

Lesson 3

Game Interaction and Collectables

Collectathon Games



1996



2003



2017

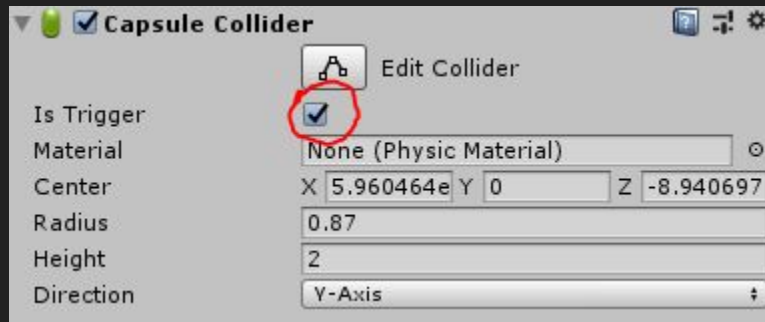


2017

Today's Topics

1. Use colliders to detect when we collect a collectable.
2. Use a game manager singleton to track progress.
3. Unlock door after collection number is reached.
4. Scene switching and data persistence.

1. Detecting Collection



Not a Trigger:

- Does interact with physics
- Uses OnCollision
- Usually tight to the model

Trigger:

- Does not interact with physics
- Uses OnTrigger
- Usually bigger than the model

Collectable.cs

```
private void OnTriggerEnter(Collider other)
{
    Debug.Log("Collected!");
    Destroy(this.gameObject);
    // Using "this.gameObject" and not just "this" is important!
}
```

2. Game Manager

The Singleton Pattern.

GameController.cs

```
private static GameController _instance;
```

```
void Awake()
```

```
{
```

```
    _instance = this;
```

```
}
```


GameController.cs

```
private static int coins = 0;
```

```
public static void CollectCoin()
```

```
{
```

```
    coins = coins + 1;
```

```
    Debug.Log("Collected coin!");
```

```
}
```

GameController.cs

```
public GameObject player;
```

```
public static GameObject GetPlayer()
```

```
{
```

```
    return _instance.player;
```

```
}
```

Back in Collectable.cs

Collectable.cs

```
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject == GameController.GetPlayer())
    {
        GameController.CollectCoin();
        Destroy(this.gameObject);
    }
}
```

3. Conditional Doors



GameController.cs

```
public static int GetCoins()  
{  
    return coins;  
}
```

Door.cs

```
private const int RequiredCoins = 2;
```

```
private void OnTriggerEnter(Collider other)
```

```
{
```

```
    if (other.gameObject == GameController.GetPlayer() &&  
GameController.GetCoins() >= RequiredCoins)
```

```
    {
```

```
        Destroy(this.gameObject);
```

```
    }
```

```
}
```

4. Scene Switching

GameController.cs

```
void Awake()  
{  
    _instance = this;  
    DontDestroyOnLoad(this.gameObject);  
}
```

DontDestroyOnLoad prevents Unity from deleting that object when a scene is changed.

Door.cs

```
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject == GameController.GetPlayer() && GameController.GetCoins() >=
    RequiredCoins)
    {
        // Destroy(this.gameObject);
        SceneManager.LoadScene("Scene2");
    }
}
```

DontDestroyOnLoad prevents Unity from deleting that object when a scene is changed.

