

CONTENTS

2	Perquisites.....	1
3	Visual Studio Installation.....	1
4	CALib Project Initialization	2
4.1	CALib Projects	4
5	Cones World Visualization	4
6	F# Learning Resources	7

1 PERQUISITES

- Visual Studio (VS) 2019 with .Net Core and F# support
- Several NuGet packages that are referenced in the project itself
 - Note these don't have to be loaded separately but will be pulled in automatically when the 'restore' step is done (see below)

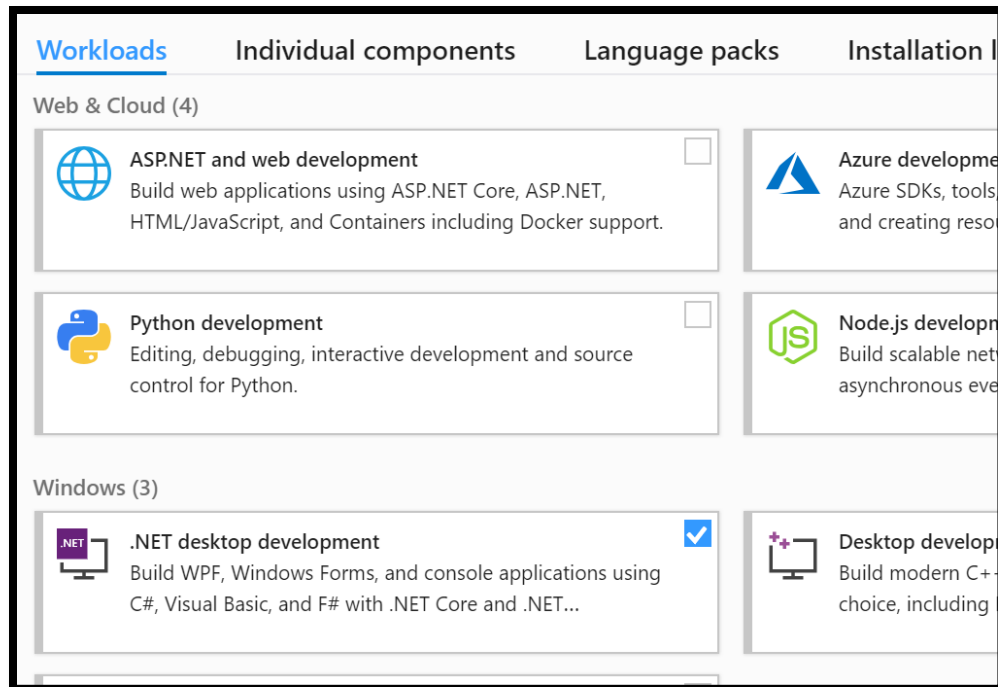
2 VISUAL STUDIO INSTALLATION

There are several methods to obtain VS including a CIT requisition. However, the easiest method to get started is to install the VS 2019 community edition which is free and can be obtained from here:

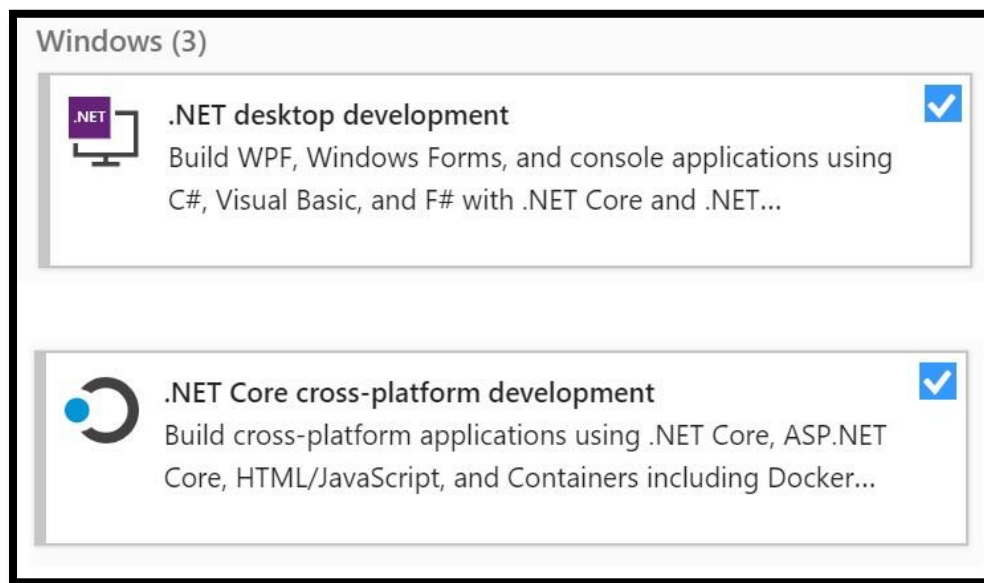
<https://visualstudio.microsoft.com/vs/community/>

The download is an installer program that ultimately installs VS with the configured options. These options are selected in the installer before proceeding to VS installation.

The 'Workloads' tab of the installer contains boxes represent bundled functionality and looks like this:



Select the boxes corresponding to “.Net desktop environment” and “.NET Core cross-platform development” to get support for F# and .NET Core:



Proceed to installing Visual Studio. It will take a while to install.

3 CALIB PROJECT INITIALIZATION

The source code for CALib is available form GitHub:

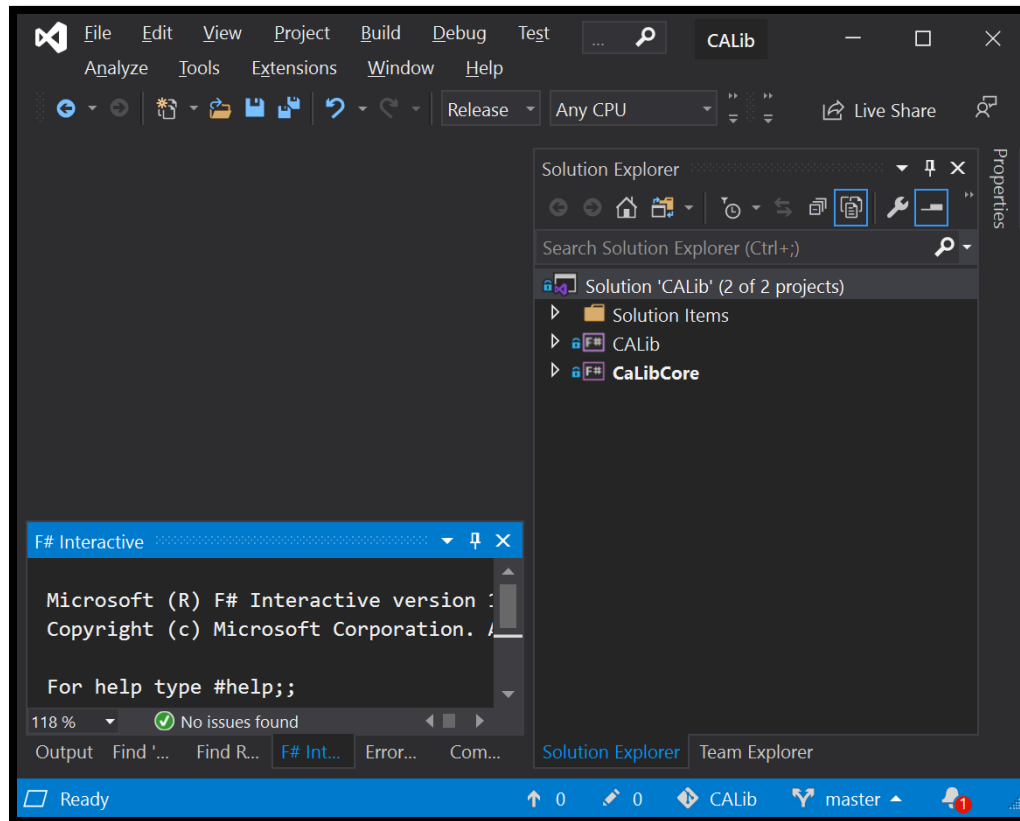
CALIB INSTALLATION AND SETUP

<https://github.com/fwaris/CALib>

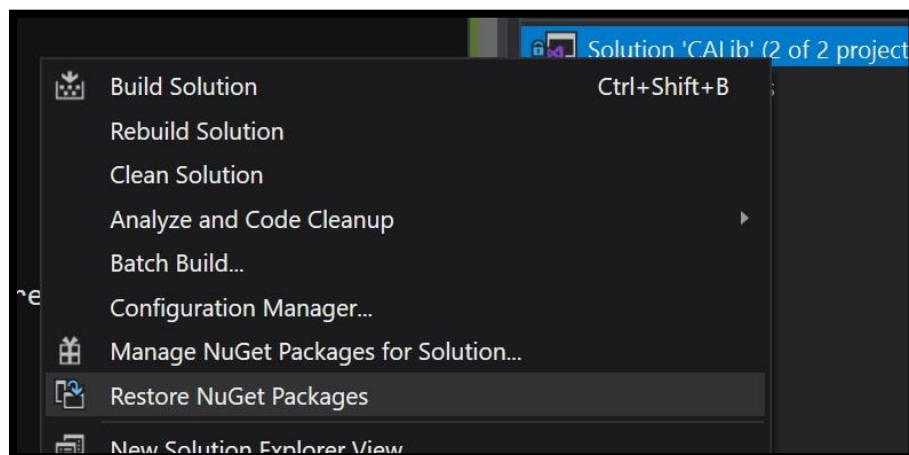
Download and extract the source code from this repository.

Open the visual studio 'solution' file CALib.sln that is contained in the extract.

You should see something like the following:

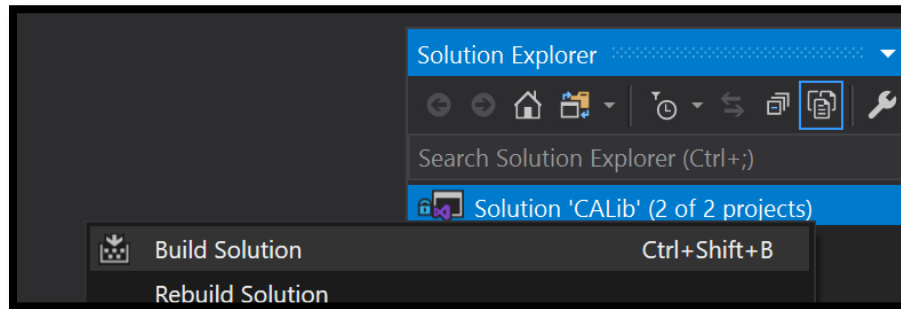


Right click on the 'Solution' node of the 'Solution Explorer' window and select 'Restore NuGet Packages' as shown below:



This will pull in all the required packages from the NuGet repository on the internet but it may take a while to complete.

After the package restore is complete. Right click again on the Solution node and choose “Build Solution” as shown below:



This will compile the solution. A successful compile means that all packages are loaded and the Visual Studio is correctly configured.

3.1 CALIB PROJECTS

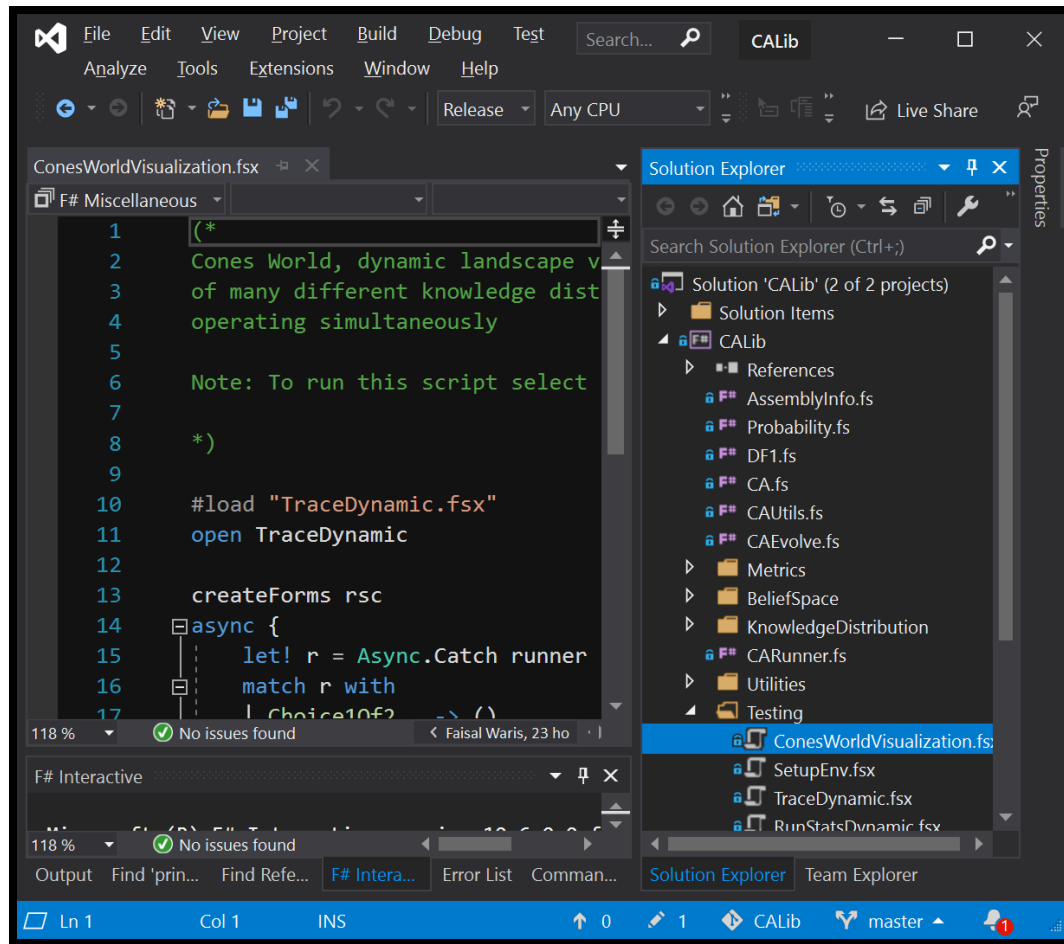
The CALib solution contains two projects:

- CALib - this can be used in two ways:
 1. Compiled to a DLL and linked with a .Net application to solve optimization problems
 2. Interactively with F# interactive (.fsx) script files, for research and experimentation purposes
- CaLibCore - a version of CALib that runs on Linux (with .Net Core). Its main purpose is to conduct experiments on the Wayne State grid computing environment

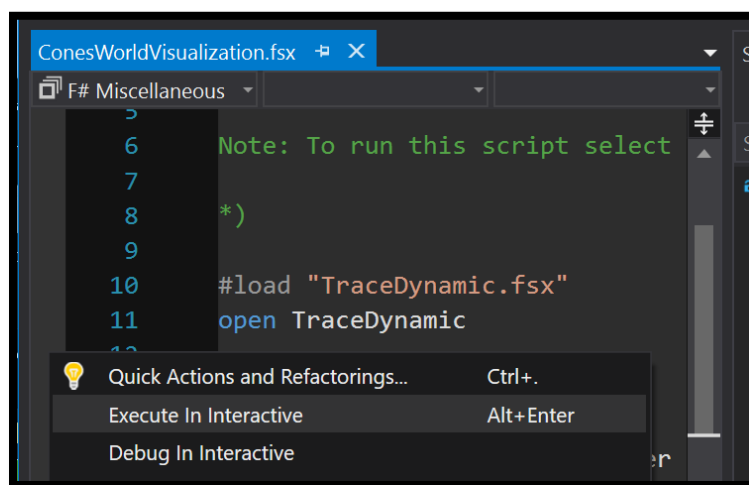
4 CONES WORLD VISUALIZATION

Open the F# script file ConesWorldVisualization.fsx. It is under the 'Testing' folder under CALib project as shown below:

CALIB INSTALLATION AND SETUP



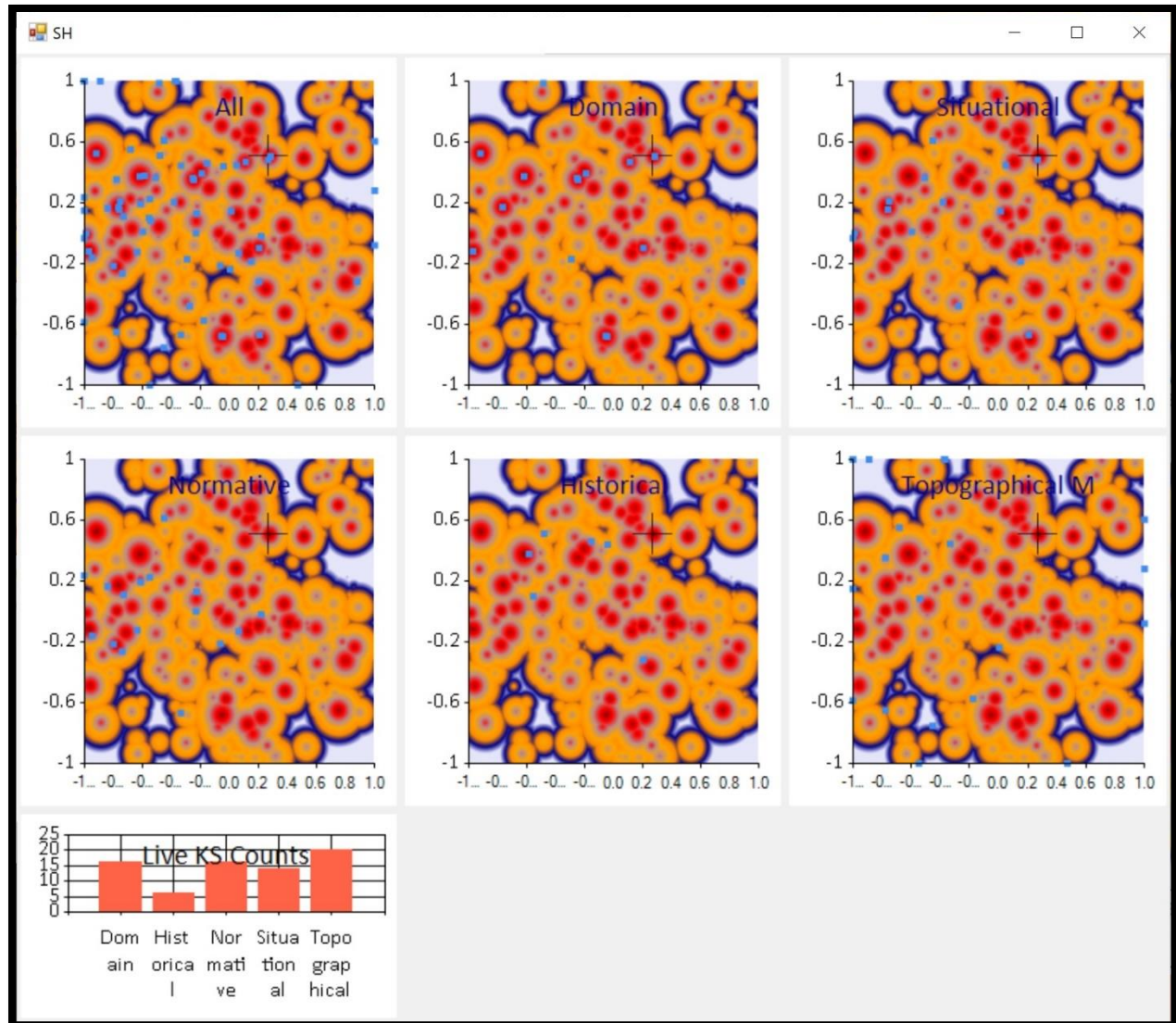
Select all of text in the `ConesWorldVisualization.fsx` file and hit Alt-Enter or right click and choose “Execute in Interactive” as shown below:



This will compile and execute the code in the script file in the F# interactive window. The script references the code in almost the entire project so it might take a while to compile.

CALIB INSTALLATION AND SETUP

After a while, you should see a series of windows appear that look like the one below:



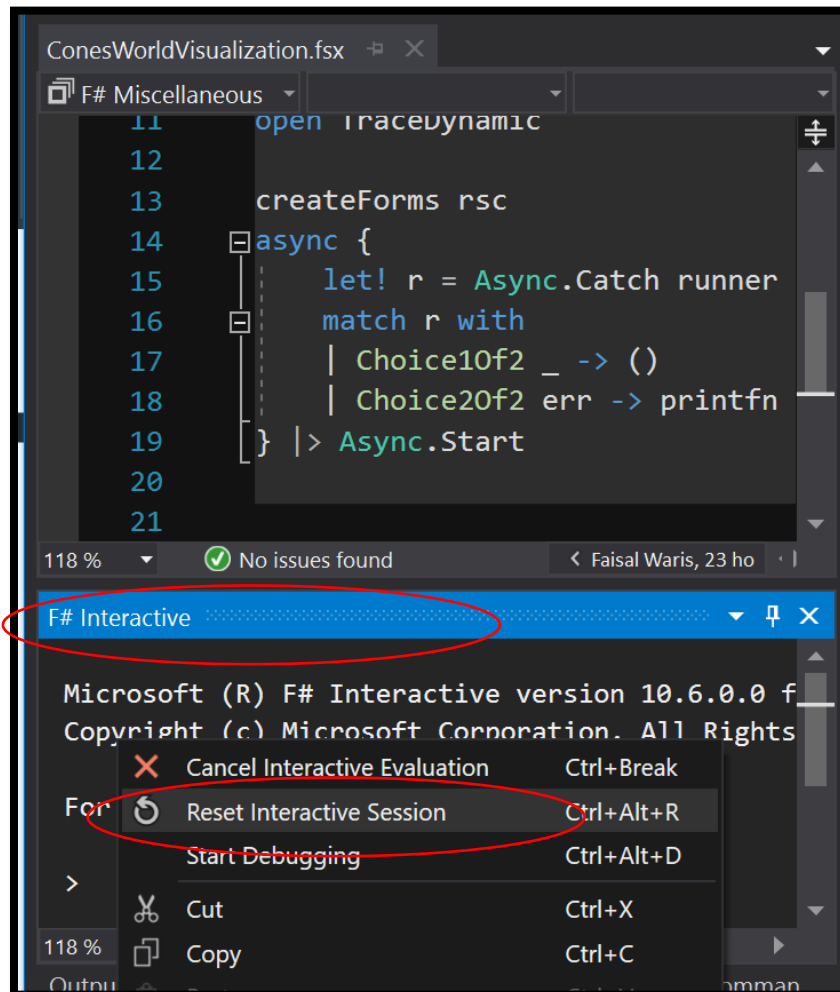
Each window is for a different knowledge distribution mechanism and visualizes the operation the CA run for that knowledge distribution. The windows are live in that the actual movement of the individuals under the control of the different knowledge sources are displayed with a delay of 250ms delay per generation.

The background in the top frames of the window are the contour plot from the Cones World landscape that is currently being operated upon.

Periodically, the landscape will be changed according the configured A-value in the script. The new landscape contour plot will appear when that occurs.

This process runs for a long time. You can kill it from the "F# Interactive" windows or by closing Visual Studio itself.

To terminate a running F# script, right click in the “F# Interactive” window and select “Reset Interactive Session” as shown below:



5 F# LEARNING RESOURCES

F# is a .Net language. It was developed a long time after C# and therefore is not as well known as C#. Its popularity is continuing to increase with time and major companies such as Walmart are now using it.

F# is a functional-first language that also supports object-oriented (OO) programming. If you have a background in Haskell or OCaml, you should feel at home with F#.

The CALib code is written in a functional programming style with minimal use of mutable variables and strong use of recursive functions. This can feel odd to those who are primarily OO developers (e.g. in Java or C#).

Luckily the F# learning resources are vast with many blogs, videos and books available.

There are over 20 F# books available from the Wayne State library 'Skillsoft' article database:

CALIB INSTALLATION AND SETUP

Just search <https://wayne.skillport.com> for F#.

Another starting point for F# learning resources is the <http://fsharp.org> site – which is the primary site for the F# language.