

COMP3752 Computer Game Development

Patterns/Principles/Team vs Team Combat

Peter Mitchell

Topics Today

- Overview of the 3D Character Controller from last week.
- Coroutines
- ScriptableObjects
- Game Design Patterns
- SOLID Principles and Unity
- Team vs Team Combat (based on Star Wars: Galaxy of Heroes)
- Designing a Level Editor

Coroutines

- A way to synchronously run methods that do not need to finish right away.

```
private IEnumerator movePlaces()
{
    for(int i = 0; i < 10; i++)
    {
        Vector3 modifiedPosition = new Vector3(
            Random.Range(origin.x - 4, origin.x + 4),
            Random.Range(origin.y - 4, origin.y + 4),
            Random.Range(origin.z - 4, origin.z + 4));
        transform.position = modifiedPosition;
        yield return new WaitForSeconds(1f);
    }
    transform.position = origin;
}
```

Coroutines

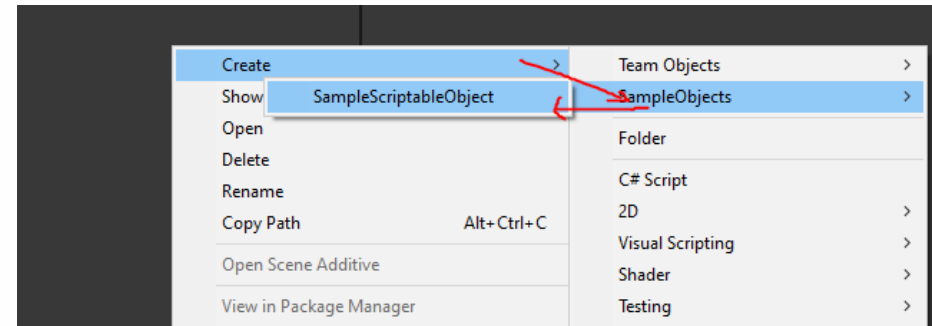
```
void Update()  
{  
    if(Input.GetKeyDown(KeyCode.Space))  
    {  
        StartCoroutine(movePlaces());  
    }  
}
```

ScriptableObjects

- Provide a way to define data independent of objects that can be hot swapped into objects with defining objects in the inspector.

```
[CreateAssetMenu(fileName = "SampleScriptableObject", menuName =  
"SampleObjects/SampleScriptableObject", order = 1)]
```

```
public class SampleScriptableObject : ScriptableObject  
{  
    public string someStringData;  
    public int someIntValue;  
    public Vector3[] someListOfPoints;  
}
```



Game Design Patterns

- Singleton Pattern
- Observer Pattern
- Command Pattern
- Component Pattern
- Flyweight Pattern
- State Pattern

SOLID Principles in Unity

- [S]ingle Responsibility Principle
“Every class should have one responsibility.”
- [O]pen Closed Principle
“Open for extension, but closed for modification.”
- [L]iskov Substitution Principle
“Functions that use pointers or references to base classes must be able to use objects of derived classes without knowing it.”
- [I]nterface Segregation Principle
“Many client-specific interfaces are better than one general-purpose interface.”
- [D]ependency Inversion Principle
“Depend upon abstractions, not concretions.”

Team vs Team



Designing a Level Editor

- If there is time I will discuss how the level editor integrates into Chad's Challenge.

Any Questions?

- Further reading with references to a lot of content I talked about can be found in the PDF version to go with this talk.
- Welcome to send me a message or ask during the Wednesday session if you have any questions about the content. Time permitting, I may give suggestions or suggest sources to look at.