**Spike:** Spike\_4

**Title:** Non-blocking Game loops

**Author:** Peter Argent, 7649991

**Goals / deliverables:**

* This Spike was centred on using threads to make a non-blocking game loop.
* The source code for the Spike can be found at <https://github.com/stormcroe/GamesProgramming2016/tree/master/Spike_04>

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

List of information needed by someone trying to reproduce this work

* Visual Studio 2013
* <http://www.cplusplus.com/reference/thread/thread/?kw=thread>
* Cplusplus.com as a reference site for more C++ syntax
* A copy of spike one’s source code to work off.

**Tasks undertaken:**

* Open a Copy of spike 1’s GridWorld in Visual Studio 2013
* Open the gameController.cpp file
* Move the “input” variable to be a field and have it initialize to NULL
* Change the parameters on GetInput() to GetInput(char\* input, bool \*gameover)  
   The input parameter is a pointer to the \_input field, this allows us to pass the input information between threads, the gameover parameter allows us to check for the game finishing so that the thread’s internal loop ends.
* Add an If statement to each threads’ while loop so that the code within activates only on: (input == NULL) for the input loop, and (\_input != NULL) for the game loop.
* Write the code that would create and join the input thread to the main thread.   
   thread GetUserInputThread(GetInput, &\_input, &\_gameover);  
   //… The game loop code is here.  
   GetUserInputThread.join();
* Note: Make sure the \_Input is reset to NULL after the game loop, otherwise the input thread won’t trigger again.

**What we found out:**

Describe the outcomes, and how they relate to the spike topic + graphs/screenshots/outputs as needed

* One thing that we found out during this spike is that passing by reference in C++ cannot be done if used in a method being passed to a new thread. However using pointers works perfectly. [Search cplusplus.com for how to use pointers]
* We need to ensure that the game loop code did not run if there was no input and the input not run while there was an input otherwise the console would quickly fill and move at a pace that a human could not read. Using Non-blocking game loops is not a reasonable solution for console games running on a basic game loop; only if there is a time sensitive flag should they be used eg. A physics engine or day-night cycle

**Open issues/risks** [Optional – **remove** heading/section if not used!]**:**

* Risk/Issue 1: Using pass by reference in the GetInput() method breaks down when attempting to send data back across threads.

**Recommendations:**

I recommend that people have a requisite understanding of how pointers are used in C++ before attempting this spike, without understanding how they are used, someone cannot get the code to work properly and send the input data across threads to the game loop.