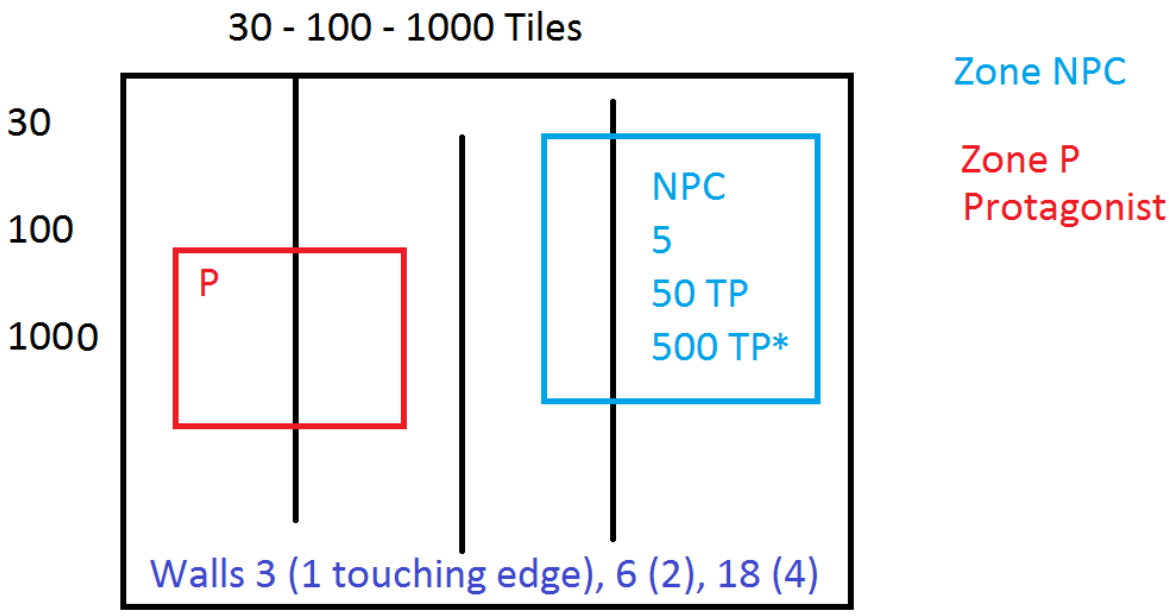
Multi-Threaded Visualisation Rubric (17th May 2021)

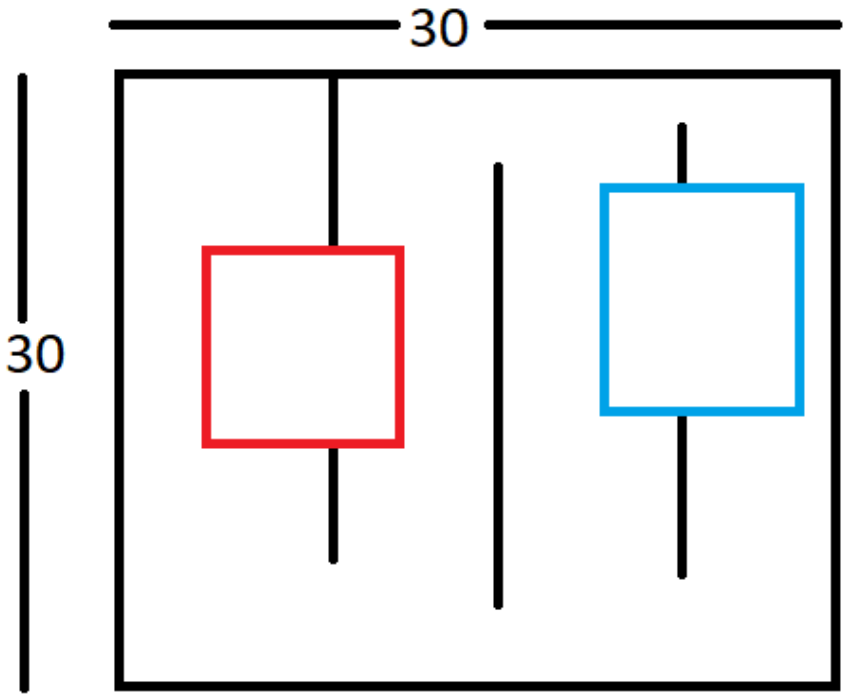
Demonstration of Project from 9.00am on that day, in roll order

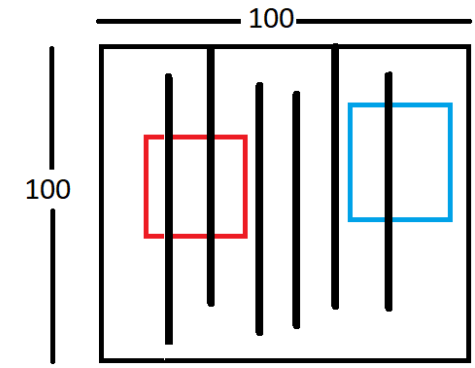
No Demonstration, no marks will be allocated.

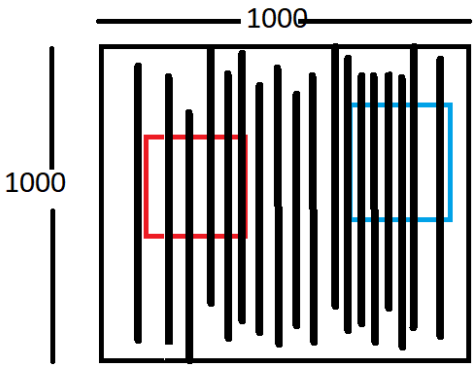


GameTicks: Slow, Normal, Fast (No Gameticks)

* Thread Pool







Project is worth 25%

Produced SFML AStar Ambush simulation which must include the following sub-systems:

- Input Management
  - Keyboard
- Primitive Shapes
  - Points
  - Lines
  - Rectangle
- Collision Detection (2D can use a library)
- NPC's that path find to the Player.
- Threading
  The introduction of threading must enhance performance (i.e increase FPS).
- SFML Threading of subsystem(s) can utilise or C++11, C++14 Threading
- Implementation of threading utilising
  - Mutex will be awarded a ***Basic Mark***
  - Critical Section will be awarded an ***Intermediate Mark***
  - Thread Pooling will be awarded an ***Advanced Mark***

- Please note Git commits will be checked week to week
- Movement can be tile to tile frame to frame.
- During demo it should be possible to see movement of NPCs
- Scene is complete when all NPCs have reached the player.
- No overlapping of NPCs on tiles
- Number of NPCs is related to MAP size
  (30x30 => 5 NPC's, 100x100 =>50 NPC's and 1000x1000 => 500 NPC's)

Marking Scheme (Practical 25%)

0 -35	35-75	75-100
• A selection of the basic title requirements have been implemented to a basic level • Subsystem implementation will achieve minimum functionality • Title implementation may contain some syntax and/or run-time errors • Title implementation code will be poorly commented and/or formatted • Title implementation will contain basic features; application will not be tested properly • Title implementation code will not follow applicable coding conventions • Title implementation will not have a mechanism for human visualisation of current execution state using SFML and/or Starter Kit. • Threaded implementation using a Mutex	• Title implementation requirement have been implemented to an acceptable level • Subsystem implementation will achieve expected functionality • Title implementation will not contain syntax and/or run-time errors • Title implementation code will be reasonably commented and/or formatted • Concurrent implementation will contain assignment features and course grained 2 process implementation • Title will be tested to a reasonable degree • Title implementation code will follow appropriate coding conventions • Title implementation will have a mechanism for human visualisation of concurrent execution using SFML and/or Starter Kit. • Threaded implementation of a Critical Section	• Title implementation requirement have been implemented to an advanced level • Subsystem implementation will not contain syntax and/or run-time errors • Concurrent implementation code will be well commented and/or formatted • Title will be expertly tested • Title implementation of code will follow coding conventions • Title implementation will have an advanced mechanism for human visualisation of concurrent execution state using SFML and/or Starter Kit. • Threaded implementation of Semaphore which manages a resource pool