**Prefabs**

**Death zone**: when player get collide with the death zone object, character’s health gets affected

one rate point per second. Used OnCollisionEnter and OnCollisionStay

**Gold**: using cash.play(), audioClip (tagged as “cash”) will play when collects by the player. Value will double if it’s in money mode mario.moneyp = true with “YellowPower” tag . Game object destroyed when collects. PlayerPrefs.GetInt will get the current gold value and PlayerPrefs.SetInt will update the value.

**Powerups:**

LineRender is used to draw free floating lines, IEnumerator starts a coroutine, WaitForSecond used to create a yield instruction to wait for number of seconds, and Physics.Raycast to casts a ray, from point origin, in direction and length, against all colliders in the scene

**1- Hulk**:

will make the character invincible (damage free) for 10 seconds. Invincible = true will trigger the mode and mode will last for 10 seconds. Character will turn green and after that will back to it’s normal color(white)

**2- Jump**:

Character’s jump length is doubled, lasts for 10 seconds. Character will turn blue when get in jump mode and return to white after 10 seconds

**3- Money**:

Character gets twice the of coin, lasts for 10 seconds.

**4- Sniper**:

Shoot laser which kills enemy on mouse click using mouseP to get the position, using ScreenToWorldPoint to transforms main camera position from screen space into world space.

Using DrawRay to draw line from the character to where the mouse points and laser color set to red.

Character turn red and lasts 30 seconds.

Laser goes through multiple enemies and all other objects in the world, destroy object with “enemy” tag. line.enabled =true make the laser visible.

**5- Teleport**:

Makes player teleport to place clicked by mouse. Mouse position stored in moseP and can give its value to transform.position

using ScreenToWorldPoint to transforms main camera position from screen space into world space.

Character is invincible to any damage for 2 seconds after teleporting. Player get’s five powers of this mode to use. teleporting reset the double jumping mechanic

Note: Due to time constraints, we realized there was a better method than the courontines that we used but could not implement it. Once a new powerup was collected in our current method, there was no way to exit the loop which caused issues when collecting new powerups of the same type as they were bound by the time constraint of the first one collected. The new way we would have implemented is to put all of the functions as if statements in update and if an additional powerup was collected it would extend that time by that amount or refresh the amount of charges.

Enemies

**1- Bomber**:

using transform.translate to move bomber in the direction and distance of translation

Bomb’s script: Bomb drops every 5 seconds, position of the bomb gets back to it’s parent using the parent transform position

Make the renderer object visible inside the bomber enemy after 5 seconds using renderer enabled

Detecting collision with player character using MonoBehaviour.OnCollisionEnter

OnCollisionStay called once per frame for every bomb touching player character

**2- Exploder**:

to create an explosion, Particle system used to create a special main module that has multiplier properties to choose the emitter shape, radius, lifetime, and size.

apply an explosion force to all objects in radius using (Physics.OverlapSphere)

character will get affect by the explosion depending on its position (health reduces, and force applied)

damage script also applied to this enemy

**3- Ghost:**

will change its color depending on the character’s position to the ghost component

Between 5 and 2 enemy will starts to lose its color and when character get passed 2 enemy will be invisible.

**4- Patroller**:

moves between 2 points min and max (beginning and end of the platform) using transform. Translate to move the character on the platform.

**5- Thrower**:

That though bouncy projectiles. load bullet that stored in resources folder using Resources.Load(“bullet”)

InvokeRepeating ("Bullets", 2, 5) will repeated 5 times with 2 second period. Bullet will destroy after 5 second.

Projectiles get rigid body component with gravity enabled.

Character (Mario script):

* Use variable (hit) to get the information back from raycast
* Physics.Raycast to casts a ray, from point origin, in direction and length, against all colliders in the scene
* By checking this conditio !onGround && secondjump = true, second jump will disable secondjump = false.

If character onGround, second jump enabled secondjump = true

* Speed will double when hit the left control key Input.GetKey(KeyCode.LeftControl, speed = 2\*orignalspeed
* Character dies when fall though gaps or character health exhausted

**GUI**

* splash screen scene contains the name of the game and takes left shift key input to go to the Main Menu that has three different levels, store, and about section. Using SceneManager.loadScene to load the scene.
* Using DontDestroyOnLoad make the target object not to be destroyed automatically when scene load.
* Enemieskilled.ToString will display how many enemies killed
* PlayerPrefs.GetInt ("Gold").ToString () stores and access the gold value, return the value and print it on screen
* mario.teleportpower.ToString () to print the count of the teleport powerups
* When hit pause sound stop and when hit play button sound resume. Has also current level that the player is on, and a button that exits to the main menu.

<AudioSource> used to control the audio output and represent the audio clip

**Procedural Generation**

Each level is procedurally generated, each prefabricated platform has up to 5 spawnables which includes, enemies, power ups, deathzones and gold. The x coordinate is used for determining where everything spawns. Once a platform has spawned the new x is x + length of platform + 1 in order to make room for the next platform.

When player is half way through in each scene, the next one is constructed. The section destroyed when its off screen.

Gaps created between platforms, if player fall through them it dies. Crossing each gap given the physics of the game. Each section

**Store:**

* Store create so that player can upgrade the character health, gold, and boost speed, zero gravity ability.
* onClick.AddListener event triggered when the button pressed
* Add Gold button will add 5000 gold to the gold counter (only for testing)
* Reset Game Data will set gold value to zero, reset character health to 100, speed back to normal, and coinboost to 2 and remove zero gravity ability
* Buying speed boost or health upgrade costs 100 gold to increase the speed/health by 10%, each buy increases the speed/health by 10% up to 150%. Second buy costs 200, third costs 500, fourth 1000, fifth 1000.
* Buying coin multiplier costs 100 gold to boost gold x2 each buy increases the boost by extra 1 boost. Second buy costs 200, third costs 500, fourth 1000, fifth 1000.
* Small 5 bars under each upgrade to indicate how many items player got, and change its color from white to black indicate that we bought one of the items from the store.

**General Note:**

* **All audio was retrieved from the internet (Bensound and SoundBible), we do not have licenses for the sound used in the game**