A *Literate Programming* and *Animation Environment* for *VDM*

Motivation

- VDM is a formal modelling language, used to produce models of primarily digital systems.
- These models are occasionally written to use graphical output.
- This project aims to assist in providing visual interactivity (**animation**).
- Additionally, it aims to create a **notebook** environment similar to *Jupyter Notebook*
- This is to assist with intuitive development of the models

Tools

The tools produced by this project are the following:

- `vdmj-remote`, an RPC system for *VDMJ* (the VDM interpreter) also allowing the hosting of web content
- `vdm-notebook-extension`, an extension for the VS Code IDE to run and manage the VDM notebook environment

How it works

1. The `vdm-notebook-extension` uses the `vdmj-remote` system to execute VDM code from cells.
2. The output by default is a console connected to the backend, this console can be used to execute normal *VDMJ* terminal commands.
3. Cells can contain an annotation such as `--@WebGUI(<path>)` to identify a path containing static web content to be hosted by the backend.
4. The web content can access the VDM backend via REST requests to execute terminal commands.

Additional features

One of the most useful aspects of this project is the RPC system. It is currently a simple REST API though could quickly be adapted to a true RPC system with a specified protocol.

Project status and future work

- The `vdmj-remote` system works well and could be improved to make a very functional system to include VDM in web infrastructure.
- The `vdm-notebook-extension` works mostly as specified but not exactly like other notebook systems:
  - *VDMJ* is not designed for piecemeal code interpretation, I have had to use temp files
  - Notebooks are designed for shell-like languages, neither VDM nor the *VDMJ* terminal are shell-like
  - Notebooks are not designed to output web content

If work was done on a shell-like version of VDM or Python bindings for *VDMJ* then the development of a notebook environment would be far simpler.

Animation

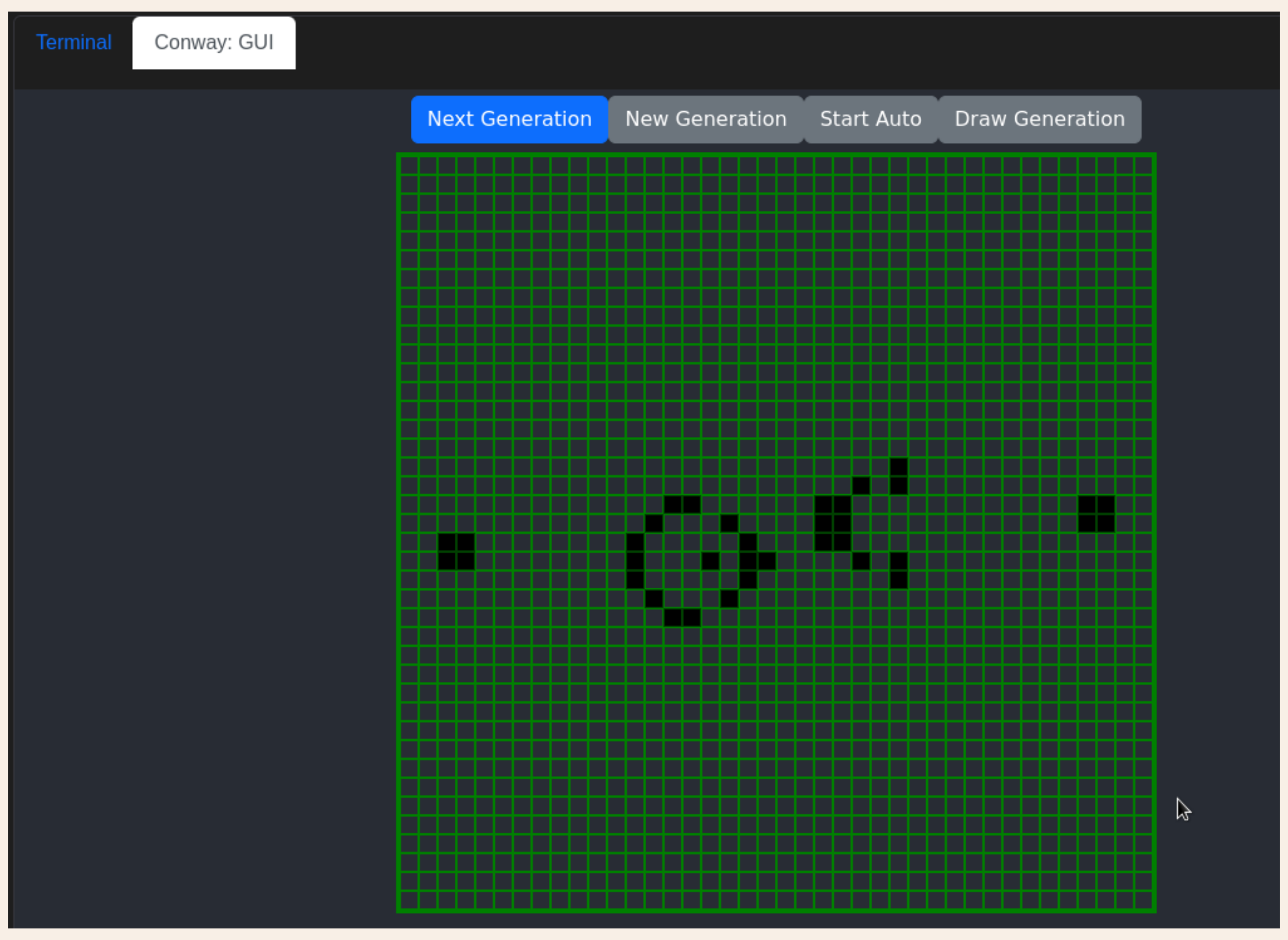
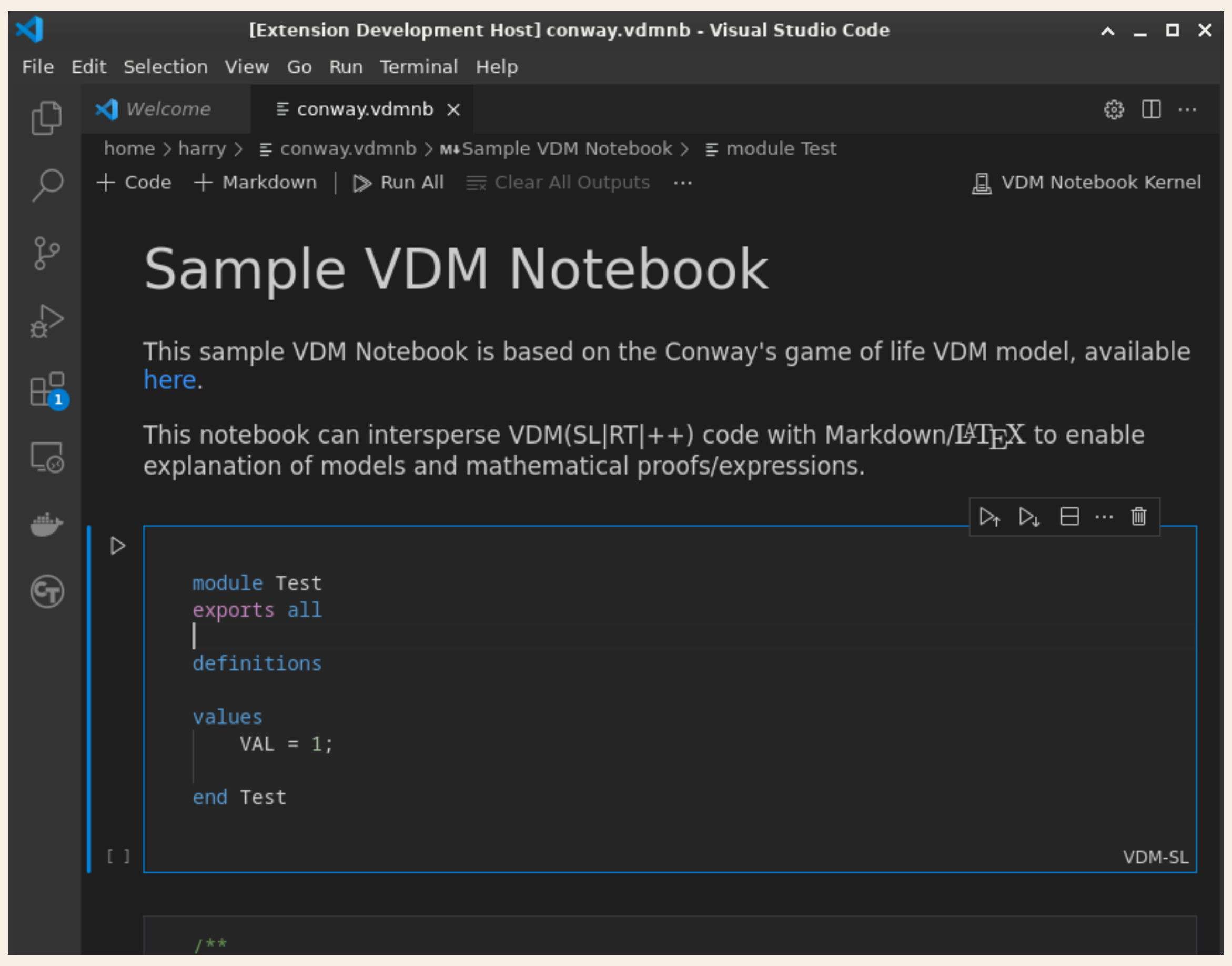


Figure: Conway's Game of Life GUI example

Notebook


[Extension Development Host] conway.vdmnb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Welcome
conway.vdmnb
home > harry > conway.vdmnb > Sample VDM Notebook > module Test
+ Code + Markdown Run All Clear All Outputs ...
VDM Notebook Kernel
[ ]
VDM-SL

Sample VDM Notebook

This sample VDM Notebook is based on the Conway's game of life VDM model, available here.

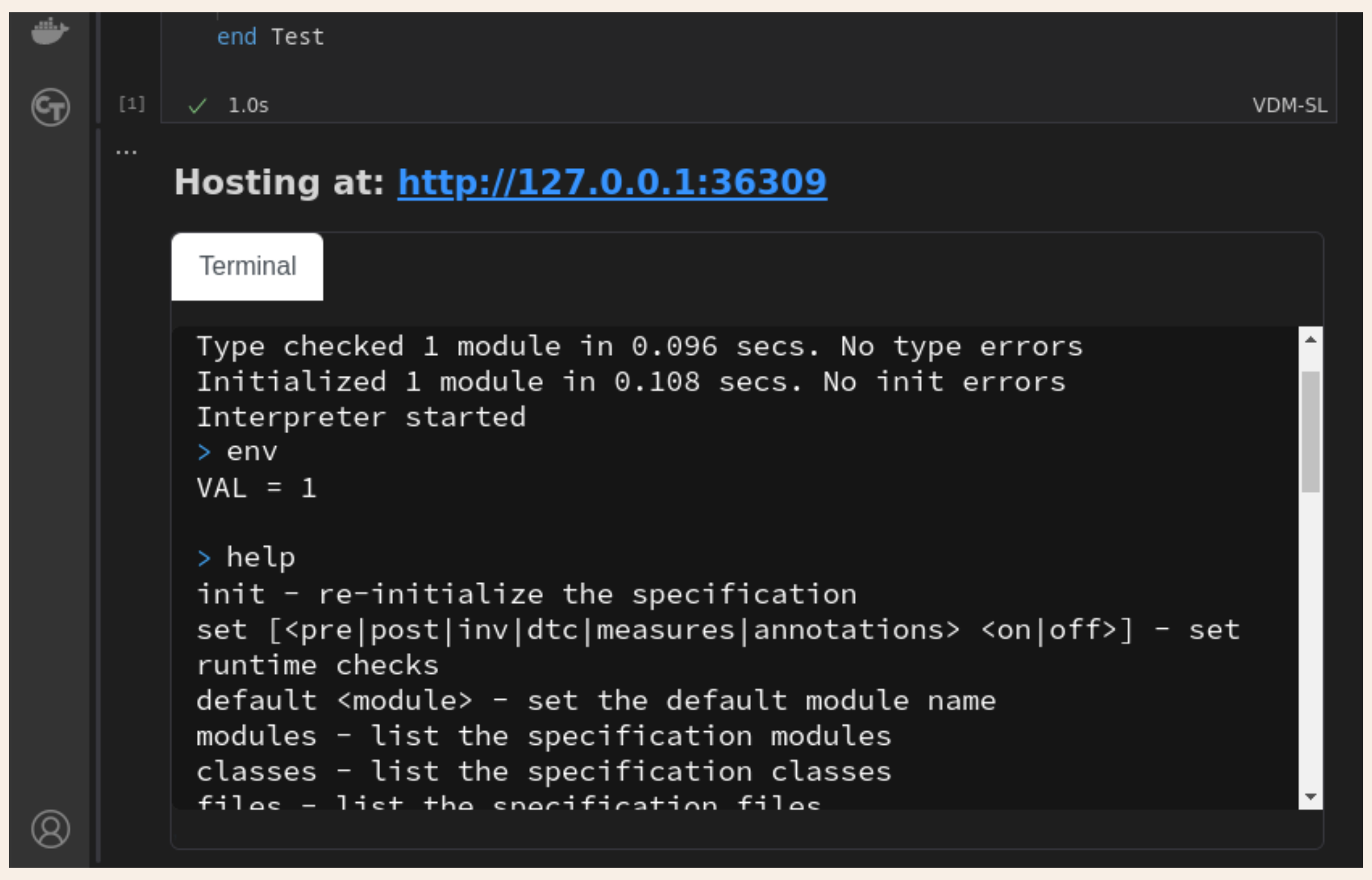
This notebook can intersperse VDM(SL|RT|++) code with Markdown/LaTeX to enable explanation of models and mathematical proofs/expressions.

```
module Test
exports all

definitions

values
    VAL = 1;

end Test
```

```
/**
```

Figure: Notebook example

Console



```
end Test
```

Hosting at: http://127.0.0.1:36309

```
Type checked 1 module in 0.096 secs. No type errors
Initialized 1 module in 0.108 secs. No init errors
Interpreter started
> env
VAL = 1

> help
init - re-initialize the specification
set [<pre|post|inv|dtc|measures|annotations> <on|off>] - set
runtime checks
default <module> - set the default module name
modules - list the specification modules
classes - list the specification classes
files - list the specification files
```

Figure: Console example (from Notebook example code)

Harry Hughes (190227241)
Supervisor: Leo Freitas
Newcastle University