between them decreased. Also, the energy system was implemented by adding ads which would increase the energy and there were total of 11 ads which were randomized. Even the maps, finish line and timer were improved as time passed.

When compared to the original idea of this project, our project is very similar in areas like the gameplay, noise and the authentication. We also implemented the mazes just like the previous group wanted to implement. The things that we were not able to implement was real time-based maps as the previous group did not make it clear as how to implement this functionality. We developed maps using application called tiled. The things that we added to this project were in-game shop where the player can select multiple options. We were also not able to implement multiplayer withing the limit as our focus on making single player working.

**Testing different aspects of the game:**

In this section, we tested most of the functions and, also mentioned different aspects of how will they be tested as well as the test procedures of the particular test and whether or not they passed. We focused our testing on the following elements: Maps, Movement of player, energy depletion, damage, ads, ammos, guns, clothes, player death, weapon info, health bar, game time, reaching finish line, health packs, zombie damage, player and zombie shooting, Zombie AI, walls and in-game shop.

**Results for the tests:**

Different individuals of the team were tasked with the test of different aspects (mentioned in the above paragraph) of the game. Each member tested and mentioned that whether or the tests passed or not. Each of the members were suppose to answer the following categories: dates of the execution, member name, expected result, actual result and the test result.

**Inspection of all aspects of the game:**

For the inspection, the code was divided into subsections for each member of the team to inspect. Using a generic checklist found online we created our own checklist based on it, and used that as our guiding point for coding inspections. Using this checklist we inspected the overall structure of the code like whether code is well structured or not. We also made sure there is no repeated code which can cause issues. We also checked whether code is properly commented so that it is easy to understand the code.

**Issues with the project and its solutions:**

One issue was the VCL media player’s requirement in the PC of the user as it was necessary for the user to play the ads. We tried different software because it’s not possible for everyone to download different software for the video. Other options we explored involved other libraries such as Pyglet, and Pygame built in video viewer. Unfortunately, Pyglet would close out the application when viewing an ad, and Pygame’s video feature has been deprecated due to instability. The other issue was the real-time based maps and the only solution we were able to come up with was to obtain information about a current location, and somehow create a fully playable map complete with starting and ending locations, as well as zombie spawns. If creating a map is possible from some application, then next step would be to developing a process to map data with the application.