Intro to Visual Studio

Ford Killeen [9731822]

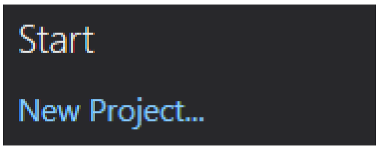
Requirements

For this introduction to Visual Studio, you will need a copy of Microsoft Visual Studio 2015. This introduction is being completed using Visual Studio 2015 Community edition (version 14.0.25325.01 update 3). You will also need a little C++ knowledge, as that is what our project will be built on.

Creating a new Project in Visual Studio

Creating the Project

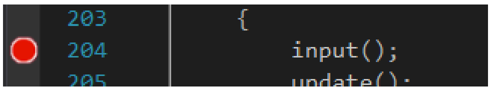
To create a new project in Visual Studio follow the below steps.



- Open up your copy of Visual Studio 2015
- You should see the Visual Studio Start Page, consisting of some handy shortcuts and recent news about Visual Studio.
- In the left hand side of the screen, select New Project…
- In the New Project window, select Visual C++ from the left pane, then select Win32 Console Application from the main pane
- In the settings below, give your project a name, in this case we can name it GridWorld, then choose the location you wish to store the project and click OK
- When the Application Wizard appears, click Next and then tick the Empty Project box to ensure we start with a fresh, clean project
- Click Finish
- Now that your project has been created, we will start adding files we have previously created in the C++ language. To do this, right click on your GridWorld project name in the Solution Explorer (usually on the right hand side of the screen), expand Add and select Existing Item… (or Shift+Alt+A)
- At this stage ensure any source files you wish to add, have been manually copied to the directory in which your project is stored
- In the Add Existing Item window, navigate to your source file and after selecting it, click Add
- Now that your pre-completed source file has been added, you should now see it in the Source Files dropdown within the Solution Explorer. If you didn’t have any source files to add, you can follow the steps above and select Add > New Item to add a new C++ File (.cpp)
- Now you’re ready to compile your project! To do this, expand the Build options from the top toolbar and then select either Compile or Build. Generally most people will choose Build as it compiles the code along with doing a few other tasks and error checks.
- Once you see that the build has been successful in the Output pane, you will be good to run your application
- To run your application you can hit the green Play button in the toolbar, or go into the Debug dropdown menu and select Start Debugging/Start Without Debugging

Breakpoints in Visual Studio

To create a breakpoint in Visual studio, follow the below steps. A breakpoint allows you to pause your application mid process to check what is going on.

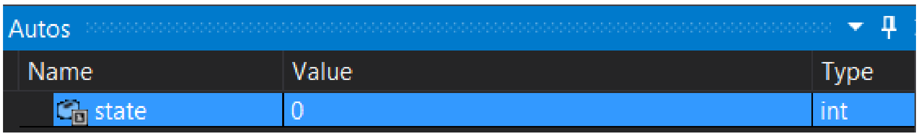


```
    {
        input();
        update();
```

- Make sure your GridWorld project is open and your source file (.cpp) is open in the editor
- Looking at the main text editor in the centre of the screen, you should see line numbers along the left hand side. Just to the left of those line numbers is a vertical grey bar. This bar is not just for looks, it has some very powerful capabilities too.
- Move your mouse cursor over the bar and single left click on it. You should see a nice red circle appear. That's a breakpoint!
- To remove a breakpoint follow the same steps, except this time clicking on the breakpoint (red circle) itself
- Now find somewhere interesting to put a breakpoint so that we can see it in action. Scroll down to your game loop in the code, and on the line that you call your `input()` method, create a breakpoint
- Now click the green play button to begin debugging the application. You should see the code execute up until the first input call, then the application stops and you will be taken to that line number in the editor
- From here you can continue the application or step through it. Stepping through it allows you to go through the flow of your code one line at a time, even going deep inside other functions and methods.

Inspecting variables during debugging

Being able to inspect variables and their values while debugging is very powerful and useful. To learn how to do this follow the steps below.



Autos

Name	Value	Type
state	0	int

- To begin, start debugging the application using the local windows debugger built into Visual Studio
- You will see your application once again stop at the breakpoint
- With the application stopped, take a look around the editor and you should see an information panel titled Autos (or Locals or Watch 1). This is your variable inspector and shows you the name, value and type of any variables within your applications scope.
- While viewing your variables, if you notice any popping in and out of the inspector that you want to watch constantly, you can right click on any variable within the code view and select Add Watch. This adds it to the watch for you to monitor continuously.