COMP140-GAM160: Game Programming

9: Events and Decoupling

Learning outcomes

- **Understand** the Static and Singletons
- **Apply** decoupling strategies to your own code base

Static Keyword & Singletons

Static Keyword

- The **Static** keyword has multiple meanings in modern programming languages
  - If you mark a variable **inside a function** as static then it is allocated once for the whole lifetime of the programme
  - If you mark a variable **inside a class** as static then it is allocated once and the data is shared between all instances of the class (otherwise known as a class variable)
  - If you mark a function **inside a class** as static then it is shared amongst all instances of the class & can only operate on static variables

`https://softwareengineering.stackexchange.com/questions/163457/understanding-the-static-keyword`

Static Example - C++

```
class PlayerStats
{
private:
    float currentHealth;

    static int Score;
public:
    static void AddScore(int Amount)
    {
        //Can't do this will get a compiler warning, can't refrence a
        //non static variable in a static function
        //currentHealth += 2.0f

        //Can do the following because we are working with a static
            variable
        Score += Amount;
    };

    static int GetScore()
    {
        return  Score;
    };
public:
    static int Deaths;
};
```

Static Usage - C++

```
//Notice we use :: (scope operator) to access the variable
PlayerStats::Deaths = 0;

PlayerStats::Deaths++;

PlayerStats::AddScore(100);
int score = PlayerStats::GetScore();
```

Static Example - C#

```
public class PlayerStats : MonoBehaviour
{
    private static int Score;
    public static int Deaths;

    public static void AddScore(int Amount)
    {
        //Can do the following because we are working with a static variable
        Score += Amount;
    }

    public static int GetScore()
    {
        return Score;
    }
}
```

Static Usage - C#

```
PlayerStats.Deaths = 0;
PlayerStats.Deaths++;

PlayerStats.AddScore(10);
int score = PlayerStats.GetScore();
```

Static keyword - Use Cases

- Useful for managing Global Data such as score
- Creating Utility functions which don't require to be part of a class instance

Static Examples

- Unity - https://unity3d.com/learn/tutorials/topics/scripting/statics
- C++ - https://www.youtube.com/watch?v=zGPefqkwBK0

Singleton

- Guarantees that there is only one instance of a class and can be accessed globally
- Usually 'lazily' initialised via a static function that satisfy the statement above
- Used for manager classes which track some sort of Global State
- **Warning!** Some consider Singletons to be an anti-pattern
- Singleton: an anti-pattern? - `https://stackoverflow.com/questions/12755539/why-is-singleton-considered-an-anti-pattern`

Unity Implementations

- Singleton - `https://unity3d.com/learn/tutorials/projects/2d-roguelike-tutorial/writing-game-manager`
- Better Singleton? - `https://stackoverflow.com/documentation/unity3d/2137/singletons-in-unity`

C++ Implementations

- Singleton - `http://gameprogrammingpatterns.com/singleton.html`

Events

Observer

- When one object is updated, all observers of this object are notified
- A list of observers are maintained by the subject
- When the state of the subject changes then the list of the observers is processed
- Each observer is then notified of the change
- Each observer should register/unregistered itself with a subject
- Very useful for UI, Input or Network systems in games
- Some of this function is already built into C#(delegates & Events) and Unity(Unity Events)

Implementations

- Unity (event system) - `https://unity3d.com/learn/tutorials/topics/scripting/events-creating-simple-messaging-system`
- C++ (Observer Pattern) - `http://gameprogrammingpatterns.com/observer.html`

Coffee Break