# TUTORIALS

**Scoring System**

Link to the tutorial video: <https://www.youtube.com/watch?v=D0lx90n0s-4&ab_channel=JimmyVegas>

In this tutorial you get to collect collectables whilst the player collides to the prefab, learn how to make the main camera move in sync with the player's movement, you’ll learn that the velocity tab needs to have a value to be able to move around the terrain, you get to learn how to trigger sfx once a player collided with a prefab.

**ScoringSystem:**

We gave two variables for this script one as a gameObject for the score text, other as a static int for the score. We set the variable to a getComponent to show the score on the scene and add points to it as the player gradually picks up more collectables.

Text

Description automatically generated

**CameraMovement:**

For the cameramovement to sync the velocity of the character movement we added the public gameObject variable and a private Vector3 variable as well, on the "VOID START" we set the Offset variable = the transform position by subtracting it with the gameObject's transform position, for the "LATEUPDATE" we got the regular transform position = the players transform position + offset

Text

Description automatically generated

**TriggerSFX:**

We add a variable with audioSource to be able to play the collectable sfx once the character collided with the asset, we added a scoring system on the onTrigger Method to be able to add score once the player collies with the collectable. and a destroy function to once the player has collected the asset.

Text

Description automatically generated

**Destroy a Cube into pieces**

Link to the video: <https://www.youtube.com/watch?v=s_v9JnTDCCY&t=3s&ab_channel=LearnGameCreating>

We first start with creating a plane floor and adding a cube as a gameObject prefab, we add a rigidbody to the cube so any gravity force could be applied, we added a box collider to the floor as well to make sure to cube and floor could add physics and collide with each other to start up the commands given to the script.

We create a script to where we tell the cube to explode/disappear once the cube collides with the floor.

Text

Description automatically generated

After that we write in the script where once the cube collides with the floor it disappears and another cube spawns with a smaller size given to the variable given this method lets us create a primitive objectives for our code, we can set the scale and position of where the cube will be created after the impact of the main cube hitting the floor, we add a rigidbody component to add mass to the cube. we then add explosion Force to separate the cubes apart from each other.

Text

Description automatically generated

**Basics of Shader Graph**

Link to the video: <https://www.youtube.com/watch?v=Ar9eIn4z6XE&ab_channel=Brackeys>

With shader graph we don’t have to go through the hassle of writing a new shader script, but we can use shader graph for everything visually and in a fraction of a time.

The shader graph doesn't have alot of programming, but it has a feature called notes where you have a graph canvas and can bring different types of effects to the graph to change the looks, visuals and styles of any material given to a model. this features only works in the RP type of projects where post processing can be included in the hierarchy.

in the shader we learnt how to add emission material on a model, learned how to animate it to fade from 0-1, learned that we can switch the mesh material preview in the shader graph by right clicking and choosing your custom mesh you have in the scene.

**3rd Person Parkour movement**

Link to the video: <https://www.youtube.com/watch?v=XAC8U9-dTZU&ab_channel=DanisTutorials>

in the video above the main concept is a fps parkour movement but I accidentally placed my camera in a way that was set to make it 3rd person, so I stick to it.

The first thing to be done is to design a simple level with cubes, add mesh colliders on the gameObject and add a rigidbody and capsule to the player.

Text

Description automatically generatedText

Description automatically generated

On the playermovement script there are assignable like playercam & Orientation, we use rotation to look with the mouse, use movement speed and the maximum value with a physics material layer mask for the ground, counter movement threshold and slope angle as variables for movement, for crouch & sliding we use crouchscale, playerscale, slideForce, slideCounterMovement, there's jumping involved which as well has a cooldown and force as variables.

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

the issues I’ve ran across during this process is changing the gravity on the project settings, so the player characters jump feels more realistic!

The Player’s Camera we make sure that it follows the player’s movement at all time where ever the camera gets placed at it helps to give the players a view to be able to see their surroundings and being able to aim whilst moving around and not having to deal with camera issues like random shake screens or losing its balance while moving.

A screenshot of a computer

Description automatically generated with medium confidence

**Speedometer**

Link to the video: <https://www.youtube.com/watch?v=3xSYkFdQiZ0&t=28s>

In this video we learn how to code a speedometer for one of the level design modules work we have; our main goal is to go from zero velocity to the max velocity while using some assets to give us an estimate of how fast the car is moving.

we first find out the zero and the max point of velocity and mark the coordinates in the canvas. then we tell it to whenever the car is moving rotate the needle to its matching velocity and coordinates to rotate from. we then write some labels for the background of the speedometer to show its value in game.

We learn how to make the arrow rotate in the canvas to match the force and velocity of the vehicle moving by calculating the velocity of the car how long the acceleration input is clicked.

Text

Description automatically generatedText

Description automatically generated