Updates:

Throughout the game development process we updated the idea of our game with the inclusion of many of the UI elements, as well as the ammo system. While playtesting the game after its initial construction we realized that there was a lack of information being given to the player while playing, such as what wave they were on and how much ammo was left before they needed a reload. So we decided to create the start screen, so that the player is aware of their goal and they aren’t immediately thrown into the game, the in game UI includes a wave counter and ammo counter so that the player is aware of how far they’ve made it and how many bullets they have remaining before they need a reload. The ammo system was added in response to the idea of the game being too easy to spam bullets while rotating, the ammo system forces the player to reload after their ammo counter reaches zero which prevents blind fire and encourages making shots count. The delay between reload and shooting is minimal since we still wanted the player to be able to start shooting again quickly.

Scripts:

The scripts we made include: AmmoUI, BulletBehavior, EndGame, EnemyBehavior, ScoreKeeper, Shoot, StartGame, UpdateText, Wavedisplay, ZombieAnimation, and ZombieSpawner. The remaining scripts were provided by the instructor. AmmoUI is a script that updates the ammo counter on the in game UI by fetching data from the shoot script. BulletBehavior is a script that is placed on the prefabs of each bullet upon spawn which controls the direction in which they travel so that they properly fire from where the controller is pointed. EndGame is a script that checks for when the player collides with a zombie, living or recently shot, and disables the interactions the player has and enables the end of game scoreboard. EnemyBehavior controls which direction that the zombies move in, by first making them look towards the player then moving them along their forward axis towards the player. ScoreKeeper is a script that holds the score value as well as contains two methods one to add to the score, which is triggered when a zombie collides with a bullet, and one to get the final score which is called when the game ends. Shoot is the script that holds the ammo system and the simple interaction to shoot through the E key, the ammo system keeps track of the currently held bullets and subtracts when firing and resets the amount upon reloading. StartGame is a script that enables movement, the zombie spawner, and the in game UI once the player hits E when loading in, as well as disabling the starting UI. UpdateText is a script that was created to update the end game display with the amount of killed zombies with the correct value from the score keeper. Wavedisplay is a script that fetches the current level value from the spawner and updates the wave counter in the in game UI. ZombieAnimation gives the zombie animator values for its parameters based on the behavior script attached to each zombie. ZombieSpawner controls the rate zombies are spawned, the current level counter which is related to the difficulty, as the level counter increases zombies become more difficult.

Assets:

We created no new assets instead we found a quality zombie asset for free from the asset store that came with the currently used zombie enemy, as well as animations, which we use the walking, attacking, and falling animation. These assets are located within a folder named zombie within our project that contains all of the assets of this package including some we decided against using. The assets we did use are limited to the zombies themselves, and changing the provided animator to include only the animations we needed and adding parameters that change their state based on their actions.

Discussion:

Overall we ran into few large challenges, with minor challenges that were mainly getting elements to read from each other. The main challenge we ran into was getting the audio of the player to not constantly trigger in scenarios such as before the game starts and the game over screen. The player walking sound would constantly trigger despite not moving as well as not even receiving the movement inputs. We solved this problem by disabling the audio source of the player until they hit the start key and then disabling it again once the game ends.

Improvements:

If we were to improve this game further, we would add powerups, zombie types, and changeable weapons. The powerups would include, infinite ammo and double score for a time, which would give the game a more arcade like flow to it encouraging repeated tries to try and beat high scores as well as a new form of randomization since they wouldn’t always trigger in the same spots or at the same time. Zombie types would diversify the gun gameplay and also contribute to the arcade like feel. By adding a zombie type that might move faster by not attack right away, or a large slow enemy that takes multiple bullets to kill it would make the game more challenging and make the player change the way they play on the spot. Lastly, being able to change the weapon type would further change the styles of gameplay that the user can choose from. For example, a slow firing but large damage weapon type would make the tank zombies easy but make it so they can get overwhelmed easier, while a fast firing but low damage weapon could make dealing with hordes easy but tanks would take even longer to kill. This diversity would make the game appeal to more people as well as bring more replay ability since there are differing ways to play now.