Threading

1. Introduction

This tutorial introduces the benefits of using Systems in that they can be run on different threads/cores.

2. Setting up the Windows project

Download the accompanying file '600098-Mutithreading.zip' and unzip it into a convenient area on your file store. Double click on the solution file to start Visual Studio and load the solution file.

3. How it works: Program.cs

The **`Program`** class contains the Main method. This class creates a Scene and calls the Start() method of the Scene.

4. How it works: Scene.cs

The **`Scene`** class creates a SystemTest and a ThreadingManager that a single thread. It also contains the Start() method.

The Start() method starts a stopwatch, creates a callback delegate to a method called DoWork() that is inside of the SystemTest class. This method is what we want to put onto threads.

The Start() method then adds the DoWork() method to start 10 times in the ThreadingManager.

The Start() method then displays how many tasks (methods) are waiting in the queue until they are all completed. It then displays the time taken for all 10 DoWork() methods to finish.

5. How it works: SystemTest.cs

The **`SystemTest`** class creates the DoWork() method. This method does some calculations.

6. How it works: ThreadingManager.cs

The **`ThreadingManager`** class creates a threadpool for the number of threads specified by the Scene class.

It contains the Start() method that adds a method to the queue.

The other methods return useful information about the threadpool.

7. Exercises

1. Review the code until you understand what is happening.
2. Adjust the number of iterations (currently 999999999) inside of the DoWork() method of SystemTest until the tasks take around 20 seconds to complete.
3. Record how many seconds the tasks take using 1 thread.
4. In the Scene class, change the number of threads to 2.
   ```
   private ThreadingManager manager = new ThreadingManager(2);
   ```
5. Record how many seconds the tasks take using 2 threads.
6. Record how many seconds the tasks take using 5 threads.
7. Record how many seconds the tasks take using 10 threads.
8. Consider how you could add this to your game engine.