

Reviewee: Hassan Bahzad (Lead Developer, Animator & Graphic Designer)

Tasks:

- Create the foundation of the game mechanics.

Result:

Player Controls:

I wanted to base the movement on Enter the Gungeon as that game has smooth gunplay, smooth animations and above all, smooth movement. Since “Sugar Crash!” is another “bullet-hell” game, I figured trying to utilise a similar movement style would definitely be a positive. The controls of the player are the standard WASD keys, and soon will be compatible with controllers.

The player control script also rotates the player’s currently held weapon by calculating the angle of the pivot point to the mouse position. This makes it so that when the player presses the primary attack/secondary attack buttons, (by default left and right mouse buttons), the weapon plays its firing animation and instantiates the projectile. This projectile is coded to only move forward until either it hits something with the “Take Damage” script or the lifespan time is up.

Camera Movement:

The camera movement was also another feature I really liked about Enter the Gungeon, however, since our game is more set to be open and therefore there is more area to cover, I tweaked the camera smoothness to be faster and had the player be able to see further across the map.

The camera works taking in two separate vector 2’s, one being the player position in relation to the camera, and the second being the position of the mouse also relative to the player camera. The camera script then calculates the average position between the two locations and moves the camera to that point. This allows smooth camera movement and a less staticy feeling movement.

Enemy Behaviour:

As of now, the enemy's behaviour is not complex. The enemy sprites are coded to just move towards the player. The script then checks if the player is within a specific distance and then stops the enemy from moving. The enemy will fire its weapon so long as the player is in a line of sight, this is done by casting a raycast from the enemy to the player, if the ray is not blocked by anything for more than 2 seconds the enemy will take their shot at the player.

Weapon Script:

The weapon script is simplistic, it checks when the player has clicked the primary or secondary action buttons and then triggers that specific event. I wrote this script to also be modifiable to set what type of weapon the current game object is. I.e. semi-automatic or full-automatic. This means I can add guns that fire with a click or continue firing while holding down the action button.

The script also allows custom firing patterns for things such as shotguns, miniguns or other variants of weapons. It does this by making configuring the range the bullet spread has, and the fire rate, easier to access. If, for whatever reason, a weapon that needs to fire in a very specific way is to be added to the game, the weapon script will soon have the feature to call reference to other scripts. Meaning, once the weapon is fired, if the type of weapon is set to custom, it will call a separate script to handle the firing method. That script will be custom made for that specific gun.

Equip/Drop Weapon Scripts:

These scripts are very simple, if the player is holding a gun then when the user presses the interact button and is in range of another gun, then the current held weapon falls out of the players hands and the new weapon is equipped.

****Note:** After reading this, Sean Garrington play tested the game to ensure everything was smoothly in place. ******

Feedback:

The game feels smooth, the shooting is nice asides from the bugs I listed to you. The bugs were as follows:

- When two pickup weapons are in the same place, pressing E picks up both at the same time.
- The camera movement is good, but textures in the background keep jittering.

Other than those two bugs, I feel as though the core mechanics are well placed. Once you add more complicated AI to the game, I think you would be best suited to add more weapons and enemies for variety.

Reviewer: Sean Garrington (Project Manager)