Guest Lecture

Wednesday, July 27th

Making Games With Unity

LINK TO SLIDES:
https://tinyurl.com/53kvuw7m

CS 160/260 • User Interface Design and Development · Summer 2022
Noah Schwartz

What Is An Engine?

What Is An Engine?

IDE v.s. Engine

What Is An Engine?

IDE

Code + Compiler
(with minimum graphical display)
Visual Studios, IntelliJ, Atom, etc.

v.s.

Engine

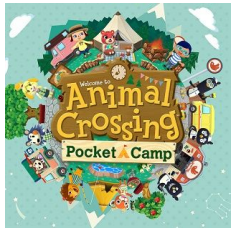
An advanced graphical framework built on user interfaces and pre-done work to make a game.
Lots of behind the scenes work.
Uses IDE's for the coding portions.
Unity, UnReal, GameMaker, etc.

Why Unity

Why Unity

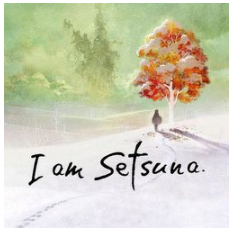

GENSHIN
IMPACT


OVERCOOKED!


Animal Crossing
Pocket Camp

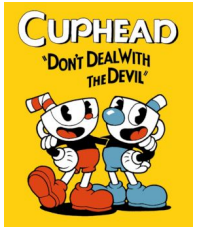

SUPER MARIO RUN

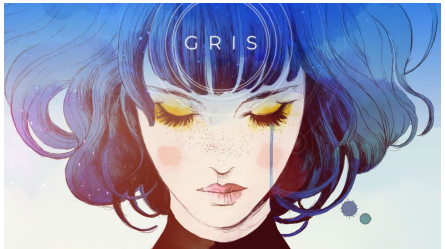

HOLLOW KNIGHT


I am Setsuna.


AMONG US


Untitled
Goose Game
by House House


CUPHEAD
"DON'T DEAL WITH THE DEVIL"


GRIS


POKÉMON
GO

Why Unity

Scalability

Make large-scale projects and develop Triple A games.

Ease of Use

Simple UI so any new programmer can make a game at any level.

Diverse Features

Various ways to accomplish the same task. Allows creative freedom.

Free

Free to install the Personal Version. Once you make over 100K a year for your game, you then switch to paid versions.

Why Unity

Scalability

Make large-scale projects and develop Triple A games.

Ease of Use

Simple UI so any new programmer can make a game at any level.

Low Floor

Wide Walls

High Ceiling

Diverse Features

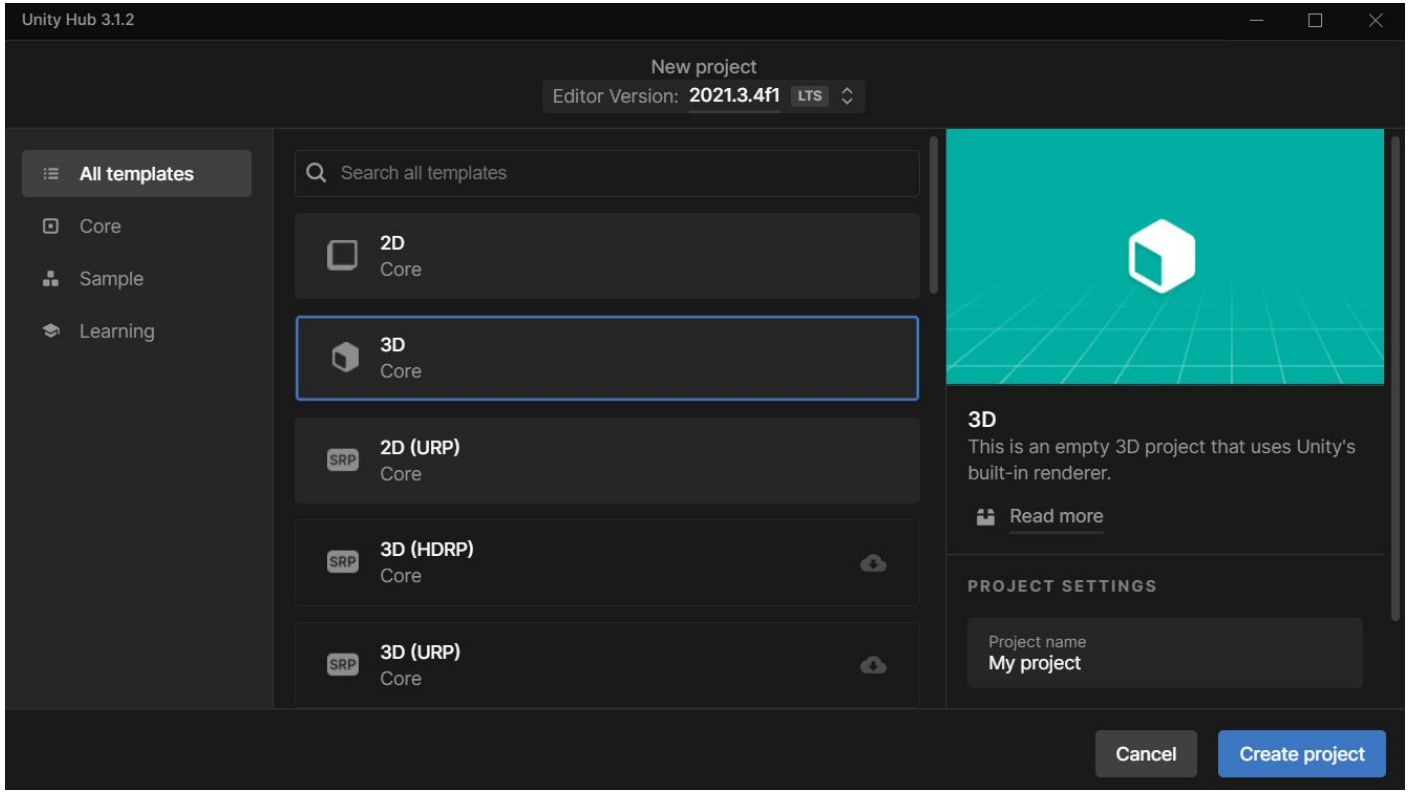
Various ways to accomplish the same task. Allows creative freedom.

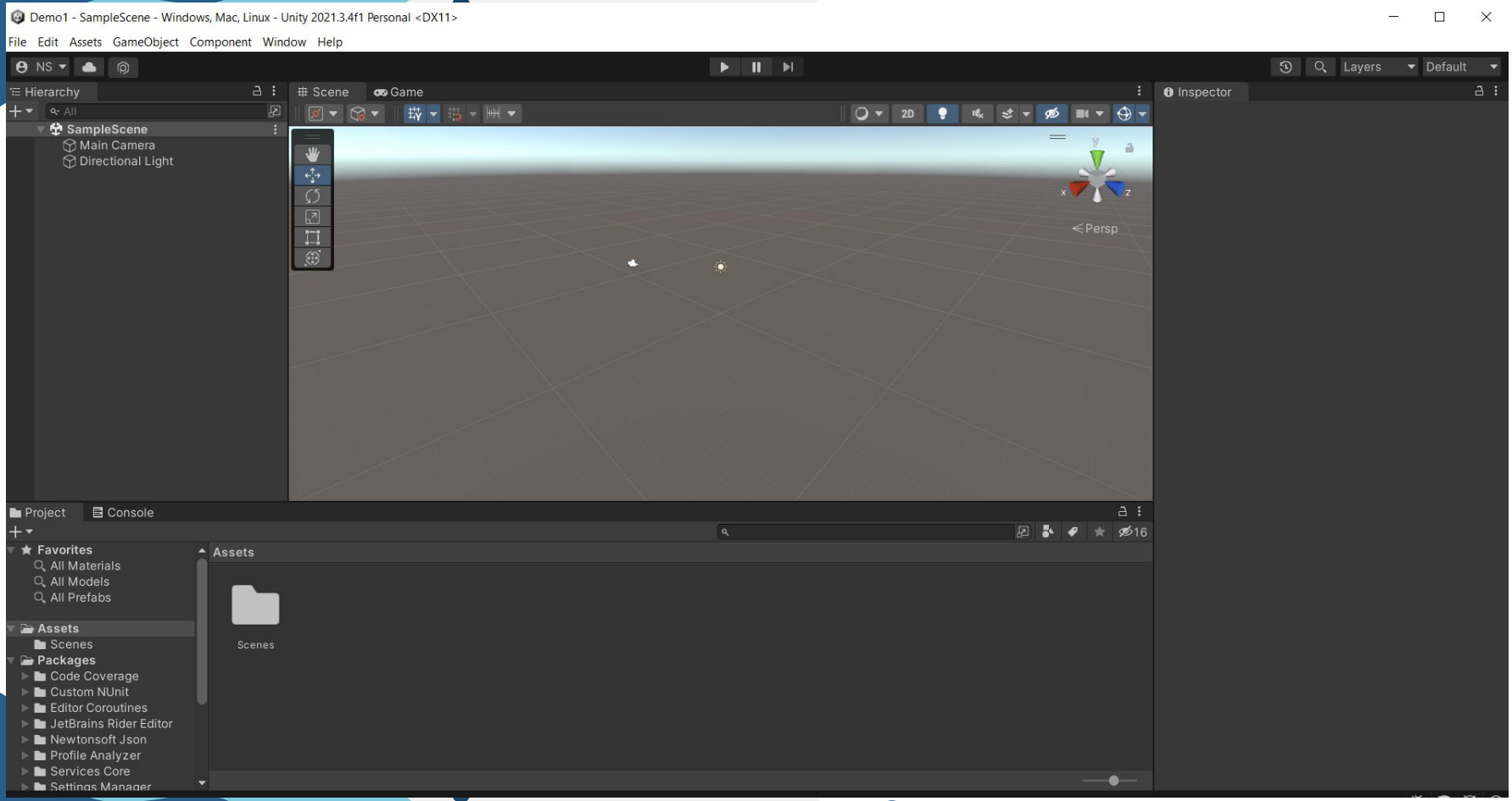
Free

Free to install the Personal Version. Once you make over 100K a year for your game, you then switch to paid versions.

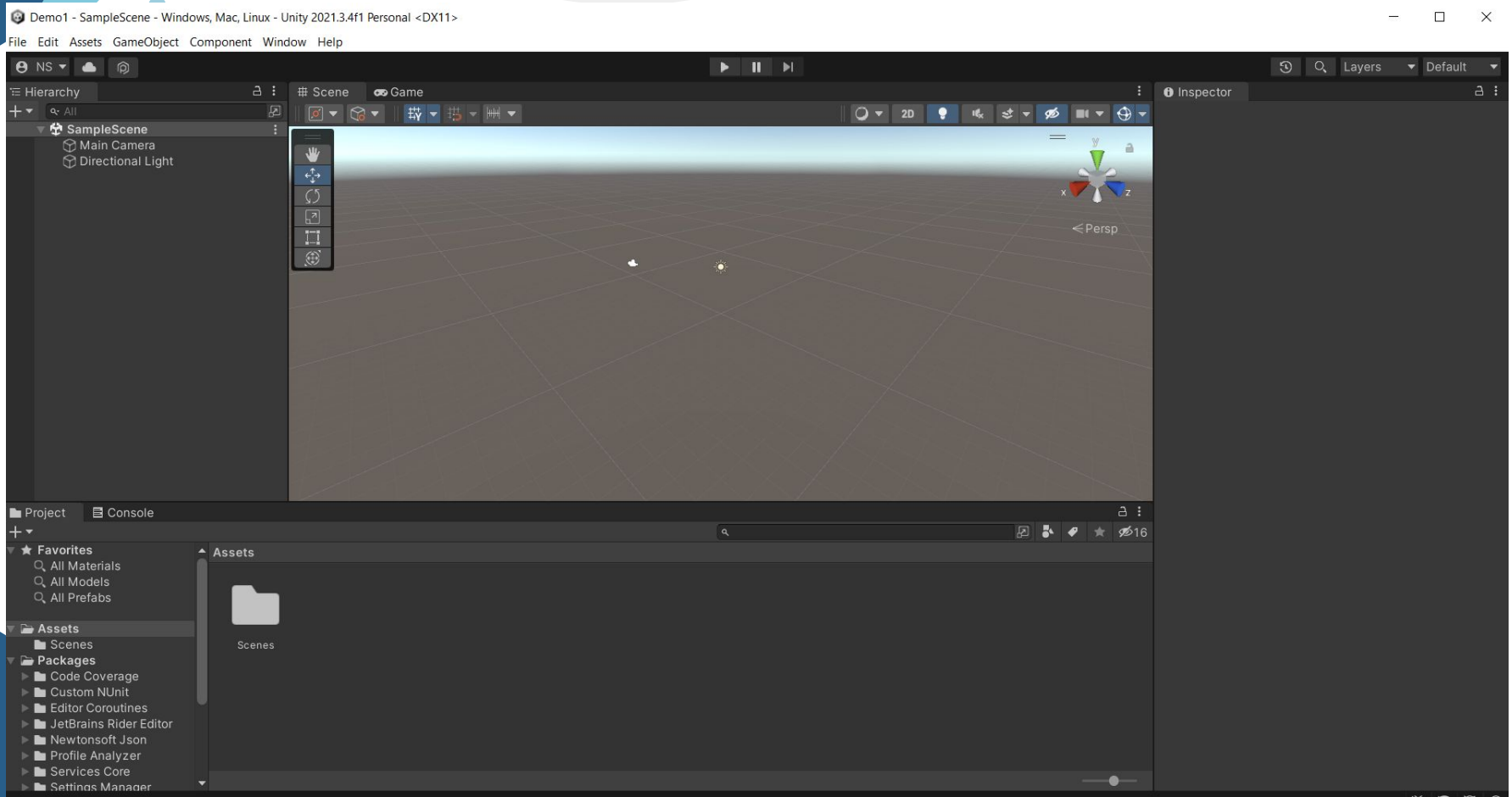
LOW FLOOR
Getting Started With Unity

LOW FLOOR
Getting Started With Unity

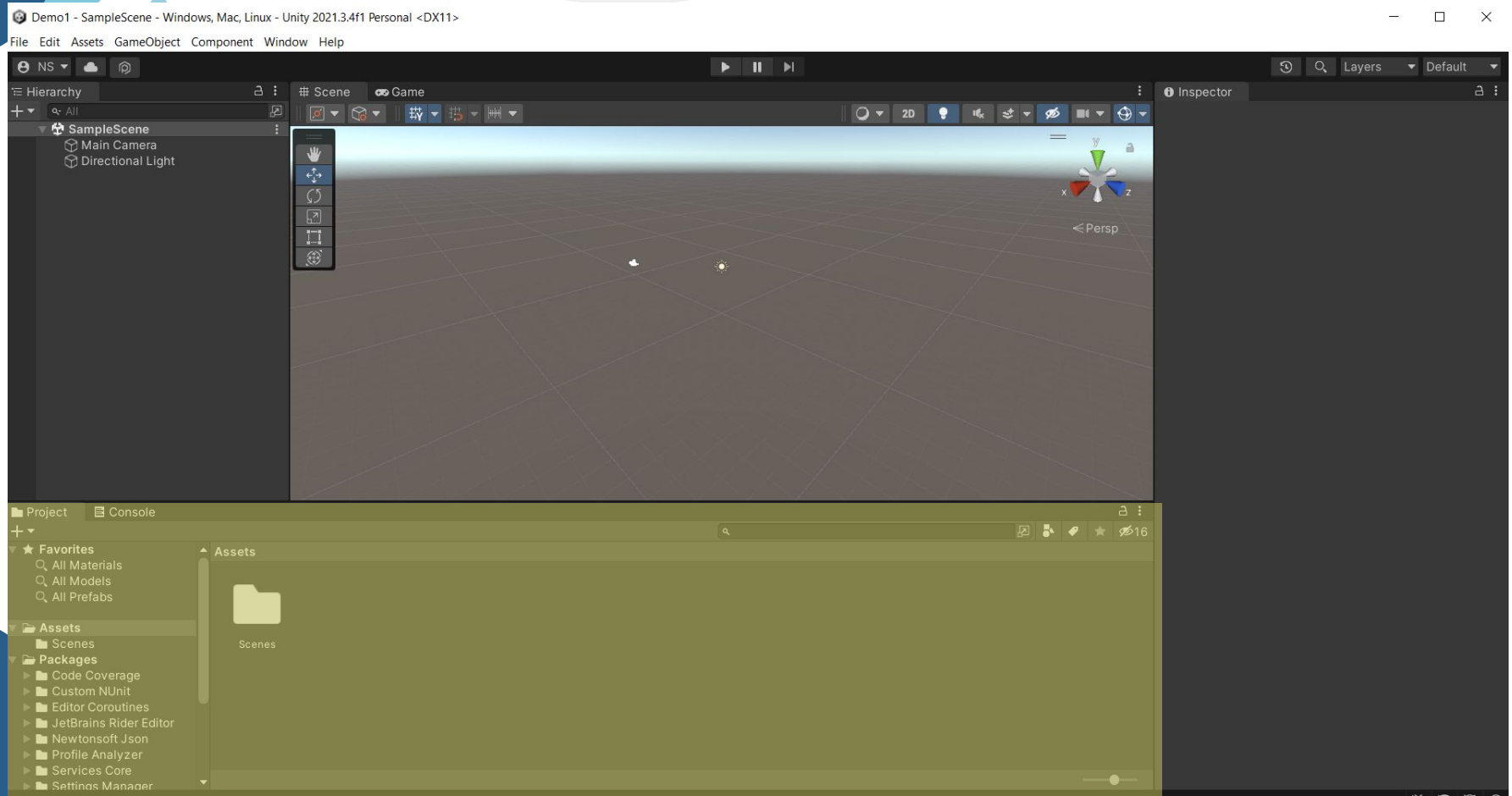



Demo1 - SampleScene - Windows, Mac, Linux - Unity 2021.3.4f1 Personal <DX11>
File Edit Assets GameObject Component Window Help
NS
Layers
Default
Hierarchy
All
SampleScene
Main Camera
Directional Light
Scene
Game
2D
Persp
Inspector
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Packages
Code Coverage
Custom NUnit
Editor Coroutines
JetBrains Rider Editor
Newtonsoft Json
Profile Analyzer
Services Core
Settings Manager
Assets
Scenes

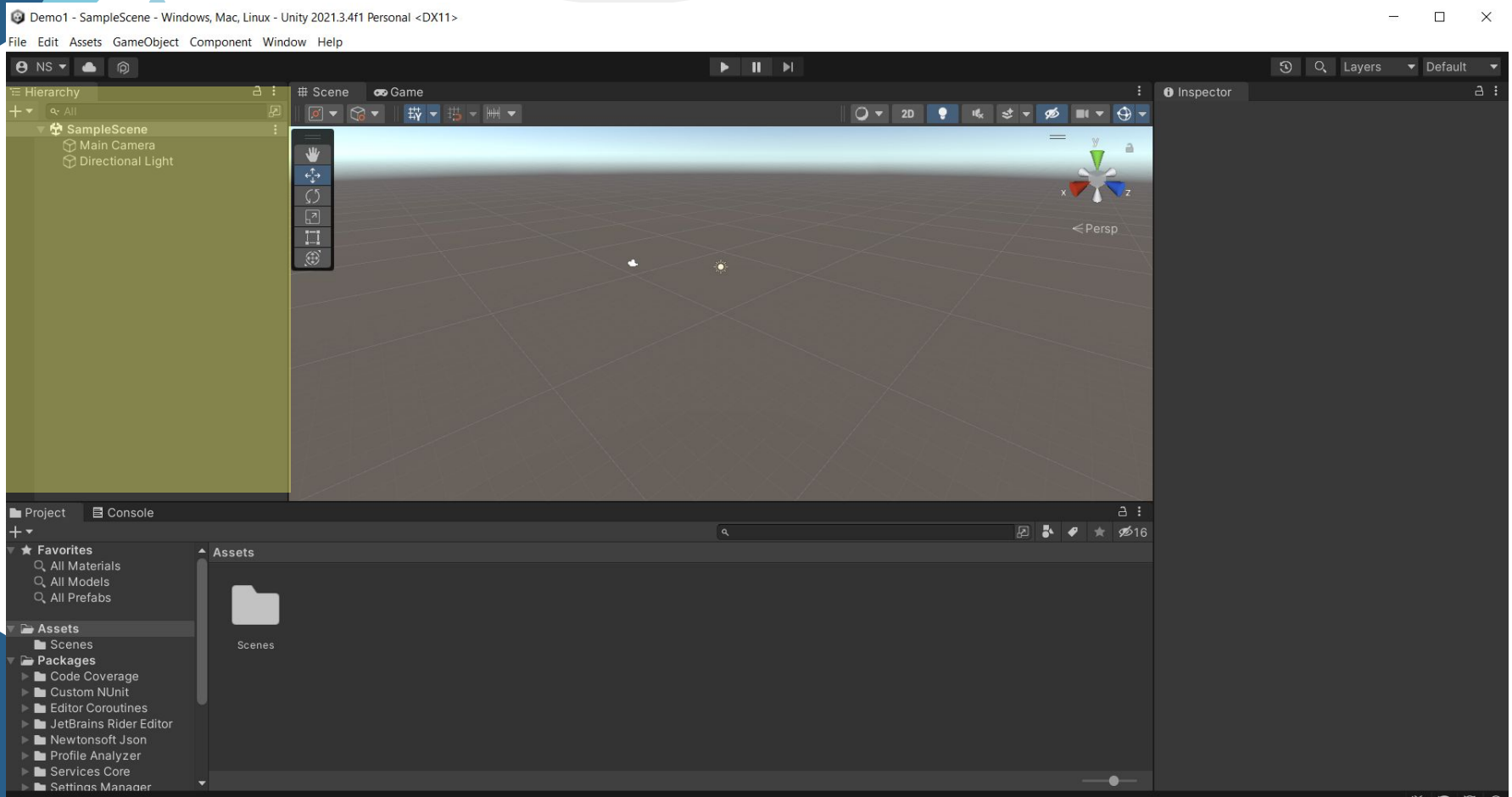
What UI elements stand out to you right away?



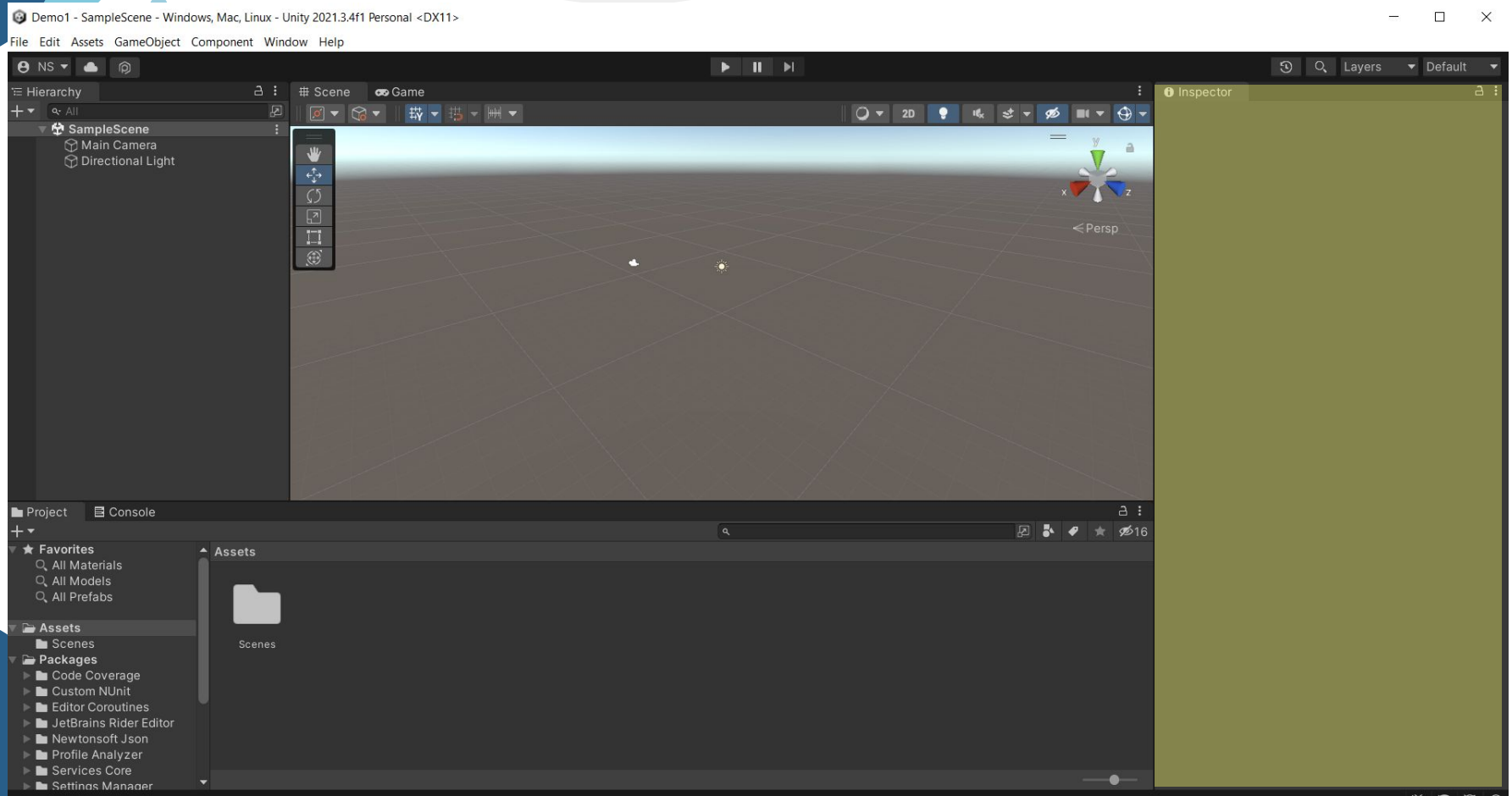
Assets


Demo1 - SampleScene - Windows, Mac, Linux - Unity 2021.3.4f1 Personal <DX11>
File Edit Assets GameObject Component Window Help
NS
Layers
Default
Hierarchy
All
SampleScene
Main Camera
Directional Light
Scene
Game
2D
Persp
Inspector
Project
Console
16
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Packages
Code Coverage
Custom NUnit
Editor Coroutines
JetBrains Rider Editor
Newtonsoft Json
Profile Analyzer
Services Core
Settings Manager
Assets
Scenes

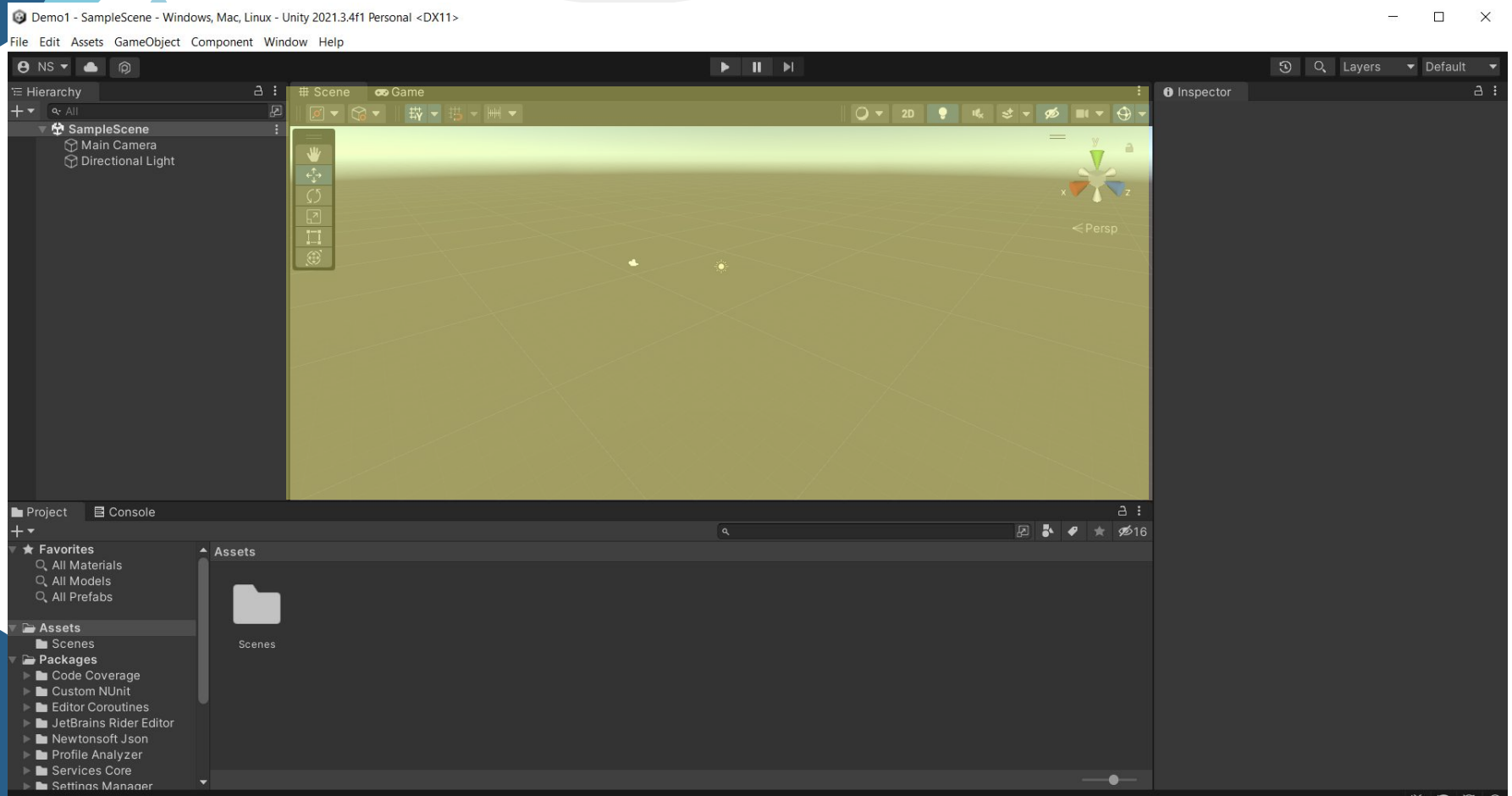
Objects In The Scene



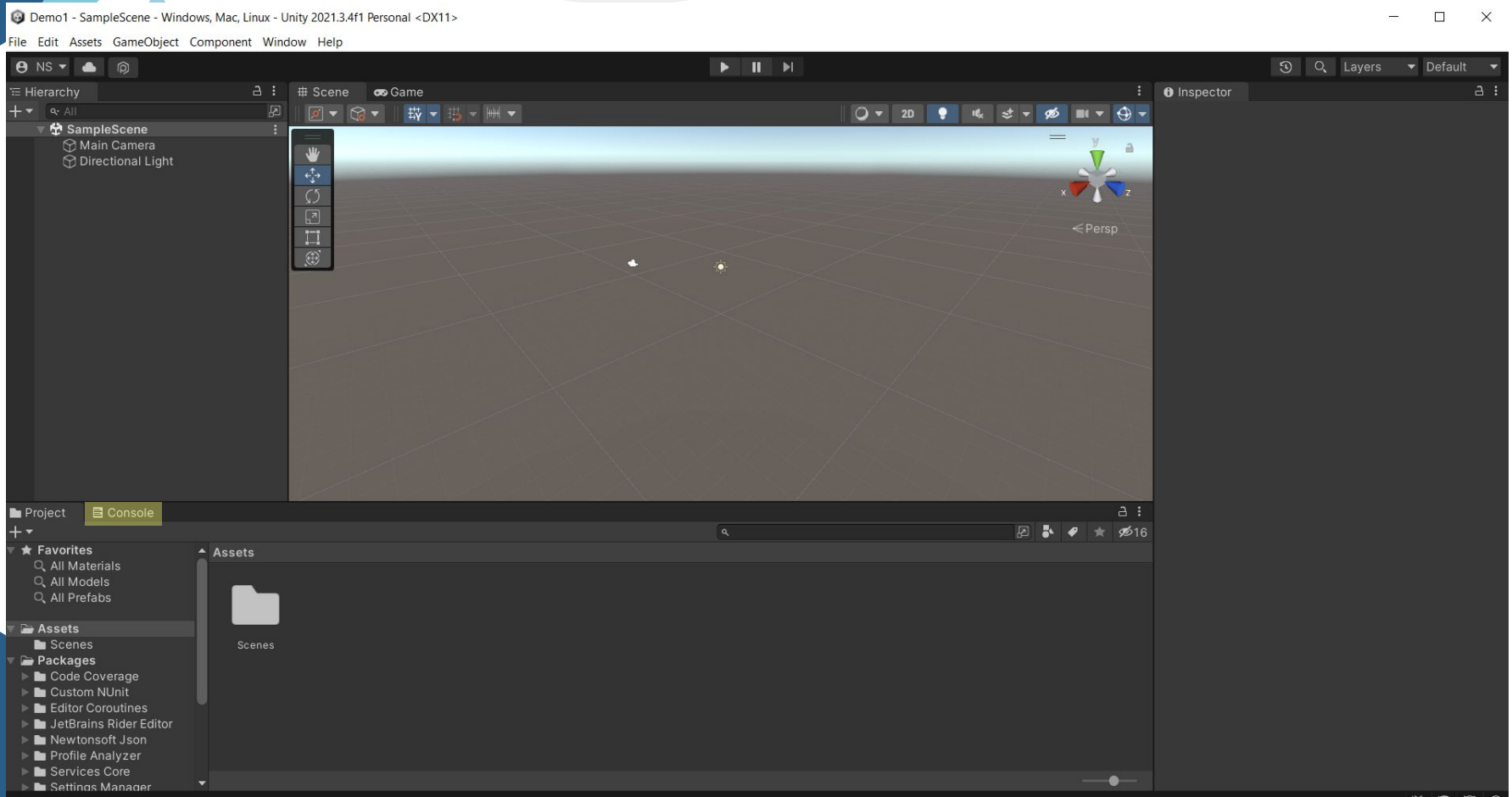
Object Properties



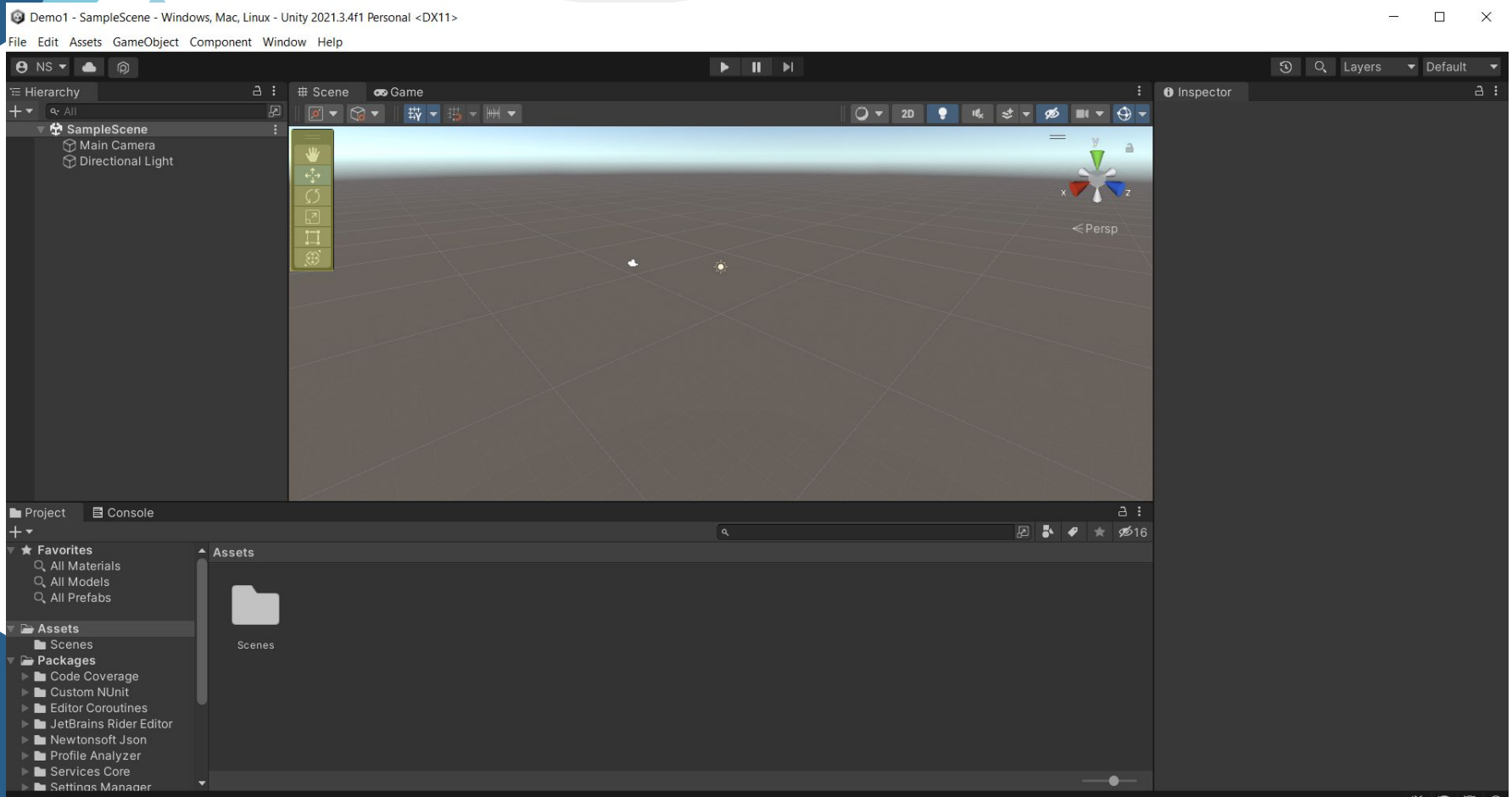
The Visual Display of the Game



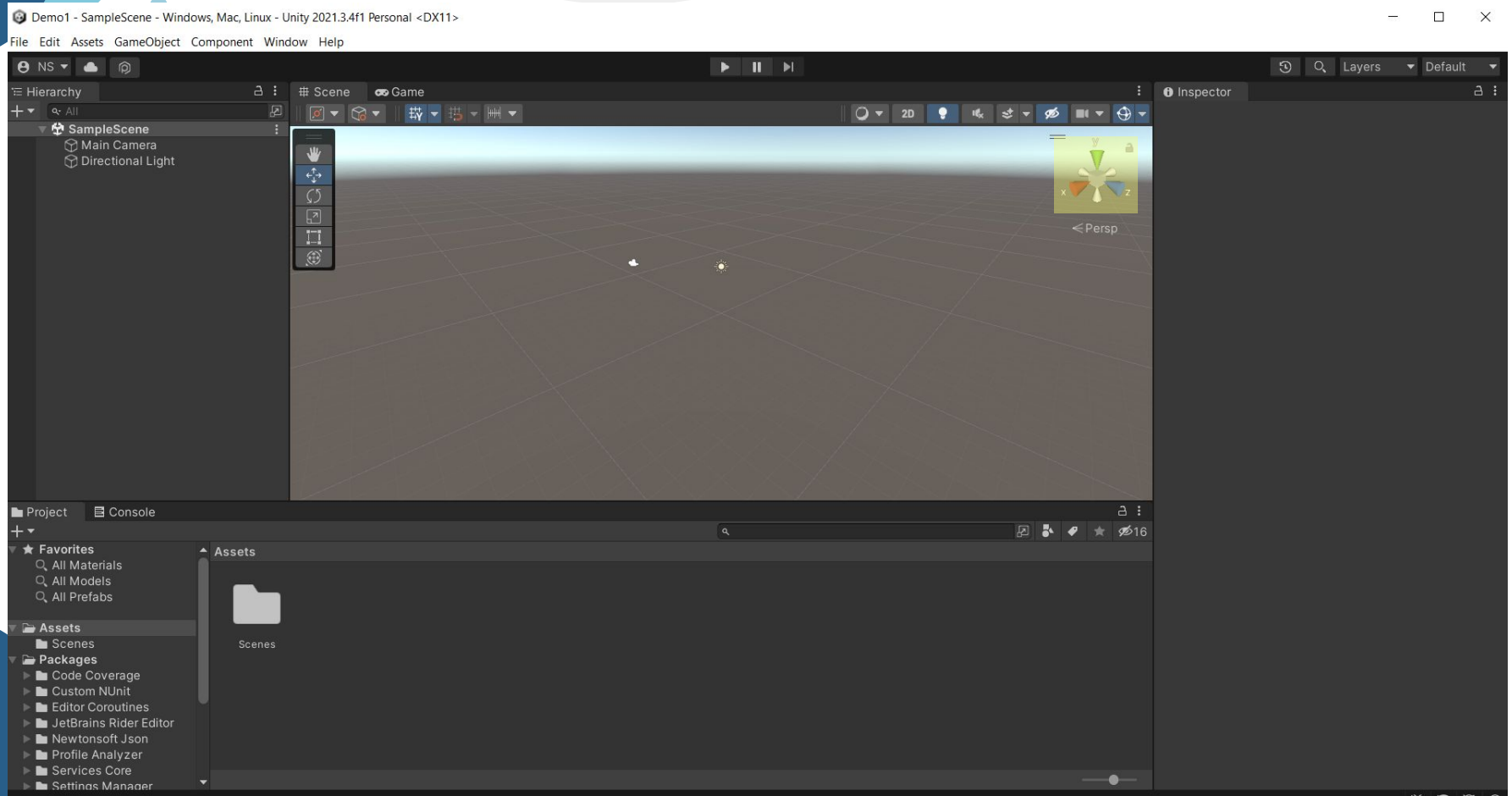
Where Error Messages or Print Messages Are



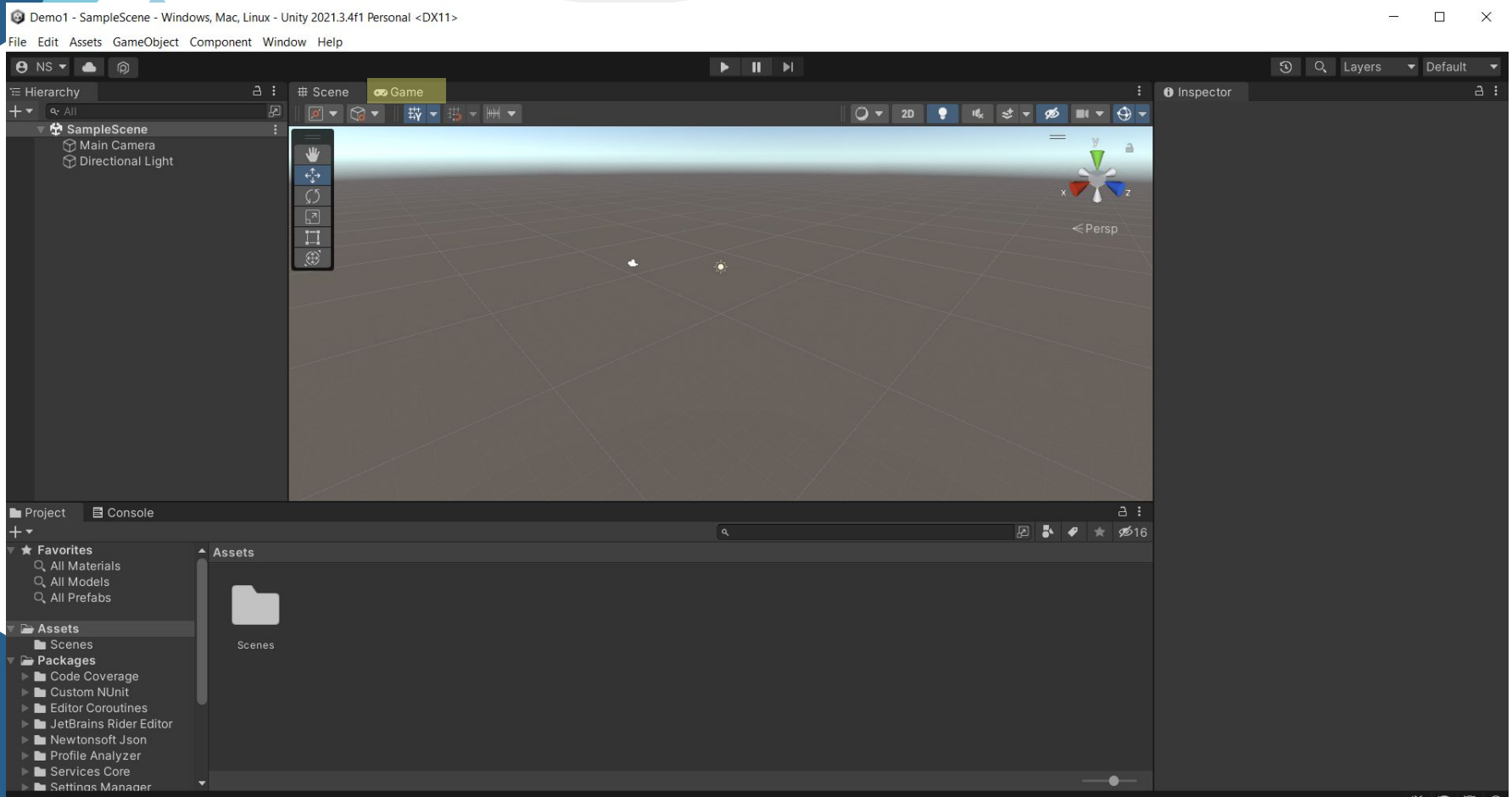
User-Friendly Tools to Navigate the Scene View



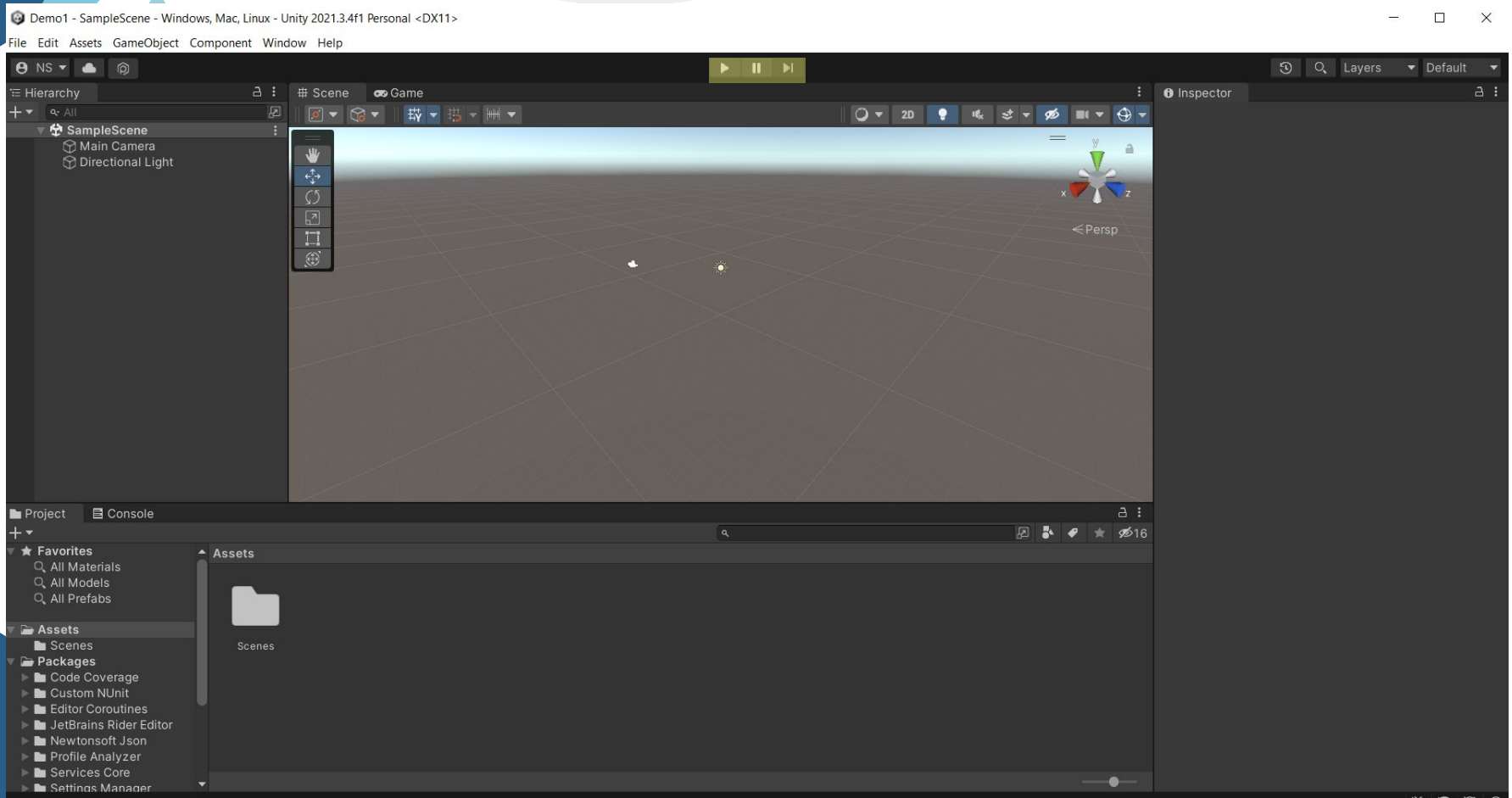
Angle Of View



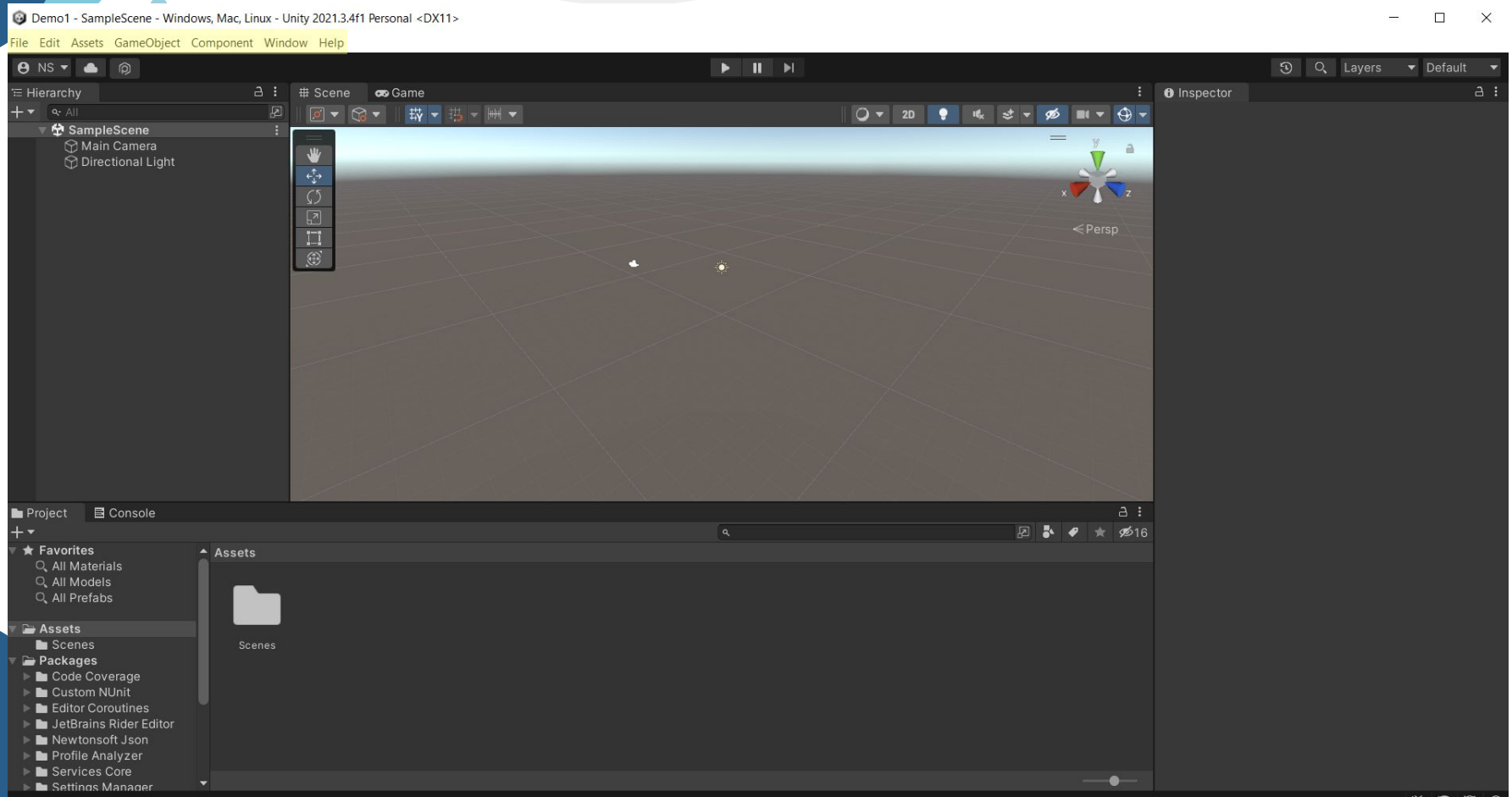
View The Game From The Player's Eyes



Run and Pause the Game



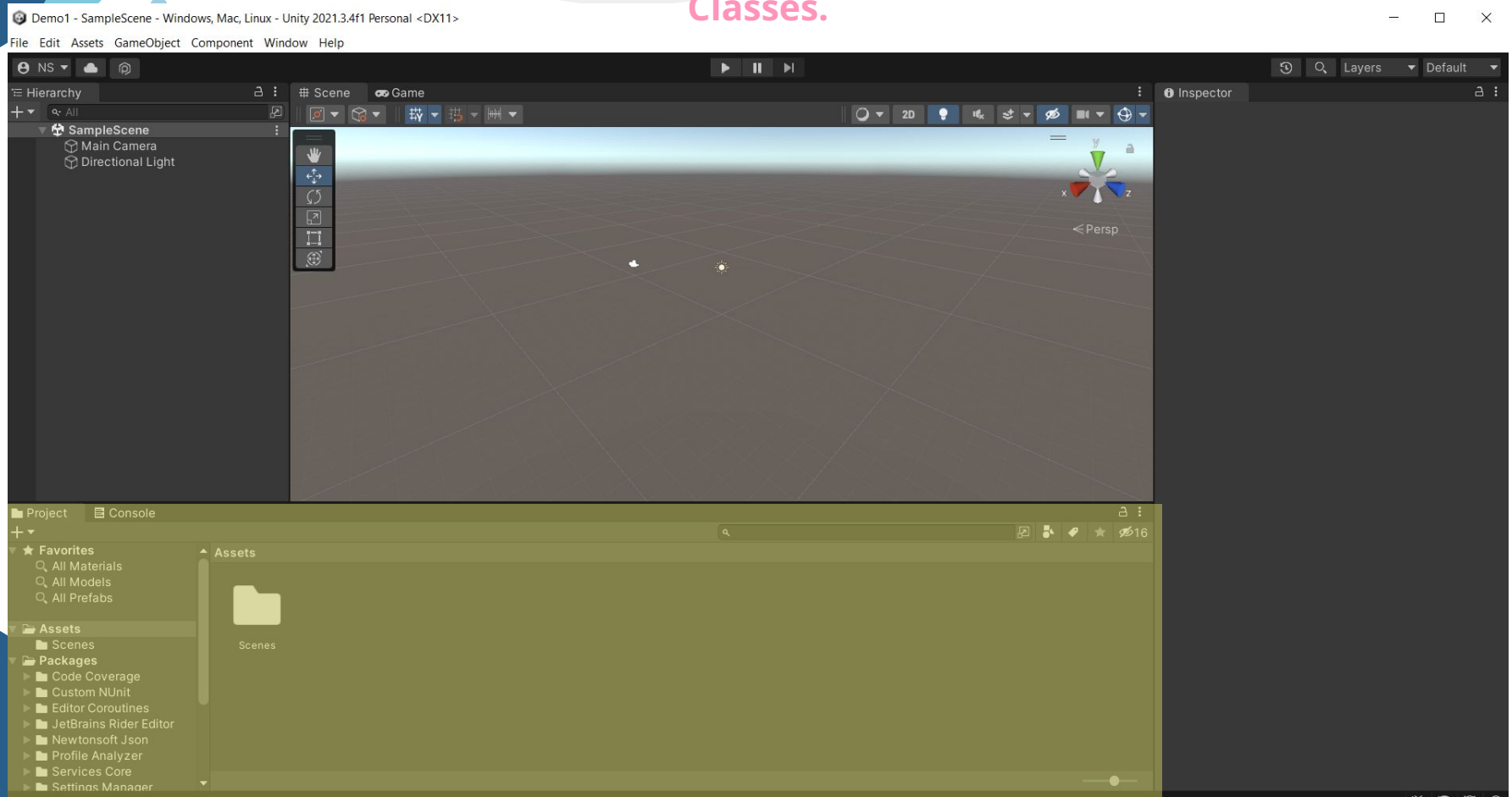
Extra Features



What’s Most Important to Know?

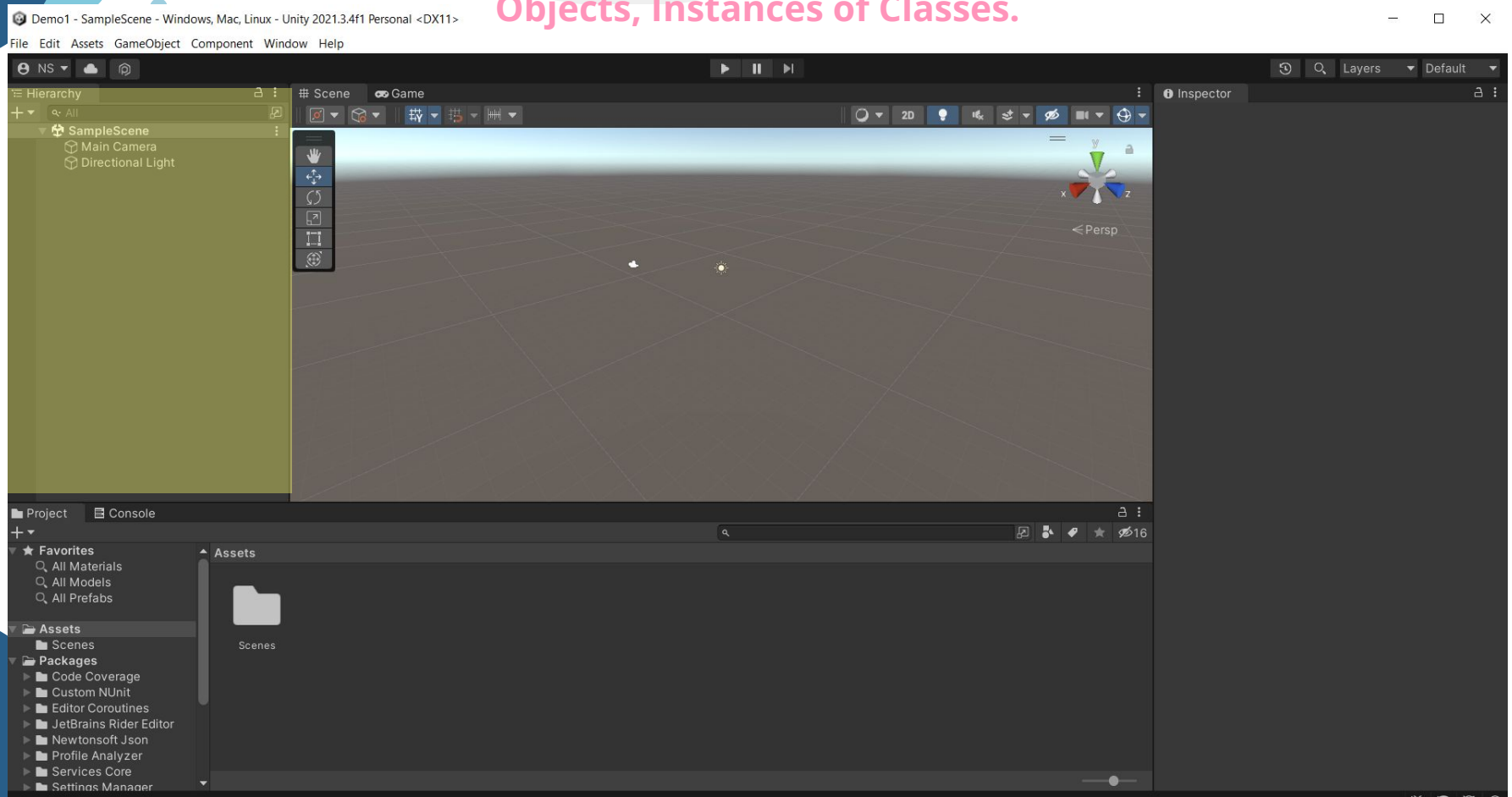
Always Exists In Every Scene.

Classes.



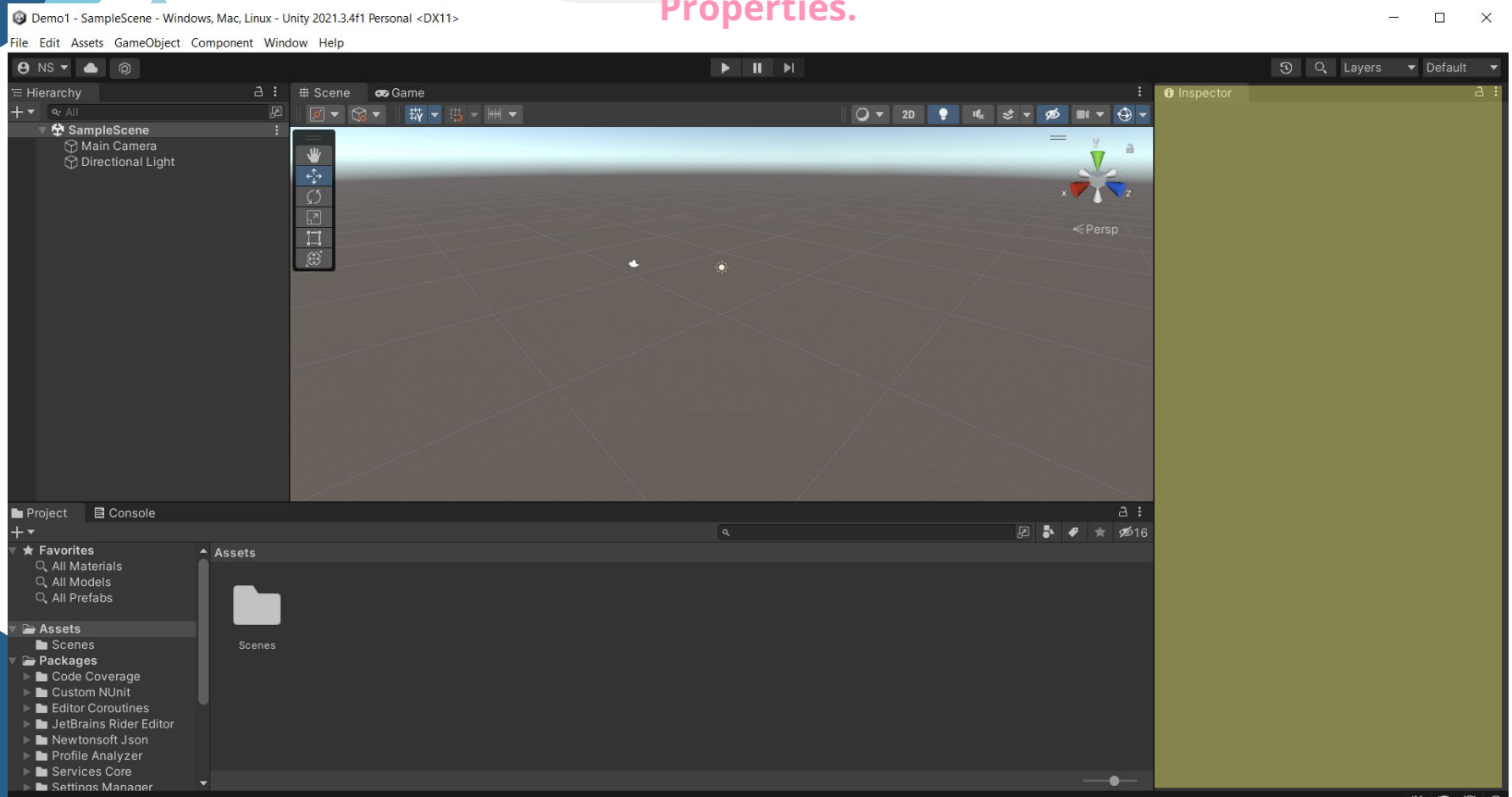
Only Exists In The Scene.

Objects, Instances of Classes.



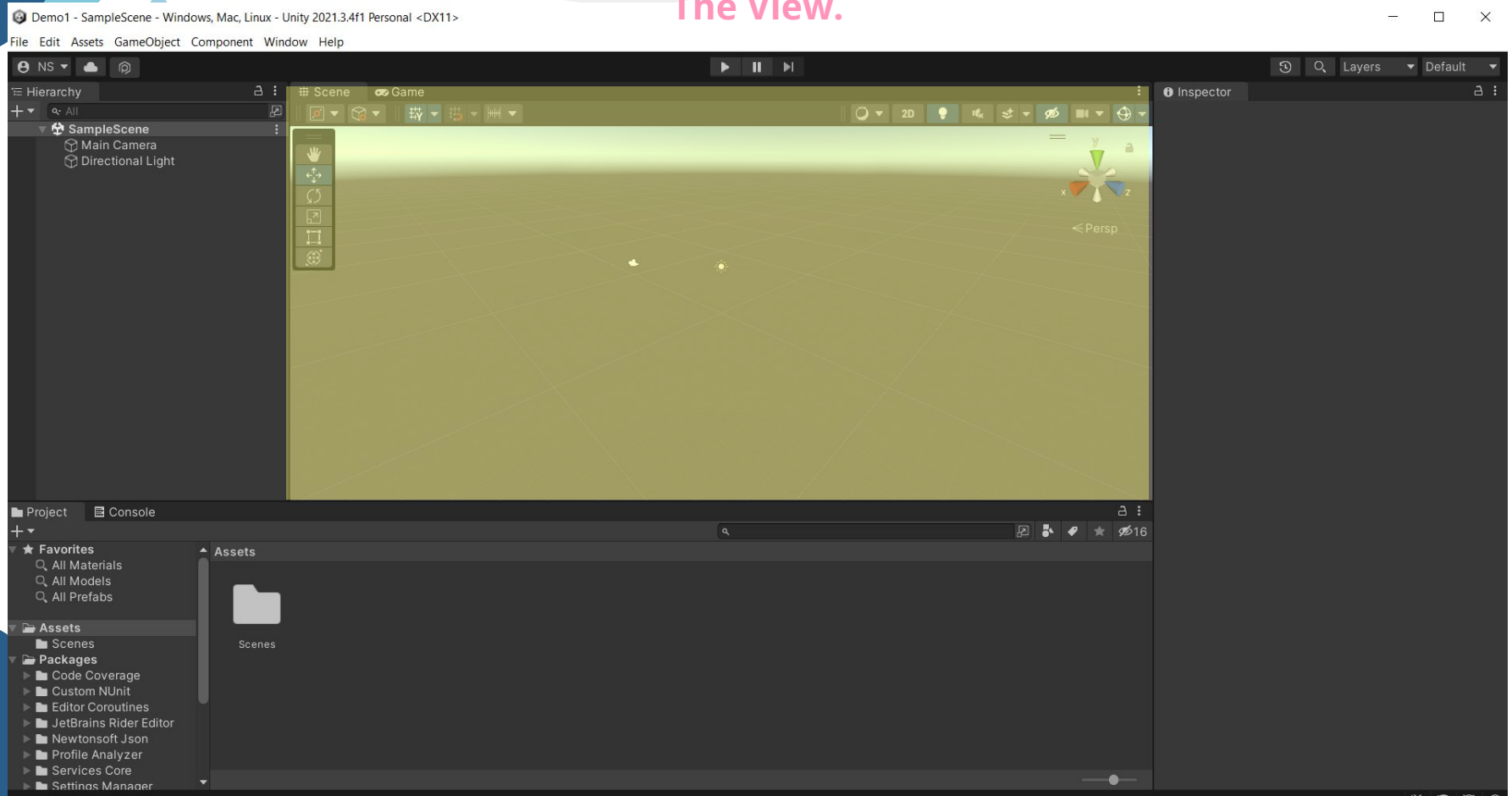
Add Components To Objects or Classes.

Properties.



Modify the Scene's Layout.

The View.



DEMO

The Languages of Unity

C#

JavaScript

The Languages of Unity

C#

JavaScript

DEMO

WIDE WALLS
Be Creative!

WIDE WALLS
Be Creative!

Example: Creating Gravity

RIGIDBODY


Rigidbody
Mass
1
Drag
0
Angular Drag
0.05
Use Gravity
Is Kinematic
Interpolate
None
Collision Detection
Discrete
Constraints
Info

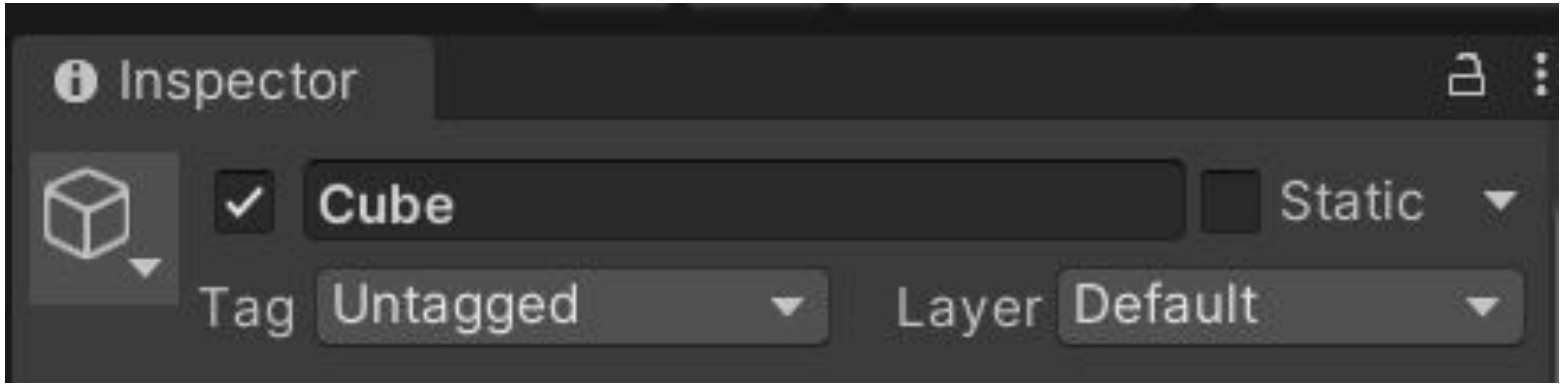
Physics

x-component	y-component
$v_x = v_{xo} + a_x t$	$v_y = v_{yo} + a_y t$
$x = \frac{1}{2}\left(v_{xo} + v_x\right) t$	$y = \frac{1}{2}\left(v_{yo} + v_y\right) t$
$v_x^2 = v_{xo}^2 + 2a_x x$	$v_y^2 = v_{yo}^2 + 2a_y y$
$\Delta x = v_{xo} t + \frac{1}{2} a_x t^2$	$\Delta y = v_{yo} t + \frac{1}{2} a_y t^2$

Physics

x-component	y-component
$v_x = v_{xo} + a_x t$	$v_y = v_{yo} + a_y t$
$x = \frac{1}{2}\left(v_{xo} + v_x\right) t$	$y = \frac{1}{2}\left(v_{yo} + v_y\right) t$
$v_x^2 = v_{xo}^2 + 2a_x x$	$v_y^2 = v_{yo}^2 + 2a_y y$
$\Delta x = v_{xo} t + \frac{1}{2} a_x t^2$	$\Delta y = v_{yo} t + \frac{1}{2} a_y t^2$

Tags and Layers



Can we do more?

Can we do more?

JUMPING

Summary of Key Aspects

Getting objects with Tags

```
GameObject.FindGameObjectsWithTag("Ground");
```

User Inputs

```
if (Input.GetKey("w"))
{
```

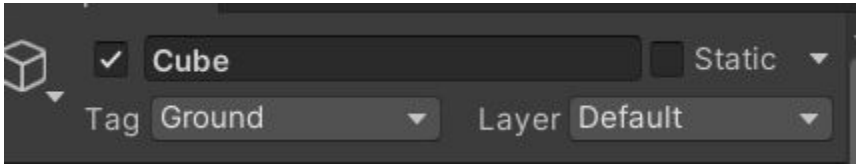
Components
Box Colliders

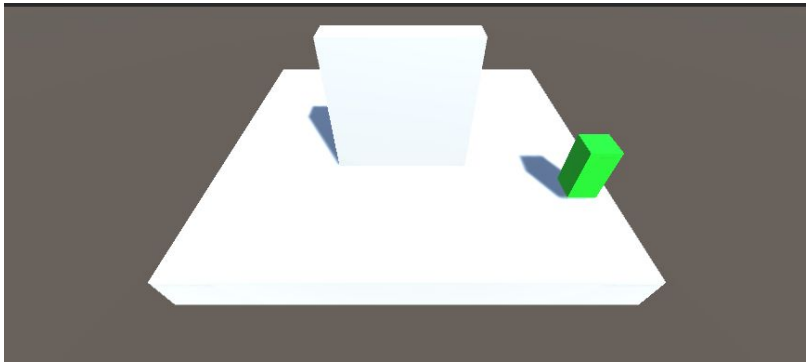
```
GetComponent<BoxCollider>().bounds.extents.y
```

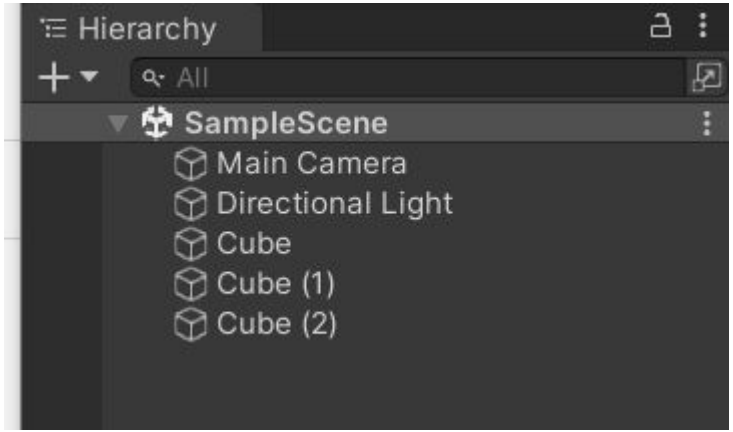
Physics

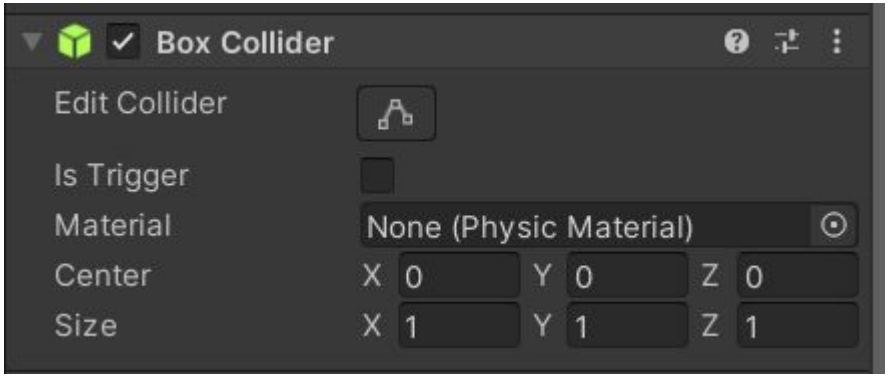
```
float ypos = startY + startVeloY * t - (1f / 2f) * accel_due_to_gravity * t * t;
```

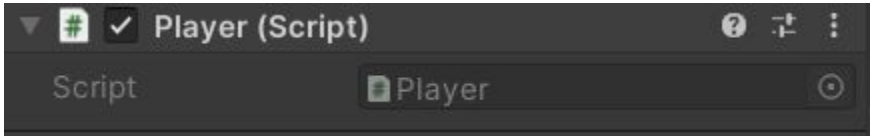
Summary of Key Aspects









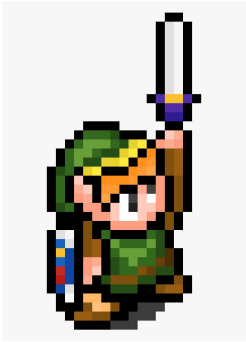


HIGH CEILING
Knowing the Details

What more would we want in an official game?

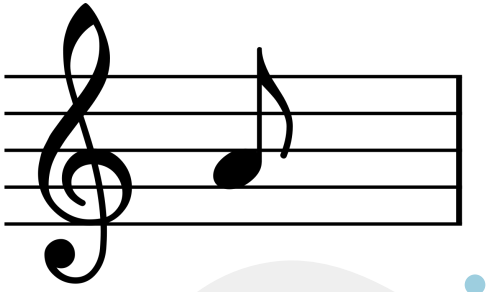
What more would we want in an official game?

Art!



Animation

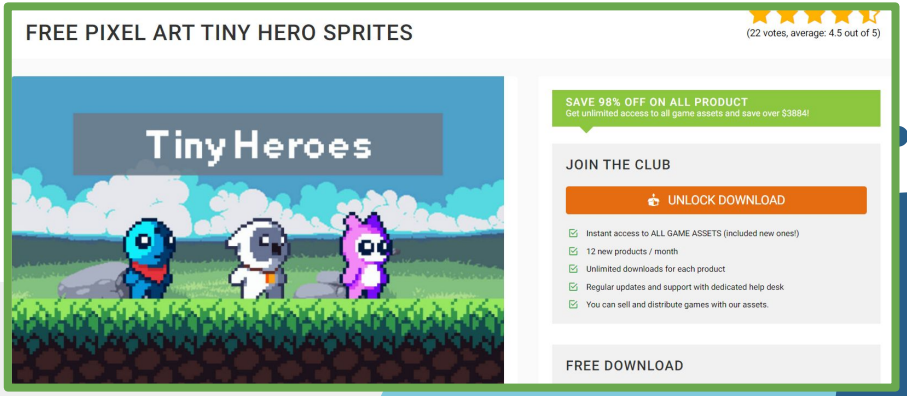
Music



Step 1: Draw! (Or use open source sprites to download)

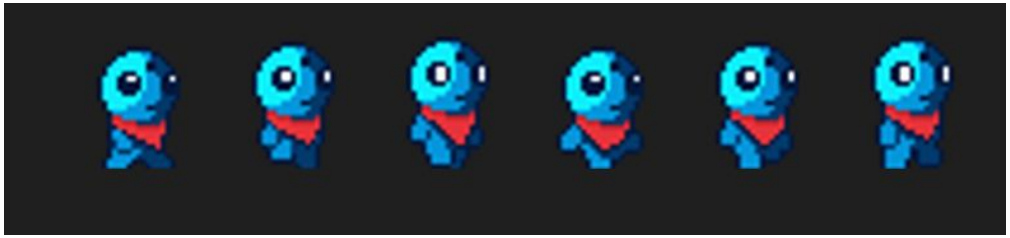
I have no drawing talent, so

And this one looks nice!

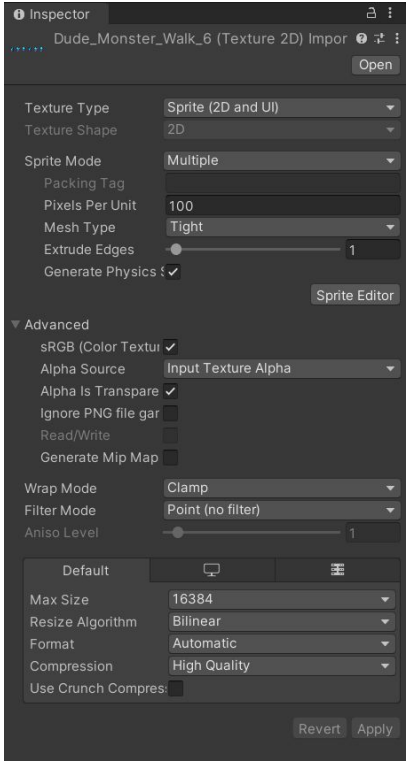


Step 2: Sprite Sheet

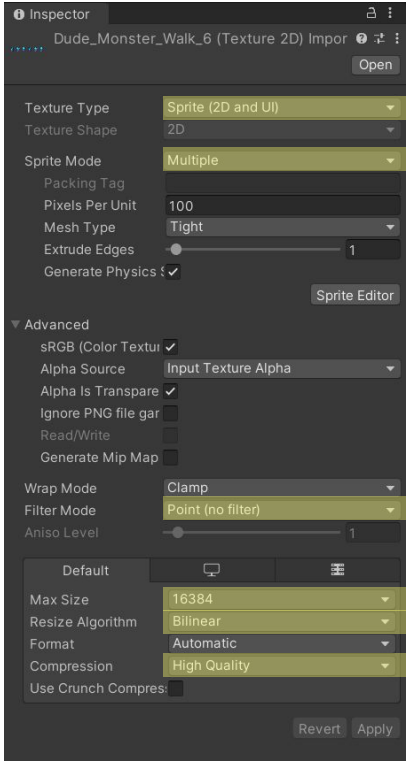
Walking Sprite Sheet



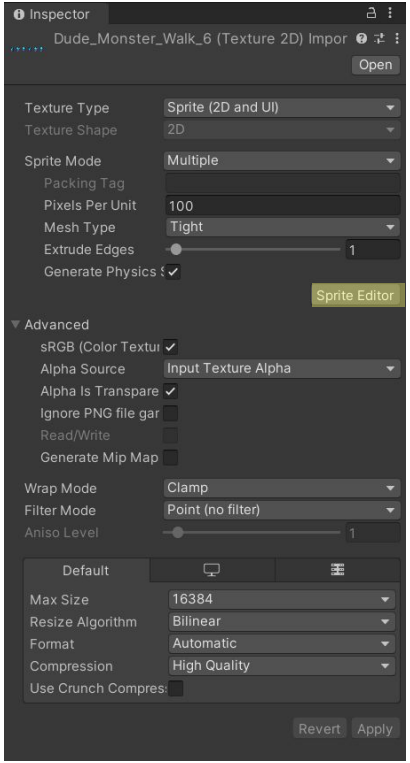
Step 3: Unity Splice



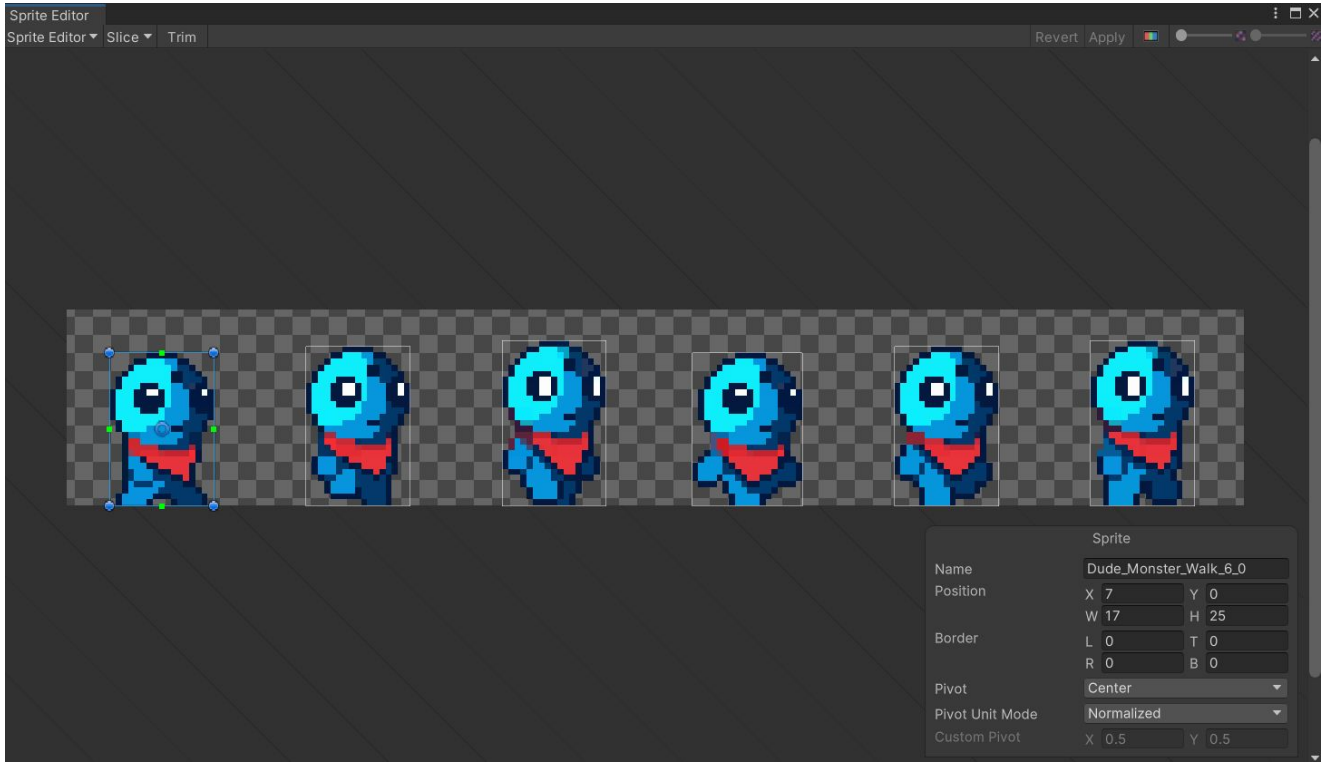
Step 3: Unity Splice



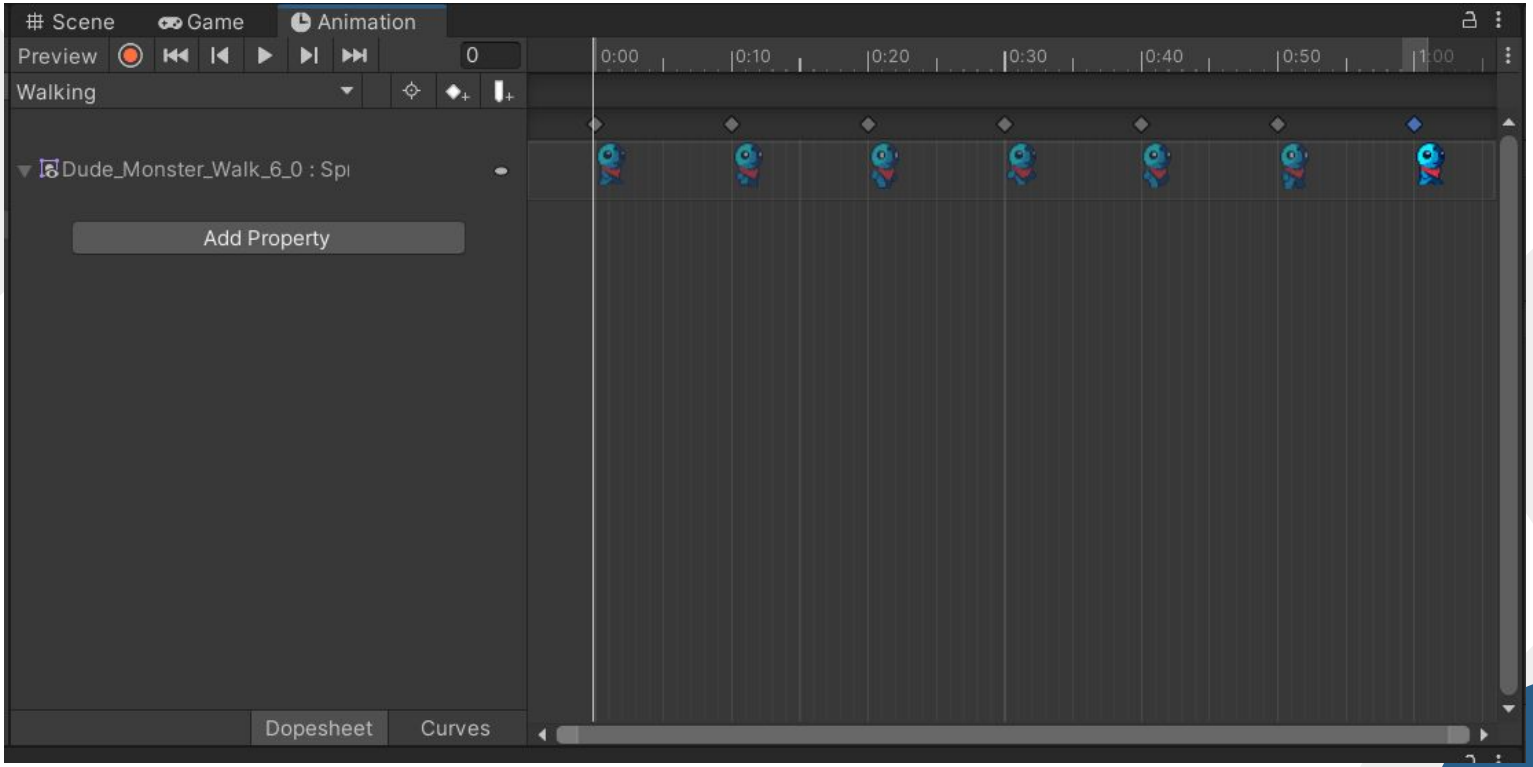
Step 3: Unity Splice



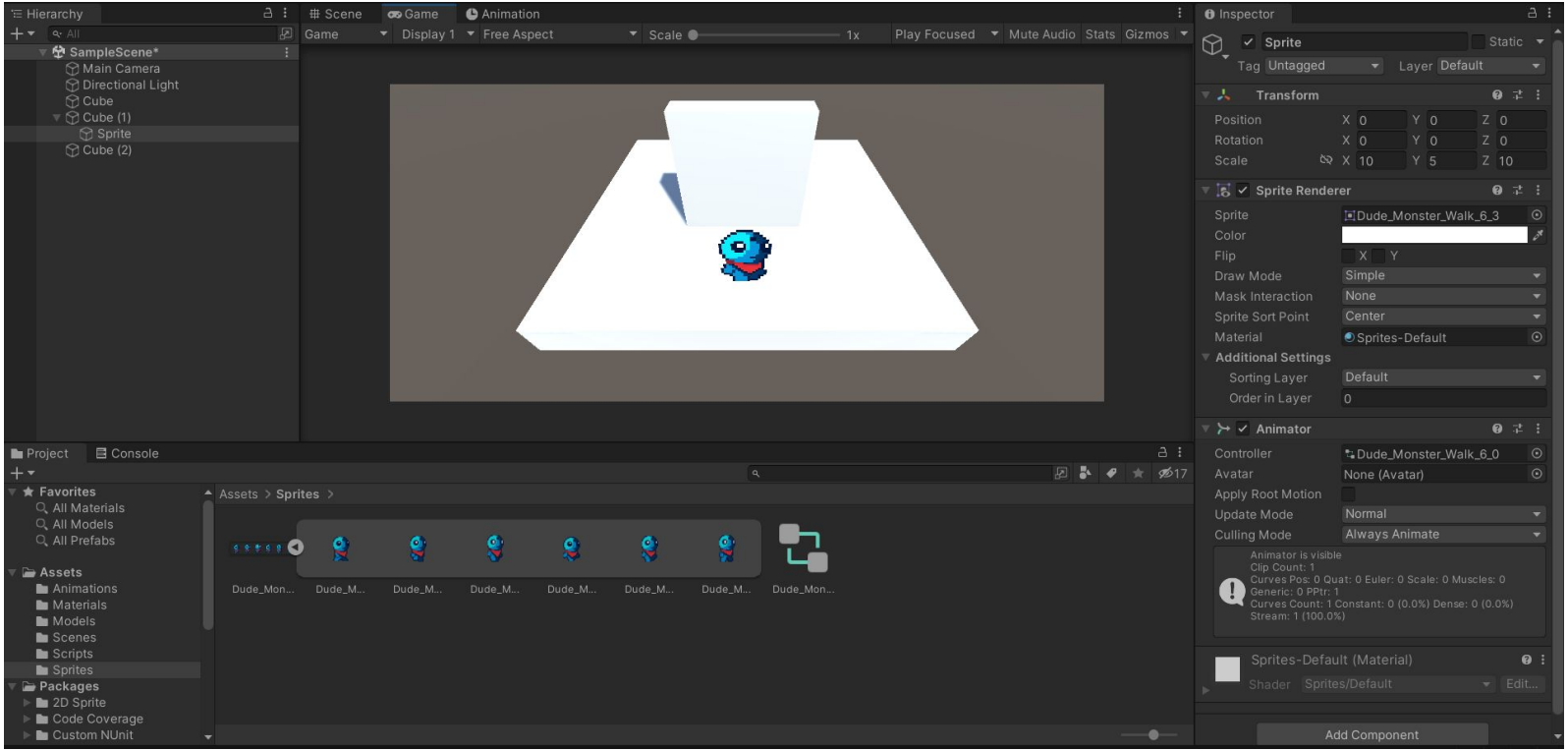
Step 3: Unity Splice



Step 4: Animation



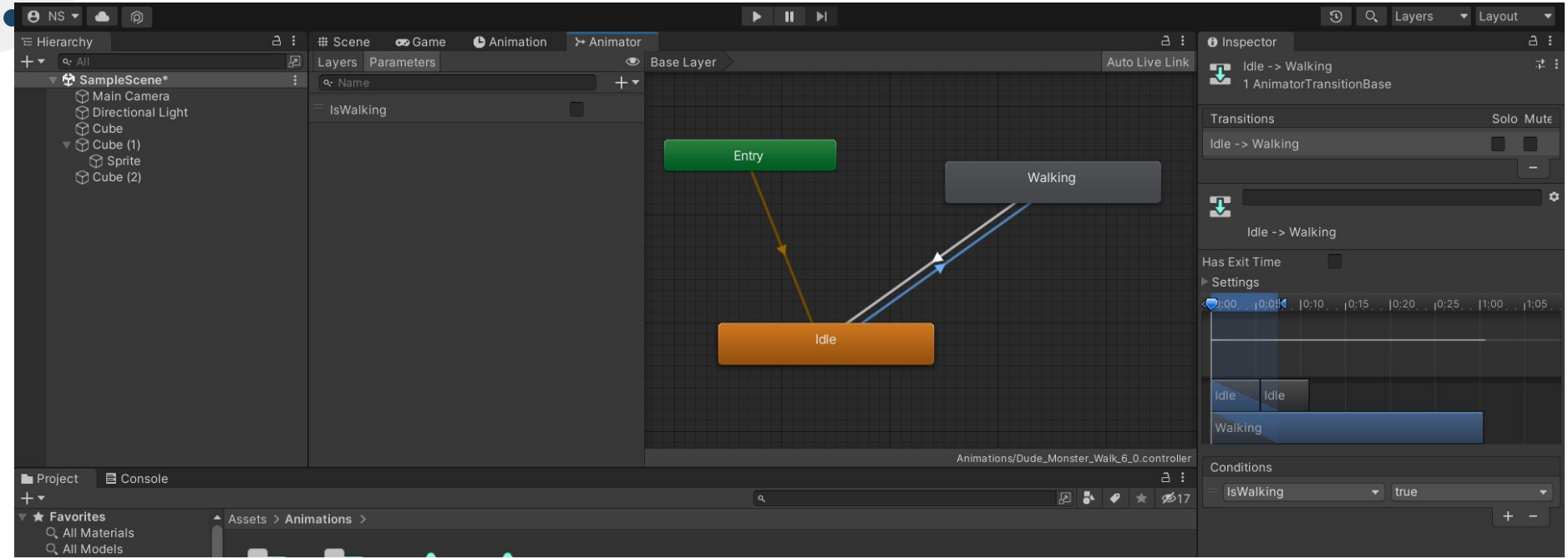
Step 5: Putting it into the scene



Hoorayyy

But how do we make it animate *only* when we move?

Animator!




```
playerAnimator.SetBool("IsWalking", true);
```

With Flips

```
if (Input.GetKey("a"))
{
    transform.position -= transform.right * speed;
    playerAnimator.SetBool("IsWalking", true);
    transform.localScale = new Vector3(-1 * Mathf.Abs(transform.localScale.x), transform.localScale.y, transform.localScale.z);
}
```

Unity

Summary

- We learned:
  - Game Engines
    - Unity
      - Low Floor, Wide Walls, High Ceiling
- We made:
  - a 3D Game
  - Collisions
  - Physics
  - Sprites
  - Animation

Suggested Strategy:

1. **Open Unity**
2. **Make a Scene**
3. **Make empty folders for**
    a. **Prefabs**
    b. **Scripts**
    c. **Materials**
    d. **Animations**
4. **Drag prefabs onto the scene to make Objects**
5. **Make Scripts for your objects**
6. **Code the player physics and collisions**
7. **Add art and animation!**

THANK YOU