C# wrapper for Python scripts

Project title: 3D Geometry HULL Calculations

SEP Group: HULL2

Team member: Phuc Hoang Nguyen – a1680079

Research Scope

Part A can be done using Python as there are libraries and components already built to be used with our implementation. However, the industry client requires us to have a C# wrapper for any programming language, thus we need to build C# wrapper for Python.

Basic concept

The team needs an implementation of the Python programming language which is tightly integrated with .NET. This also needs to be able to use .NET and Python libraries. Additionally, the team has to choose something that has been included in the nuget package manager to streamline our workflow.

After much consideration, the team has chosen IronPython.NET as it meets all the requirements.

Preliminary research

IronPython.NET a wrapper for calling into Python code from C# in .Net 3.5. In .Net 4.0 and above, using the dynamic keyword makes it easy to call into Python without having to worry about types. However, if you must use .Net 3.5 for any reason (like better XP support) this wrapper makes it easier to call Python code from C#.

An example usage can be found below:

```
// Compile script
PythonScript script = new PythonScript("example.py");

// Call top-level function
string returnValue = script.CallFunction<string>("get_string", "World");
Console.WriteLine(returnValue);

// Create Python object
PythonClass pyClass = script.GetClass("ExampleClass");
PythonObject pyObj = pyClass.Instantiate(5);

// Get property
int number = pyObj.GetProperty<int>("number");
Console.WriteLine("Number property is {0}", number);

// Get return value as Python object
PythonObject pyObj2 = pyObj.CallMethod("get_object");
int number2 = pyObj2.GetProperty<int>("number");
Console.WriteLine("Number property for 2nd object is {0}", number2);
```