# Process Log

1. Created a new Unity Project and changed the name of the scene to ‘main’
2. Publish repository to GitHub
3. Download from Kenny’s Assets https://www.kenney.nl/assets/platformer-pack-redux
4. Created Prefab for Player and Platform
5. Set Resolution to 1920x1200 for Android tablet
6. Create Player Controller Script with purpose of being able to move and jump
7. Create button for player to jump for mobile
8. Add to Player Controller Script to check if player is grounded if so then you can have the ability to jump, otherwise you can’t jump infinitely.
9. Create Camera Controller Script to the follow the player as it traverses through the level
10. Tested the Camera script by creating a long cube and having the player traverse through that and works as expected
11. Found a bug where the player sticks to the side of the platform to fix you create a material
12. Created a platformer generator script that generates platforms infinitely.
13. To optimize performance for our game we need to be able to delete platforms that we’ve already gone past so as to save memory. So I created a platformer destroyer script that is similar to the generator script.
14. Created Android Build so as to test the game on Android since we’ve got an endless environment and a player character that can move and jump. Everything works as expected.
15. Added a Random.Range for the distance between platforms as to make the game more challenging and engaging.
16. Created a pickup Item (coin) that the player can pickup to increase his score (created score system).

17. Created an array of platforms that can be selected one of the platforms has a coin and other has a bomb which triggers the reloading of the scene which is the end state for this game.
18. Downloaded a zip of my github and opened the project in unity to make sure everything works properly

## How the functional requirements were completed

JUMP: Increase the velocity of the rigidbody of the player gameObject by the variable jumpForce, this is wrapped in a condition that the player is touching the ground via Physics2D.IsTouchingLayers.

Interact with Obstacles/Collect items: Created a bomb/coin gameObject and attached the item script {which has OnTriggerEnter2D()} and do different events based on whether it's a bomb or a coin. If bomb reload scene. If coin increment score by 1.

Endless Platforms: I've attached two game Objects to the main camera which are a generationPoint and a DestructionPoint. And at setDistances a platform is generated based on its width and distanceBetween(which is randomized) and if the platforms position.x is less than the DestructionPoint.x destroy that platform.

Scoring System: Created a text canvas object and made a reference to it in score manager and call the function AddToScore() in item.cs when collide with coin.

End State: The end-state for this game is defined as reloading the same scene when player falls off platform or collides with bomb.

Functionality that can be built to mobile: You can build directly to pc or android and the jump mechanic is done via button on android.

## Notes:

Spacebar or press button to jump

This project was made over the weekend and couldn’t do it Wednesday-Friday because of my full-time placement.