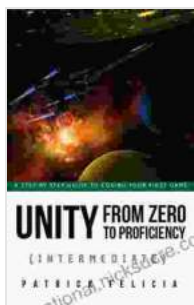


Step by Step Guide to Coding Your First FPS in Unity

Unity is a popular game engine that is used to create a wide variety of games, from 2D mobile games to 3D console games. In this guide, we will show you how to code your first first-person shooter (FPS) game in Unity.



Unity From Zero to Proficiency (Intermediate): A step-by-step guide to coding your first FPS in C# with Unity.

by Patrick Felicia

★★★★☆ 4.3 out of 5

Language : English
File size : 63774 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Word Wise : Enabled
Print length : 525 pages
Lending : Enabled



This guide is designed for beginners who have no prior experience with Unity or game development. We will start by setting up a new Unity project and then walk you through the process of adding gameplay mechanics like shooting, movement, and AI.

Setting Up a New Unity Project

To start, you will need to download and install Unity from the Unity website. Once you have installed Unity, launch the application and click on the "New

Project" button.

In the "New Project" window, enter a name for your project and select a location to save it. Then, click on the "Create Project" button.

Once your project has been created, you will see the Unity editor interface. The editor interface is divided into several different panes, including the Scene view, the Game view, and the Inspector view.

To create a new scene, click on the "File" menu and select "New Scene".

Creating the Player

The first step in creating an FPS game is to create the player character. To do this, click on the "GameObject" menu and select "3D Object" > "Cube".

The cube will appear in the Scene view. This will be the player character.

Next, we need to add a camera to the player character. To do this, click on the "GameObject" menu and select "Camera".

The camera will appear in the Scene view. Position the camera so that it is looking out from the player character's head.

Now, we need to add a script to the player character that will control its movement. To do this, click on the "Component" menu and select "New Script".

Save the script as "PlayerController".

Open the PlayerController script and paste the following code:

```
c# using UnityEngine;
```

```
public class PlayerController : MonoBehaviour { public float moveSpeed =  
5.0f; public float jumpForce = 5.0f;
```

```
private Rigidbody rb;
```

```
void Start(){rb = GetComponent(); }
```

```
void FixedUpdate(){float moveHorizontal = Input.GetAxis("Horizontal"); float  
moveVertical = Input.GetAxis("Vertical");
```

```
Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);
```

```
rb.AddForce(movement * moveSpeed);
```

```
if (Input.GetKeyDown(KeyCode.Space)){rb.AddForce(Vector3.up *  
jumpForce, ForceMode.Impulse); }}
```

This script will allow the player character to move around the scene using the WASD keys and jump using the spacebar.

Creating the Gun

The next step is to create the gun. To do this, click on the "GameObject" menu and select "3D Object" > "Cube".

The cube will appear in the Scene view. This will be the gun.

Next, we need to add a script to the gun that will control its shooting. To do this, click on the "Component" menu and select "New Script".

Save the script as "GunController".

Open the GunController script and paste the following code:

```
c# using UnityEngine;

public class GunController : MonoBehaviour { public GameObject
bulletPrefab; public float bulletSpeed = 10.0f;

void Update(){if (Input.GetMouseButtonDown(0)){GameObject bullet =
Instantiate(bulletPrefab, transform.position, transform.rotation);
bullet.GetComponent().AddForce(transform.forward * bulletSpeed,
ForceMode.Impulse); }}}}
```

This script will allow the player to shoot bullets by clicking the left mouse button.

Creating the Bullets

The next step is to create the bullets. To do this, click on the "GameObject" menu and select "3D Object" > "Sphere".

The sphere will appear in the Scene view. This will be the bullet.

Next, we need to add a script to the bullet that will control its movement. To do this, click on the "Component" menu and select "New Script".

Save the script as "BulletController".

Open the BulletController script and paste the following code:

```
c# using UnityEngine;
```

```
public class BulletController : MonoBehaviour { public float speed = 10.0f;
```

```
void FixedUpdate(){transform.Translate(Vector3.forward * speed *  
Time.deltaTime); }
```

```
void OnCollisionEnter(Collision collision){if (collision.gameObject.tag  
=="Enemy"){Destroy(collision.gameObject); Destroy(gameObject); }}
```

This script will make the bullet move forward at a constant speed and destroy itself when it collides with an enemy.

Creating the Enemies

The next step is to create the enemies. To do this, click on the "GameObject" menu and select "3D Object" > "Cube".

The cube will appear in the Scene view. This will be the enemy.

Next, we need to add a script to the enemy that will control its AI. To do this, click on the "Component" menu and select "New Script".

Save the script as "EnemyController".

Open the EnemyController script and paste the following code:

```
c# using UnityEngine; using UnityEngine.AI;
```

```
public class EnemyController : MonoBehaviour { public Transform player;  
public float moveSpeed = 5.0f;
```

```
private NavMeshAgent navAgent;
```

```
void Start(){navAgent = GetComponent(); }
```

```
void Update(){navAgent.SetDestination(player.position); }}
```

This script will make the enemy move towards the player at a constant speed.

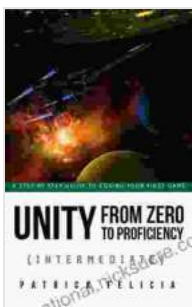
Testing the Game

Now that we have created all of the game objects and scripts, we can test the game. To do this, click on the "Play" button in the editor toolbar.

The game will start running and you will be able to control the player character using the WASD keys and jump using the spacebar. You can shoot bullets by clicking the left mouse button.

The enemies will move towards the player and try to attack them. You can destroy the enemies by shooting them with bullets.

This guide has shown you how to code your first FPS game in Unity. You have learned how to create



Unity From Zero to Proficiency (Intermediate): A step-by-step guide to coding your first FPS in C# with Unity.

by Patrick Felicia

★★★★☆ 4.3 out of 5

Language : English

File size : 63774 KB

Text-to-Speech : Enabled

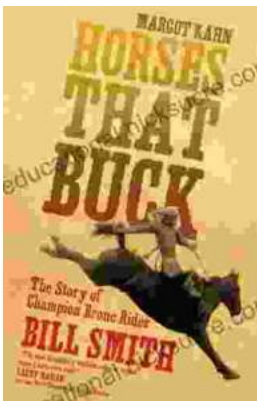
Screen Reader : Supported

Enhanced typesetting : Enabled

Word Wise : Enabled
Print length : 525 pages
Lending : Enabled

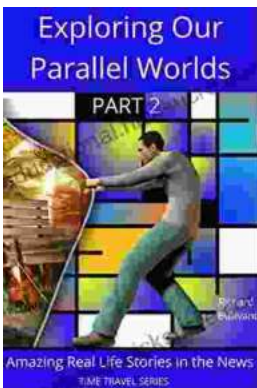
FREE

DOWNLOAD E-BOOK



The Story of Champion Bronc Rider Bill Smith: A Legacy of Grit and Glory in the Wild West

In the annals of rodeo history, the name Bill Smith stands tall as one of the most celebrated bronc riders of all time. His extraordinary skill, unwavering...



Amazing Real Life Stories In The News

The news is often filled with stories of tragedy and despair, but there are also countless stories of hope, resilience, and heroism. Here are just a...