**Spike:** Spike\_3

**Title:** Debugger Use

**Author:** Peter Argent, 7649991

**Goals / deliverables:**

* There is a fixed copy of Spike3.cpp at <https://github.com/stormcroe/GamesProgramming2016/tree/master/Spike3>

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

List of information needed by someone trying to reproduce this work

* Visual Studio 2013
* Cplusplus.com is handy for syntax, and general help.

**Tasks undertaken:**

* Compile the spike3.cpp that we were given in Visual Studio.
* When the program brakes, choose to create a breakpoint in the code where the program stopped working.
* Check that all variables have been initialized using the “Autos” and “Locals” Tabs in the bottom left of the IDE
* If you cannot find bad variables try using the “Call Stack” window to step back from the breakpoint and find if the variable you were dealing with had bad values from elsewhere.
* To find the memory leak, make sure that each resource has had delete called on it, check the main() function especially, and then each class separately.

**What we found out:**

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

* Logic Errors can only be found by noticing where a program does not do what you expect it to and setting breakpoints there so you can step back and forth in code to find the error.
* Syntactical errors are often picked up in Visual Studio using the inbuilt ‘Intellisense’ that underlines wrong or missing syntax, if it would not compile, the intellisense picks it up. If an error would break a program in runtime, the intellisense will not pick it up
* By using a breakpoint and the Locals, Autos and Call Stack tabs, we can step through code and read what variables are doing at certain points in the program. This allows us to see whether a variable is acting in the correct way.

**Recommendations** [Optional – **remove** heading/section if not used!]**:**

I recommend having multiple programmers looking at the same code base when debugging, as having multiple viewpoints and people to explain code to helps immensely during testing/debugging.