Advanced Scripting Workshop (AWS): The Game
[Lockdown Edition]

Built in only three days, with no prior knowledge of this module or the sheer size of the hellish task I had taken on. (With credit to my girlfriend, who graciously agreed to postpone valentines for a day.)

Contents:

---

1. Overview

The Advanced Scripting Workshop game (AWS) is an old-school, 8-bit gaming platform that is designed to present information to the player.

In Lockdown Edition, the player controls a human with access to several rooms of their house. The player can walk between rooms to interact with the room items, which present information in short snippets. The player is free to return to items in any order, and spend as little or as long as they like reading each panel.

Directions to the next item in the logical flow are given at the end of each interaction. If followed, the order is similar to a conventional presentation. Boundary conditions are applied to the outer walls and items to mimic a solid, impassable object.

2. Motivation

“Definition of presentation is left purposely vague…Use your creativity”

-Dave Schultz, student presentations document (2021)

This presentation format is somewhat motivated by GatherTown, which supports single poster uploads. This seemed like a brilliant idea, but I wondered if it could be expanded to presentations. The playable-presentation format has some practical advantages:

- The presentation is not constrained by a time limit, only the amount of time the player chooses to commit.
- The presentation can be observed at the players pace, they can go back and re-read, or jump ahead to what interests them in a non-linear fashion.
- The presentation could support various media files, such as audio, video recordings, and images.

By breaking the presentation into separate interactable items:

- The player receives a break and an activity between large amounts of information
- The player can associate items with their content- ideal for visual learners.

This game is built with a python module called Pygame, that allows game coding from scratch. Having discovered the module only three days before submission, I set about creating the game I envisioned as fast as possible, designing it around the house I am “locked down” in.

3. Getting Started

NOTE1: The game did compile into an executeable on ARCHER2 using CX_Freeze before an unknown error occurred. Until I find a fix, please play locally only (no compile required).

NOTE2: This game has been written on Mac OSX only, but should run in windows.

To play the game, clone the presentation pack repository from github to your local machine. This contains the main source code, and many more files required to load the game visuals, music etc.

`git clone https://github.com/jcarter2024/Presentation_Pack.git`

The code imports several modules that are likely included with your current python3 environment:
`sys, numpy, itertools`

and will require `pygame`. You can install pygame, and any missing modules with Pip:

```
python3 -m pip install -U pygame --user
```

For detailed pygame setup please see their startup guide:
https://www.pygame.org/wiki/GettingStarted

With this installed, you should be ready to play. Open the game with python 3 in the terminal:

```
python3 scripts/main.py
```

Or if editing, open in an IDE and click **Run** (tested with Anaconda Spyder only.)

Pygame uses system-default programmes to open the game window, but it may not pop up to the front of your workspace. If you don’t see a window, check behind other programmes.

You should see the following screen:

Click on **Enter** to begin the game.

4. Navigation:

On entry, the player spawns at the door. There are four areas to explore, the lounge, kitchen, bedroom and bathroom. The presentation is accessed via item-interactions, the items and their logical order* are:

Lounge:	Window[1], Book (coffee table)[4]
Kitchen:	Pool-Table[3], Globe[2]
Bedroom:	Computer[5], Desk[6]
Bathroom:	Bath[7], Mirror[8]

The top left lounge window is the starting point, and directions given thereafter.

Figure 1. Various item locations (red) and initial player location (blue).

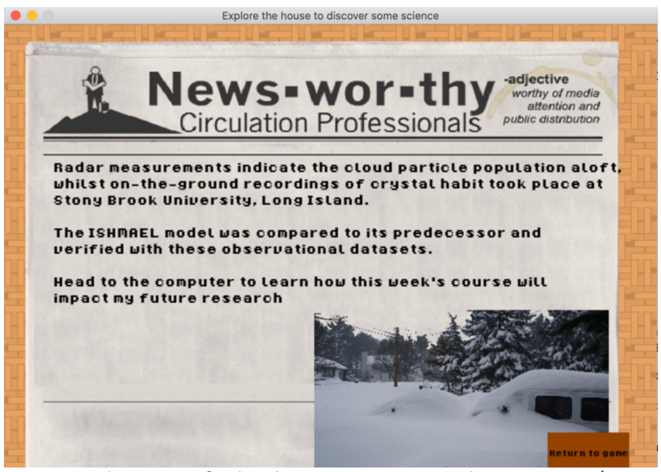

Figure 2: The screen for book interaction. Note the next page/ return to game button.

To walk, use the up/down/left/right arrow keys on your keyboard.
To interact, stand close to an object and press the **x** key. This will bring up the presentation screen.
Presentation screens have multiple pages that can be scrolled by clicking the *next page* button.

5. Glowscript Riming Simulation

Before beginning the game on Friday, I had a very conventional PowerPoint presentation lined up. The twist of creativity I would add, was a simulation of the riming process. This topic is at the heart of my PhD, and at its core is a dynamical process that would be better explored visually, rather than in writing.

I eventually came across Glowscript, an online IDE developed to visualise physics. The package enables a somewhat-quick workaround to create geometrical shapes, that can be rotated and explored in real-time.

Unfortunately, the IDE is only available online, so this simulation can't be embedded to the game. However, there is an expanded Glowscript package that may make this available in the future.

The simulation shows an ellipsoid, which plays the role of a cloud ice crystal, and is conveniently similar to ISHMAEL's spheroidal representation. In the reference frame of the ice crystal, a field of droplets rises from the simulation base, either colliding with the crystal or passing by. The droplet field is replenished continually to simulate an approximately uniform droplet number concentration.

Colliding particles will remain fixed to the surface of the ellipsoid, just as supercooled droplets instantaneously freeze to the ice surface. This adds to the height of the ellipsoid, and the length to a lesser degree, causing both ice-mass growth and a tendency of the aspect ratio towards a sphere's. This too, mirrors the ISHMAEL code, and in turn the axial growth of real ice in clouds.

The ellipsoid is set to be slightly transparent, so that the *rimed* droplets can be observed after impact. The accretion rate and growth rate can be output in real-time. The graphical window can be rotated I any direction, and zoomed using the mouse and CTRL key.

I have saved the glowscript as a python file available in the top-level directory as **Riming.py**. This can be copied and pasted into the Glowscript IDE, (Note: free sign up is required).
More information on getting started with Glowscript is available here
https://www.glowscript.org/docs/VPythonDocs/index.html

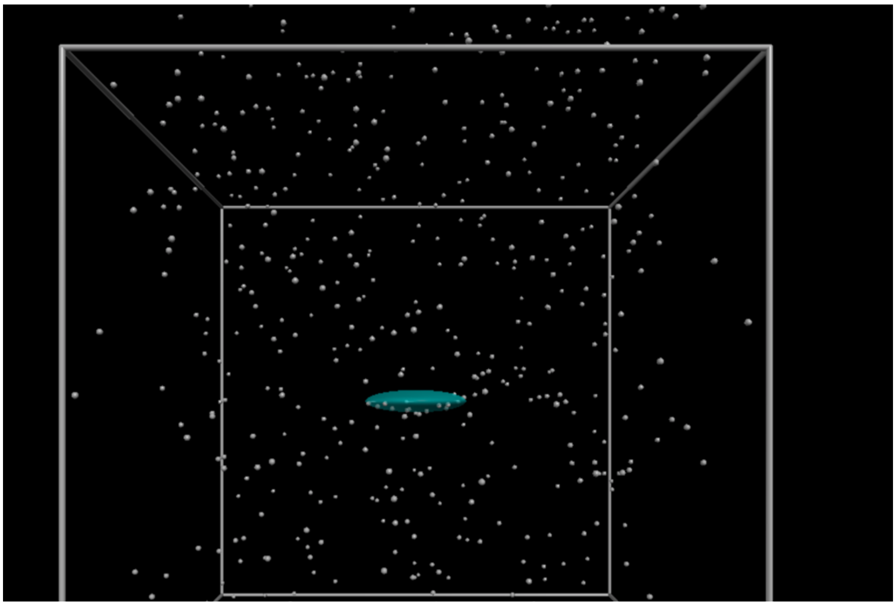

Figure 3: Glowscript riming simulation. At an early time, the crystal is approximately planar and begins to accrete droplets.

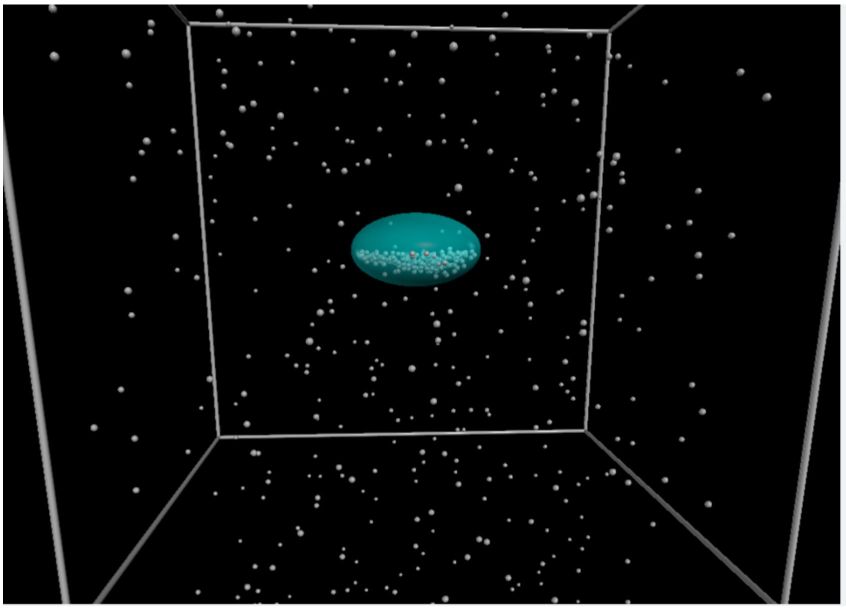

Figure 4: At a later time, the crystal has accreted many droplets. Correspondingly the crystal tends towards a more spherical shape.

6. Full Transcript:

Below is the full transcript of all screens, just in case you're pressed for time (or dislike having fun).

<u>Window</u>

Topic: What is cloud microphysics why is it important.
Visual connection: Look outside → See clouds → establish context

Though clouds can exist on the kilometre scale, they consist entirely of small particles that occupy the microscale, that typically grow no larger than a few millimetres. These particles each have unique qualities that are generated during formation and are built upon by interactions with their local environment. Capturing the consequences of those interactions is a key challenge for weather forecasting, as the accuracy of forecasts depends on a computational model's ability to correctly predict the interactions of hundreds of millions of particles.

One important subset of these particles is cloud ice. Unlike liquid droplets, ice can form in a multitude of shapes from stellar crystals to bullets, and this shape affects the crystal's ability to interact with other crystals. It may become more likely to fracture, generating reflective ice-shards, or it may aggregate to other crystals more readily, creating large snowflakes. My research examines how models reproduce the riming process, an interaction between supercooled liquid water and cloud ice.

Head to the Globe to learn more about weather models and simulated ice.

<u>Globe</u>
Topic: Computational forecast models
Visual connection: Models are divided into grids that cover areas of the globe/broad view of weather simulation.

No computer in the world is powerful enough to simulate even a few cubic metres of cloud particles. To scale the cloud, models employ parametrisations that broadly capture qualities of the particle population and portion the particle size distribution into bins. Each size range can be operated on as one, with particle interactions determining the new size distribution at the end of the time step. A weather model employs a microphysics scheme to calculate the changing distribution, but these schemes vary in the qualities they record. Ice shape has often not been properly accounted for due to its complexity, but advances in computational power have provided an opportunity to more efficiently represent the shape of ice. Now the ISHMAEL microphysics scheme offers a unique opportunity to investigate the effects of riming.
Go to the Pool table to learn more

<u>Pool Table</u>
Topic: The riming process
Visual connection: Riming occurs when ice and supercooled droplets collide, this mirrors the collision of balls on a pool table. (NOTE: The balls indicate snooker, but it is coded as pool).

Liquid water can exist at temperatures as low as -40 degrees Celsius in a stable enough environment. Often, ice that forms at the top of clouds gains mass at a faster rate than droplets, causing it to descend through a field of supercooled droplets. On impact, droplets accrete to -or rime, the surface of the ice, immediately freezing and contributing a further mass gain. Riming has several interesting aspects, for example, it depletes the cloud's droplet and ice population, reducing overall cloud lifetime. It also alters the crystal shape, distorting the aerodynamic profile of the ice and further increasing fall speed.

A simulation is provided at this link: https://glowscript.org/ where you can load the file I have supplied.

It is not possible to capture the exact shapes of crystals, but the ISHMAEL scheme can parametrise them by representing geometries with spheroids. This inclusion allows the riming process to be more closely tied to the size evolution of the ice field. Examinations of the ISHMAEL model indicate that it more accurately captures the scale of riming than traditional models, and can produce a more accurate precipitation forecast

To read more about the case study, visit the newspaper on the coffee table.

<u>Book</u>
Topic: The case study
Visual connection: The case study is a real life event, here it is displayed as a newspaper print for emphasis. Also relates models to the real world.

To compare ISHMAEL and conventional schemes, a winter storm that impacted the north east united states in February 2013 was simulated. The storm brought record snowfall to several areas, causing two states to declare a state of emergency. The storm was particularly devastating to infrastructure, aviation and property.

This storm was chosen due to its well-defined phases of precipitation, which brought snow and rain initially, followed by a period of intense riming, and concluded by snowy aggregates. Radar measurements indicate the cloud particle population aloft, whilst on-the-ground recordings of crystal habit took place at Stony Brook University, Long Island. The ISHMAEL model was compared to its predecessor and verified with these observations.

<u>Computer</u>
Topic: How ARCHER2 will benfit my research
Visual connection: No explanation needed.

Running WRF simulations is an intensive process. With supercomputer access a single two-day simulation can be conducted within 24 hours at high resolution. Currently, this research is limited by the Manchester CSF-HPC which is extremely busy, and often undergoes periods of downtime.

With the use of Archer2 simulations can be carried out at higher resolutions without a large time penalty. This will directly impact ISHMAEL simulations by increasing the amount of ice-spheroids available to simulation, creating more feedback to the riming process due to localised environmental changes.
Archer will provide a crucial second platform to conduct simulations, not only broadening the availability of simulation time, but also providing an important safety net to combat HPC downtime.

<u>Desk</u>

Topic: How the BASH component of the course will aid my research
Visual connection: Crafting and 'writing' of code.

The BASH component of this course has provided new tools for file handling, that will allow for more efficient workflows, and increased confidence on the command line.

Compiling WRF, and making amendments to its source code can be a lengthy and stressful process. The addition of new commands, and information on environmental variables in particular, will be integral to resolving virtual environment issues.

Interacting with system files is inefficient in my preferred language of python. BASH naturally lends itself to file handling, and will reduce the amount of time wasted manually editing text files. Simple scripts can be built for otherwise laborious manual tasks.

<u>Bath</u>
Topic: GIT and Github
Visual connection: Dropping electrical items in the bath will cause damage, and its usually your own fault. Similarly, the damage caused by not backing up your code can be substantial, and broken code is usually your own error, stemming from an undocumented amendment.

The GIT component of this course will allow for my research tools to be structured, stored and developed in a cohesive manner. I often develop lengthy code, but have been unwilling to use GIT due to its perceived complexity. Now code can be safely backed up, and shared more easily for collaboration. As I work on and develop the ISHMAEL code, it will be essential to have records that can be shared with its authors across the world. Most importantly it will protect against the main source of error, my own edits that break the code in ever more confusing and deceptive ways.

<u>Mirror</u>
Topic: Conclusion
Visual connection: Reflect on the contents of the presentation, with brief summarised points.

To recap (and reflect), the content delivered on this course will make for a much more efficient workflow.
My research seeks to understand how we can improve computational models of the riming process, by incorporating ice shape.
WRF is used to simulate storm events, during which microphysics schemes are compared and assessed to deduce their accuracy.
These simulations are impeded by a lack of compute resource, which will be resolved by splitting my research between two HPC facilities, and by utilising ARCHER2 to conduct the most intensive simulations. Access to two facilities is crucial to accessing resources during downtime.
Compiling and editing this code is difficult, but made easier by increased knowledge of command-line tools, which will speed up many simple tasks previously done by hand.

Documenting edits to the source code is now possible with GIT. This allows for potentially catastrophic errors to be reverted, and also simplifies international collaboration. Most importantly it is a safe place to store a great many scripts that could all be lost with a simple hardware issue.

Future and Re-use

The game could be improved significantly with more time, but the base is there. It was something of a miracle that I managed to produce this in only a couple of days! If you would like to reuse this code, develop it and deploy it in future you are welcome to. Please do reference me!

Some improvements that could be made:
. Video inclusion with each interactive slide (presenter or graphics)
. Expanded map with wider areas to explore
. Minigames within the presentation
. Cleaner fonts and style (retro was a choice, but possibly a little unprofessional)