**Spike:** 04

**Title:** Non-Blocking Game Loop

**Author:** Ford Killeen, 9731822

**Goals / deliverables:**

*Goals this spike aims to achieve:*

* Create a version of GridWorld that uses non-blocking keyboard input
* The input function must not block/pause the game loop

*Deliverables required:*

* Code for the non-blocking game loop
* Spike report

**Technologies, Tools, and Resources used:**

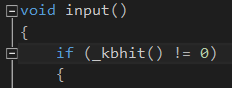
The following is required to complete this spike:

* Visual Studio 2015
* Existing GridWorld blocking implementation (or specification)

**Tasks undertaken:**

The list below details the steps taken to complete this spike.

* Copy your spike 01 GridWorld (blocking) implementation to a new folder
* Investigate how we can implement a non-blocking version of the game
* After deciding on a way of achieving the non-blocking input, begin modifying the existing GridWorld game
* For my implementation I initially thought of doing a multi-threaded implementation but later decided against it when I found the \_kbhit() function, which is part of the C run-time libraries and can be used by adding #include <conio.h>
* This function returns an int value greater than 0 if there was a keyboard entry, this makes it very easy to determine whether any input should be retrieved and does not block the game loop



* Build and test your implementation until you believe it is working well enough
* Note, another thing I did was create a queue of inputs that the update() function would pull from. This was done by using a deque object

**What we found out:**

We found out that the implementation of a non-blocking game loop is not all that difficult to do, though there are many different ways to achieve the same goal.

**Recommendations:**

I would recommend finding the easiest or simplest solution for a non-blocking implementation you can, that you actually understand, and implement that. Don’t waste time trying to understand something different if you can complete the same problem with an easier method.