جـــامعة ٦ أكـــتوبر
OCTOBER 6 UNIVERSITY

علوم الحاسب ونظم المعلومات
Info Sys & Comp Science

Multimedia Project

(Battle Game Using Unity)

Code : 422

- Rawan Mustafa Abd-Elhamid (Is)
- Ahmed Amr Mohamed (CS)
- Mustafa Magdy (CS)
- Yossef Rabie Ahmed (Is)
- Mazen Hassan
- Ahmed Mahmoud Ibrahim (Leader – CS)

Under supervision

Dr. Bassem Sheta

Content

Idea :- battle game and It depends on two main characters

(hero and enemy)

Describtion:-

- Game that's main engine is UNITY and it's purpose is to combine between different media like (image – animation – audio) Based on many tools that we used.

Tools:-

- UNITY – Unity Animator
- Visual Studio 2019

Programming Language:-

- C#

Project Phases:-

1. **Import Assets.**

2. **Cinemachine Camera.**

3. **Colliders.**

4. **Player Movement.**

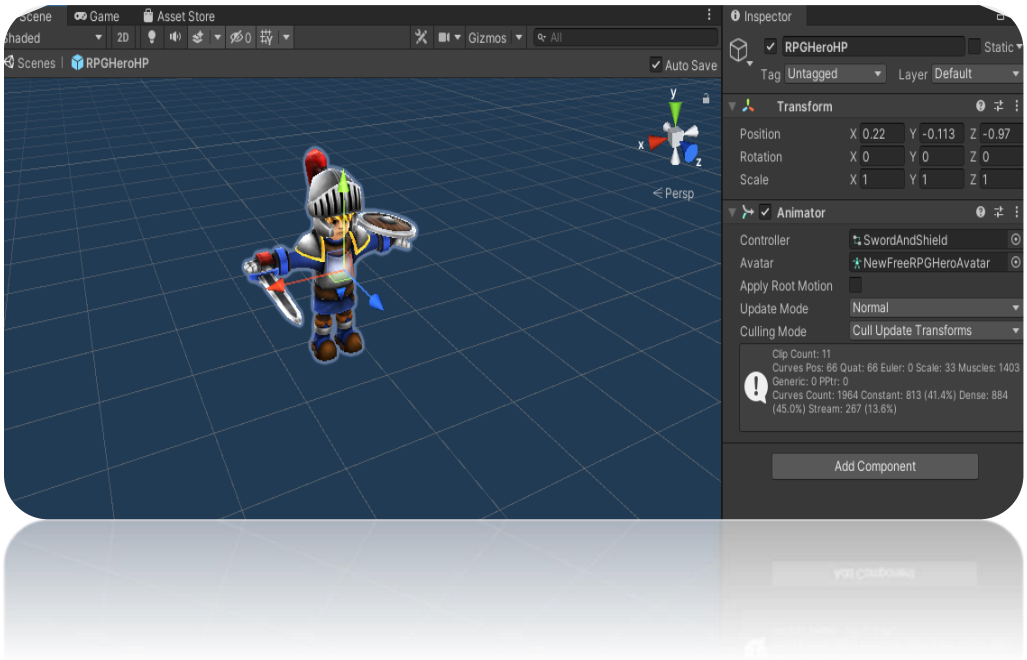
5. **Jump & Gravity.**

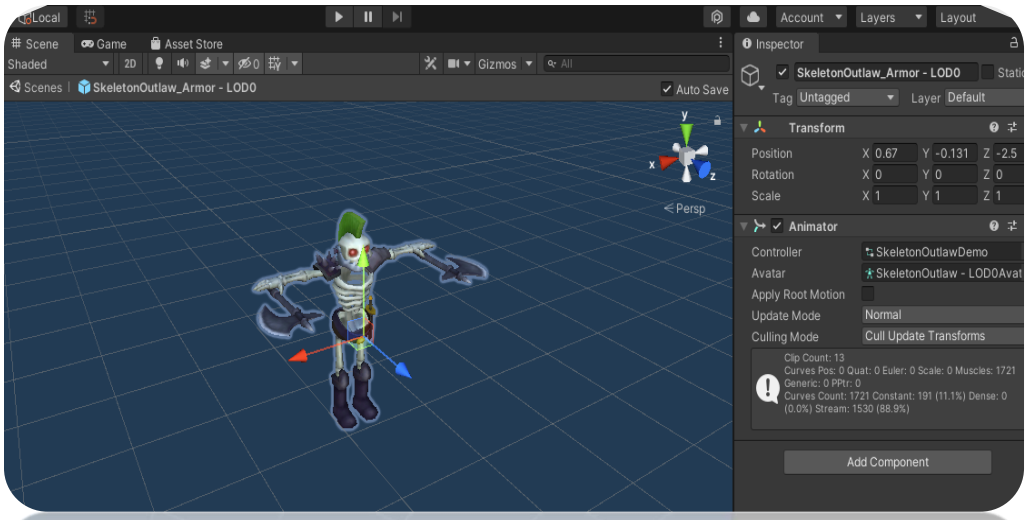
6. **Animations.**

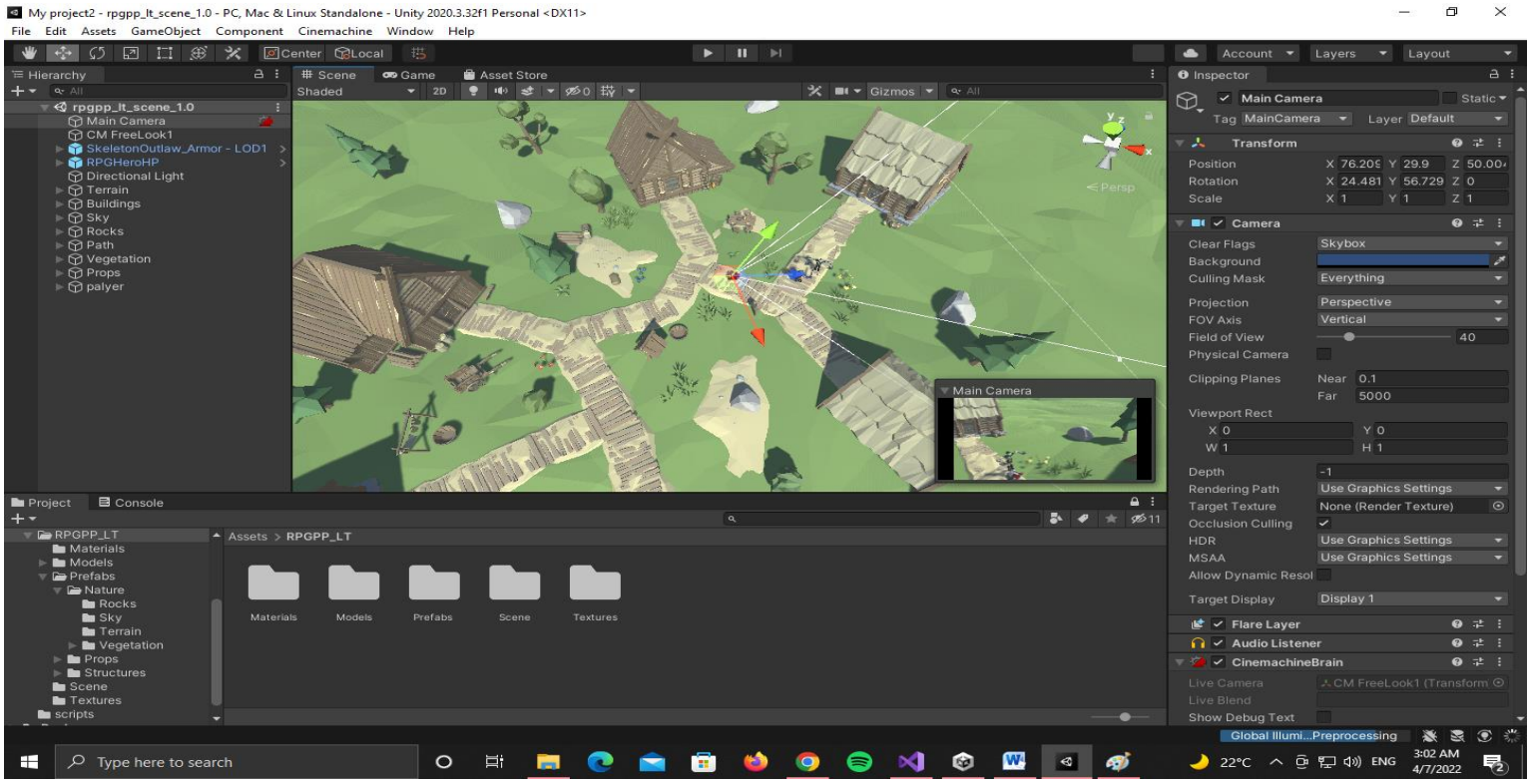
7. **Sound Effects**

1- Import Assets.

- Import Main Assets Like Place, Hero and enemy from UNITY store.



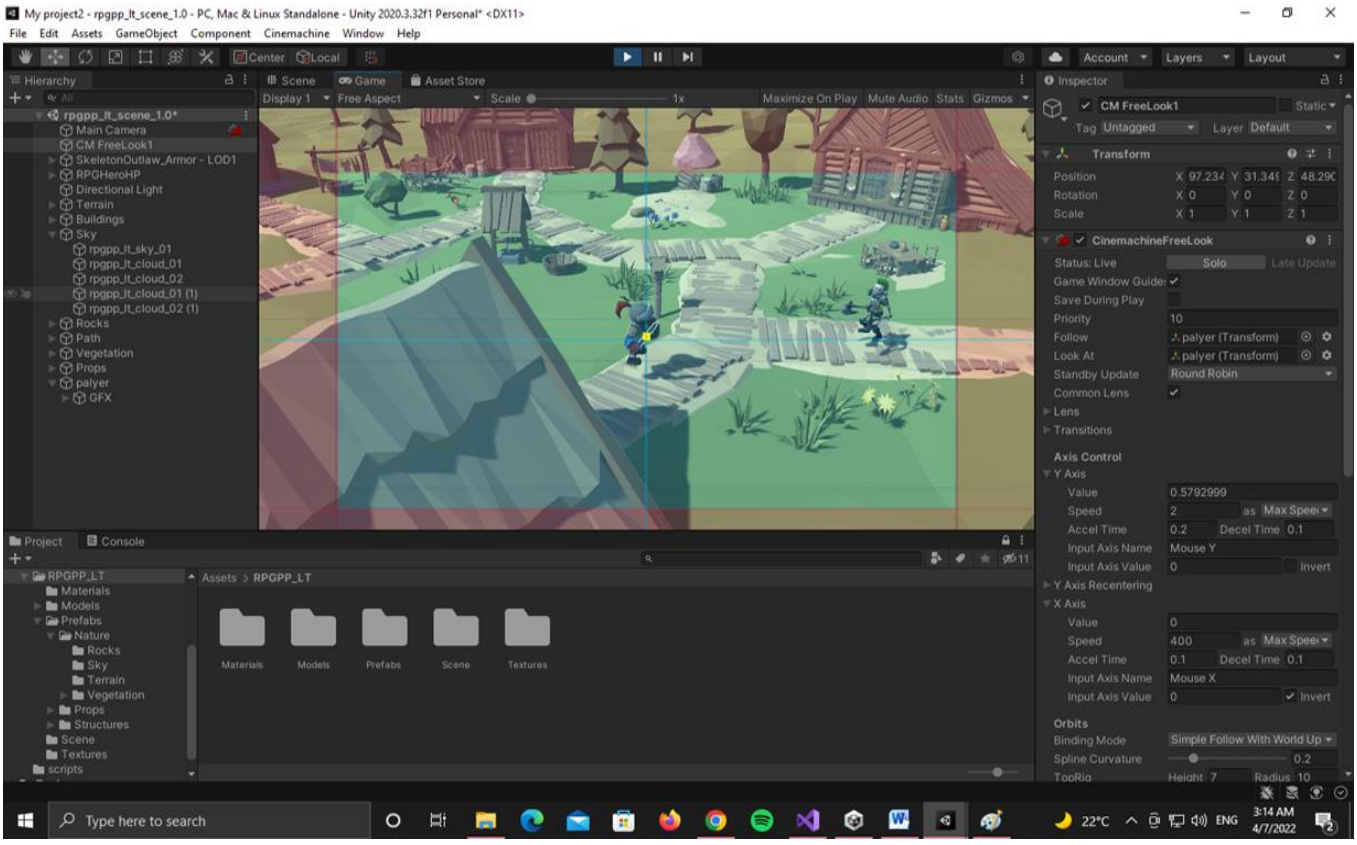

Local
Account
Layers
Layout
Scene
Game
Asset Store
Shaded
2D
Gizmos
Scenes
SkeletonOutlaw_Armor - LOD0
Auto Save
Persp
Inspector
SkeletonOutlaw_Armor - LOD0
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Animator
Controller
SkeletonOutlawDemo
Avatar
SkeletonOutlaw - LOD0Avat
Apply Root Motion
Update Mode
Normal
Culling Mode
Cull Update Transforms
Clip Count: 13
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 1721
Generic: 0 PPtr: 0
Curves Count: 1721 Constant: 191 (11.1%) Dense: 0
(0.0%) Stream: 1530 (88.9%)
Add Component


My project2 - rpgpp_lt_scene_1.0 - PC, Mac & Linux Standalone - Unity 2020.3.32f1 Personal <DX11>
File
Edit
Assets
GameObject
Component
Cinemachine
Window
Help
Hierarchy
rpgpp_lt_scene_1.0
Main Camera
CM FreeLook1
SkeletonOutlaw_Armor - LOD1
RPGHeroHP
Directional Light
Terrain
Buildings
Sky
Rocks
Path
Vegetation
Props
palyer
Main Camera
Inspector
Transform
Camera
Clear Flags
Skybox
Background
Culling Mask
Everything
Projection
Perspective
FOV Axis
Vertical
Field of View
Physical Camera
Clipping Planes
Near
Far
Viewport Rect
Depth
Rendering Path
Use Graphics Settings
Target Texture
None (Render Texture)
Occlusion Culling
HDR
MSAA
Allow Dynamic Resol
Target Display
Display 1
Flare Layer
Audio Listener
CinemachineBrain
Live Camera
Live Blend
Show Debug Text
Project
Console
RPGPP_LT
Assets > RPGPP_LT
Materials
Models
Prefabs
Scene
Textures
Nature
Rocks
Sky
Terrain
Vegetation
Props
Structures
scripts
Global Illumi...Preprocessing
Type here to search
22°C
ENG
3:02 AM
4/7/2022

2- Cinemachine

- Cinemachine is a Unity asset that quickly and easily creates high-functioning camera controllers without the need (but with the option) to write custom code.

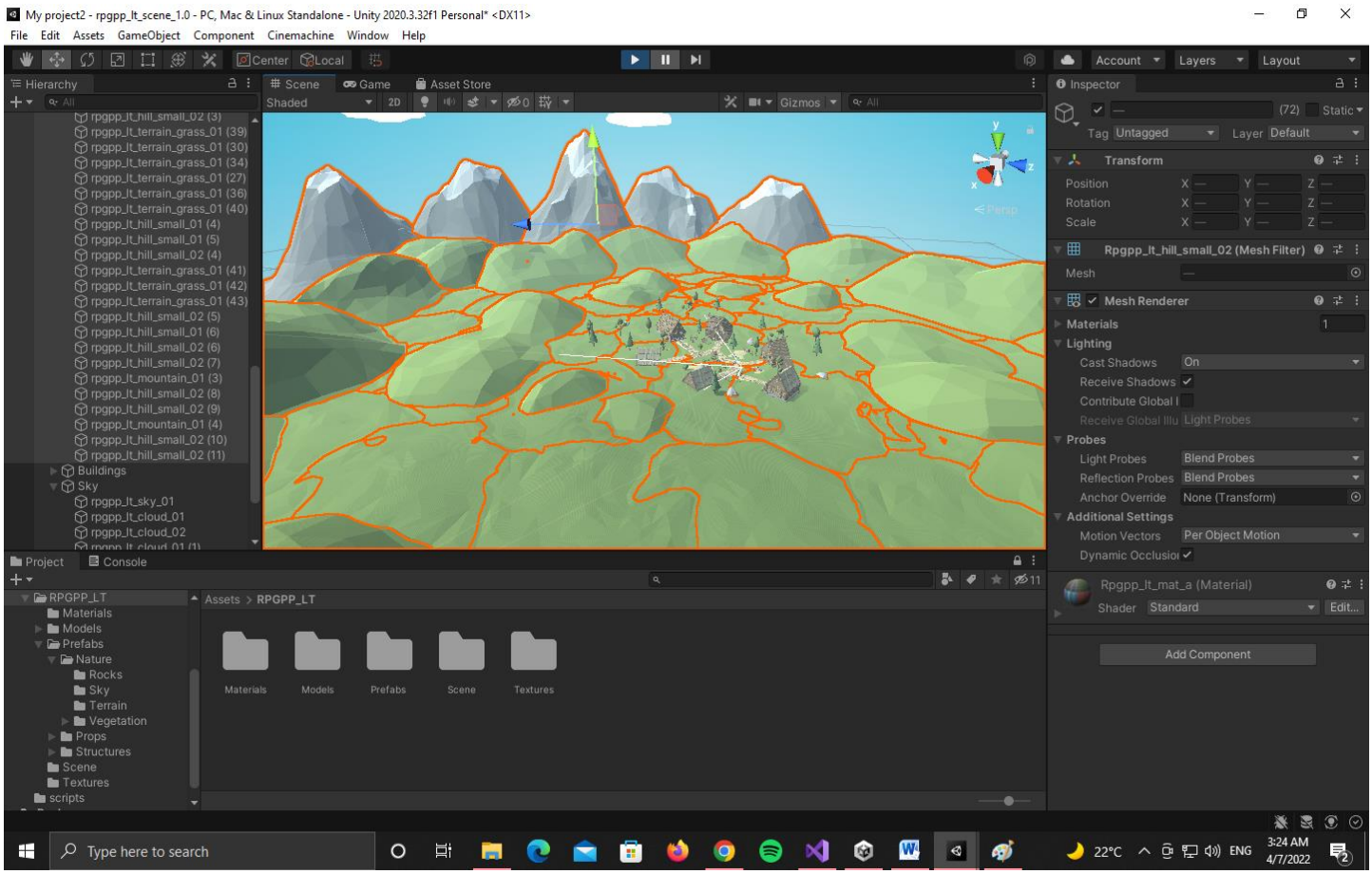
- Cinemachine does not add more camera components to scene, but instead makes use of so-called “virtual cameras.” These virtual cameras control the position and rotation of the Unity camera - you can think of a virtual camera as a camera controller, not an actual camera component.



3- Coliders

- In This phase Unity handles collision between GameObjects with colliders, which attach to GameObjects and define the shape of a GameObject for the purposes of physical collisions.

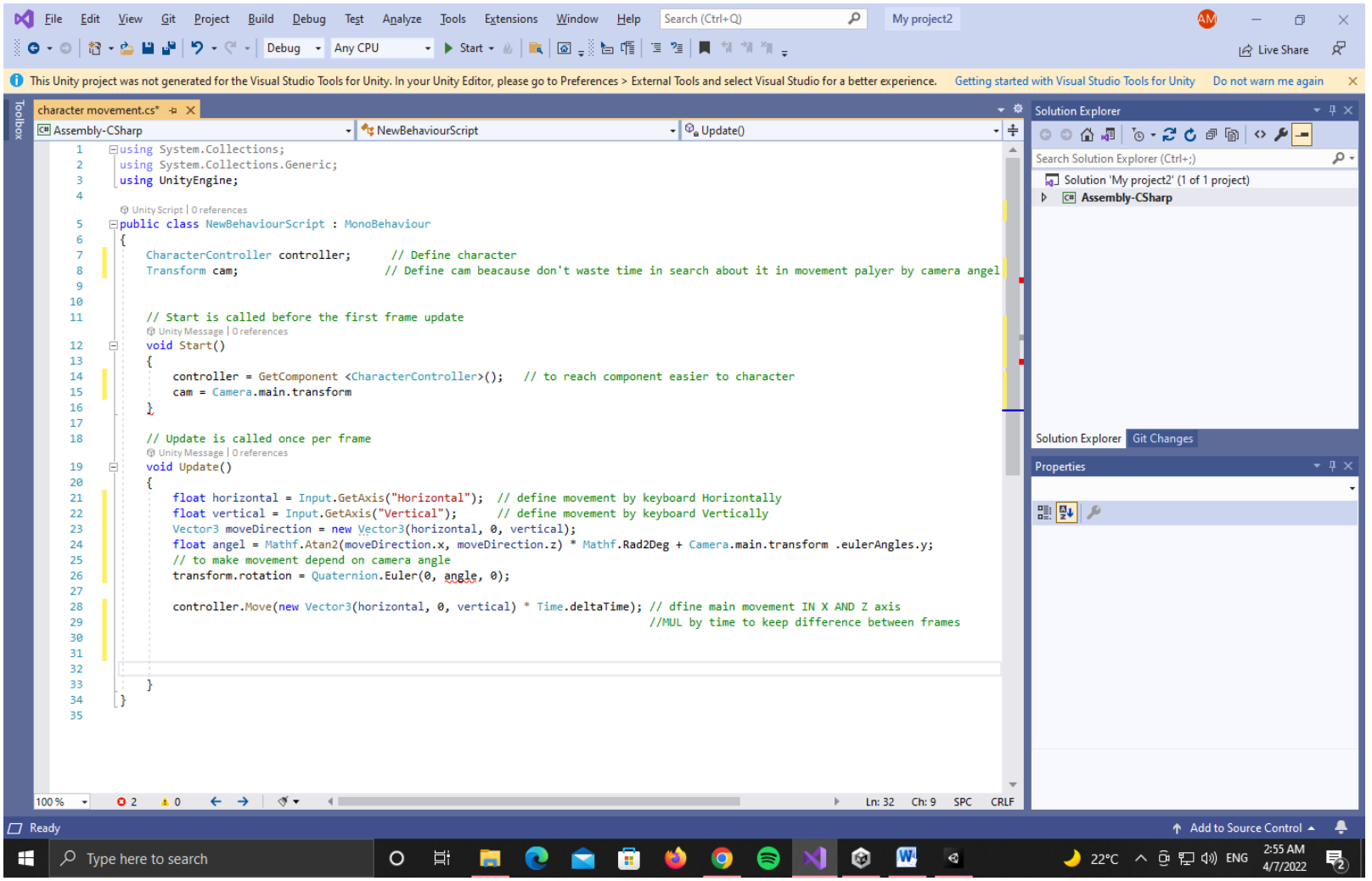
- A collider is invisible, and does not need to be the exact same shape as the GameObject's mesh



4- Player Movement

- Supplies the movement of a GameObject with an attached CharacterController component.

- The CharacterController Move motion moves the GameObject in the given direction. The given direction requires absolute movement delta values. A collision constrains the Move from taking place. The return, CollisionFlags, indicates the direction of a collision: None, Sides, Above, and Below. CharacterController.Move does not use gravity.



5- Jump & Gravity.

- The Rigidbody component in Unity allows an object to move under physics simulation.

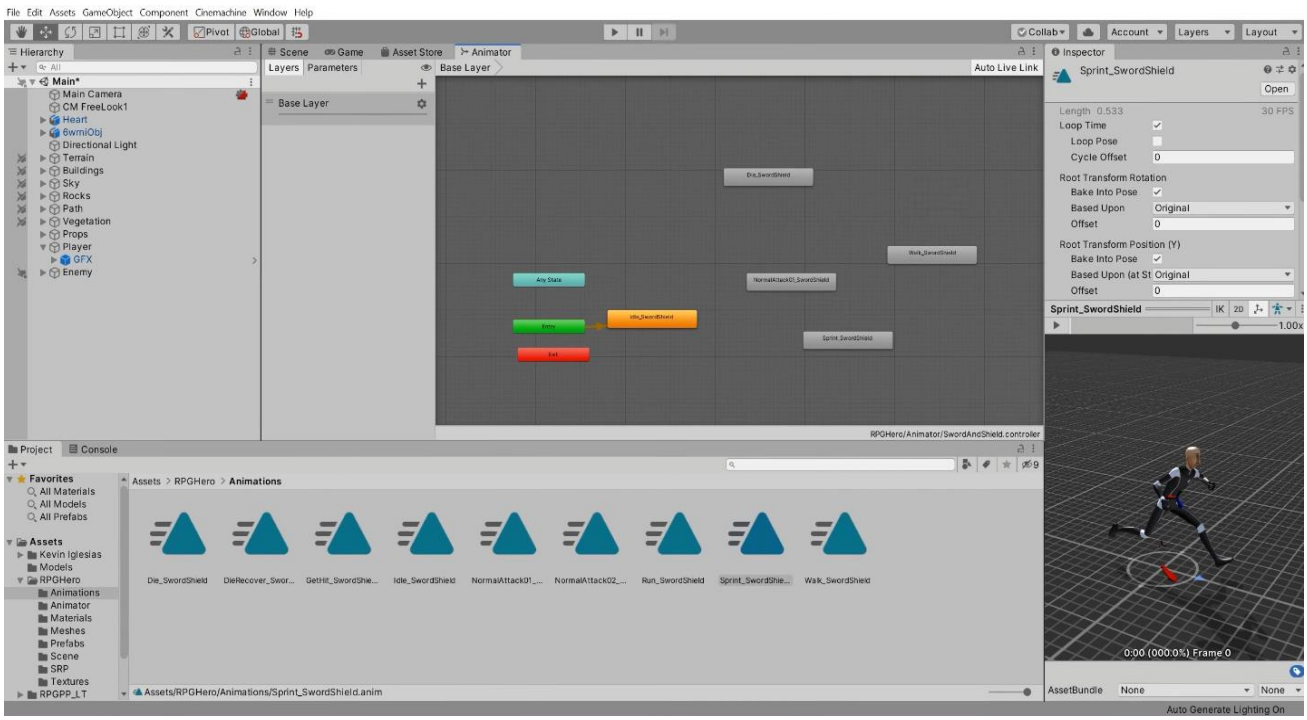
- Meaning that it will be affected by gravity, it can be moved using physics forces and, when used with a Collider, it will collide into other objects that also have Collider components attached.

```
19          cam = Camera.main.transform;
20      }
21
22      // Update is called once per frame
        Unity Message | 0 references
23      void Update()
24      {
25          float horizontal = Input.GetAxis("Horizontal");
26          float vertical = Input.GetAxis("Vertical");
27          Vector3 moveDirection = new Vector3(horizontal, 0, vertical);
28
29          if (Controller.isGrounded)
30              if (Input.GetAxis("Jump") > 0)
31                  verticalVelocity = jumpValue;
32                  else
33              verticalVelocity -= gravity * Time.deltaTime;
34
35          if (moveDirection.magnitude > 0.1)
36          {
37              float angle = Mathf.Atan2(moveDirection.x, moveDirection.z) * Mathf.Rad2Deg + cam.eulerAngles.y;
38              transform.rotation = Quaternion.Euler(0, angle, 0);
39          }
40          moveDirection = cam.TransformDirection(moveDirection);
41
42          moveDirection = new Vector3(moveDirection.x, verticalVelocity, moveDirection.z);
43          Controller.Move(moveDirection * Time.deltaTime * speed);
44      }
```

6- Animations

- Unity's Animation features include retargetable animations, full control of animation weights at runtime, event calling from within the animation playback, sophisticated state machine

- hierarchies and transitions, blend shapes for facial animations, and much more.



7- Sound Effects

- The Audio Source plays back an Audio Clip in the scene. The clip can be played to an audio listener or through an audio mixer. The audio source can play any type of Audio Clip and can be configured to play these as 2D, 3D, or as a mixture (SpatialBlend).