Graphics Display Options

One of the goals for the project is to have a graphical interface panel that the code can interact with. I will refer to this area as the **stage** (ardán). The code that the user writes should be able to manipulate this stage in some way. There are a few options for how this could be achieved.

Stages

Scratch-like

An approach I am calling *Scratch like* is an approach that would allow the user to control individual sprites that are displayed on the stage, the user can change the co-ordinates and scale of the sprites, allow sprites to interact by message passing.

Turtle graphics

Another approach is to allow the user to program a small robot (usually called the turtle). The turtle can draw lines as it moves around the plane. This approach is used by *LOGO* and its derivatives, it allows the user to program complex figures easily, by co-ordinating the movement of a curve. It's effectively like a programmable etch-a-sketch.

2D grid

An alternative of the *Scratch-like* approach is allowing the user to program different entities that exists on a 2D grid plane with integer co-ordinates, The user can move elements around the grid and customise the display of the grid. This is the approach used by programs like *AgentSheets*. For example a user could program a maze solver that can solve mazes drawn on the grid. While this approach could seem like a simplification of the *Scratch-like* approach it is powerful in its own right, by limiting the code to a 2D grid it enables the user to quickly build different environments, For example, a simple game akin to **Frogger** could be developed with relative ease. This approach would also enable cellular automata simulations akin to Conway's Game of Life.

Combination of all above options

In this approach we pass control of the format of the stage to the user, The user can create a runtime object of a certain type of stage (e.g., turtle, 2d grid) and then pass that around to be drawn on. This would allow the greatest variability. It may also be possible to allow the user to layer several stages over one another, but this might be excessive.

If this approach is taken it would allow more iterative work on the language to be done, A simple stage such as turtle might be created first, and then as the project proceeds more complex stage types might be added, e.g., 2D grid.

Sprites

If a stage like the *Scratch-like* stage is used then the user will have to be able to use sprites. It's extremely unlikely I could provide sufficient sprites with the language so the user will have to be able to upload their own. If this is allowed then a project is no longer just the code, a project becomes it's own entity containing code and other resources. The 2D grid stage would be more versatile if sprite upload was allowed, but it will be much less limited withouth them than the Scratch stage.