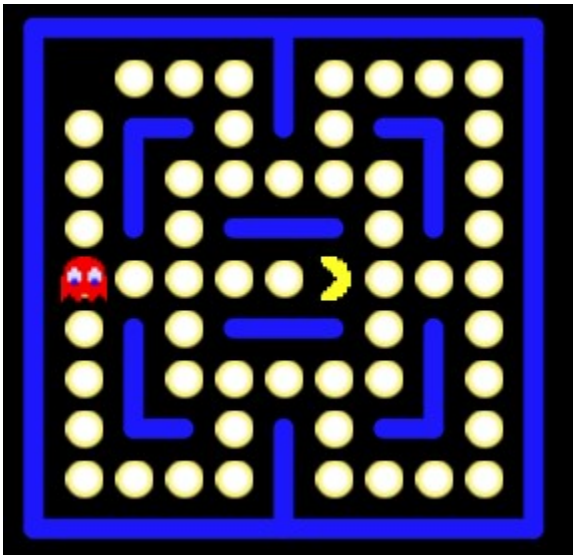
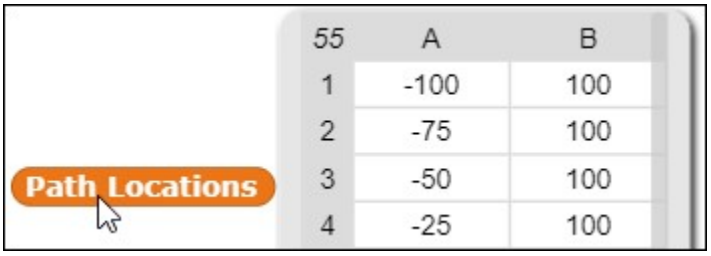
Constructing a Maze Game in Snap*!*

Pacman is a classic game in which a character navigates a maze munching power pills while being chased by ghosts. The principles used to reconstruct this game in Snap! are broadly applicable to any maze game. An inspection of the portion of the game board shown in the illustration below reveals that it is based on a 9 x 9 grid. The grid is separated into segments by blue walls. The remainder of the grid is populated by power pills that provide energy when consumed, the yellow Pacman character, and a red ghost.



There are several strategies that could be used to navigate a maze. For example, the character could move forward until it touches a blue wall, and then back up. In this implementation, a list of all of the valid path locations has been stored in a reporter block named **Path Locations**:

The elements of the 9 x 9 grid are placed 25 steps apart. For convenience, the upper left-hand corner of the grid has been placed at the location of an X-Coordinate of – 100 and a Y-Coordinate of + 100.

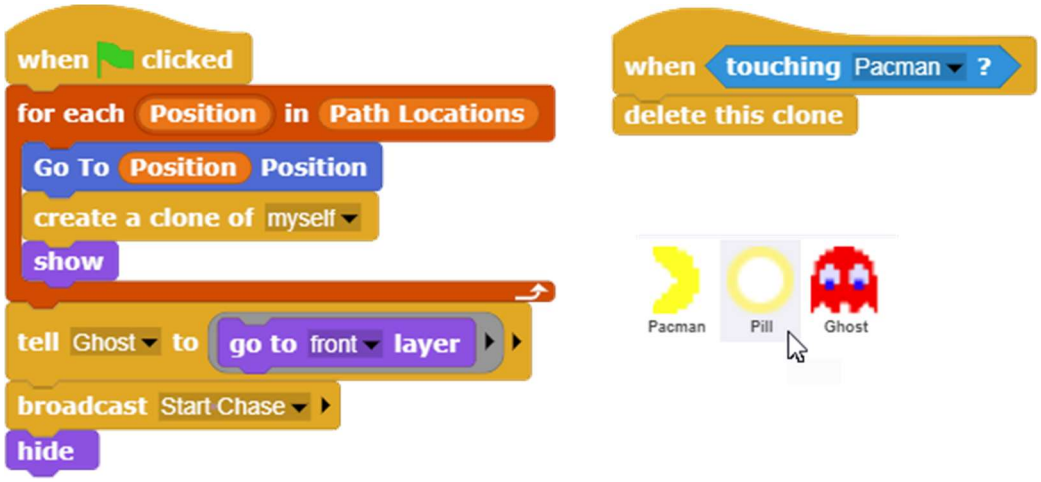


The direction of travel is controlled by the arrow keys on the computer keyboard. Before advancing, the character checks to see if the next square on the grid is a valid location:

The **Advance?** block reports whether the next position is a member of the list of **Path Locations**:

If the next position is a valid location on the path, the Pacman advances. The Advance procedure rotates through three costumes (Pacman mouth open, closing, and closed) as the sprite moves forward 25 steps and consumes a power pill in the process. One iteration of the "chomping" sound that accompanies the movement is also played.

The power pills are placed on the grid by creating clones of a white circle on every valid location on the path. Each clone has a script that deletes the clone when it is touched by the Pacman.



The ghost chasing the Pacman checks each of three possible moves (left, right, and forward) to see which ones are valid moves. It then selects the move that will be closest to the Pacman, and repeats the process until it touches the Pacman.

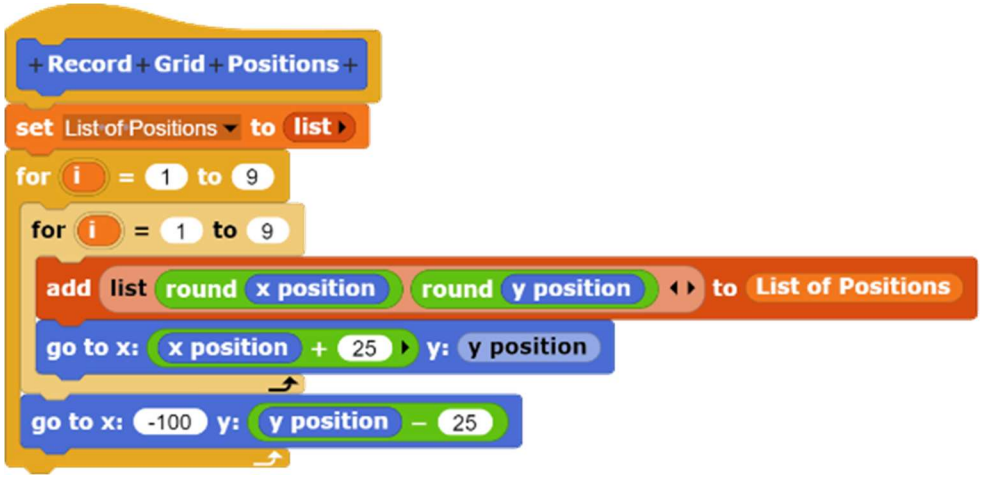
The game ends if the ghost touches the Pacman. If the Pacman successfully eats all of the power pills on the game board without being touched by the ghost, the game restarts for another round.

Building the Maze

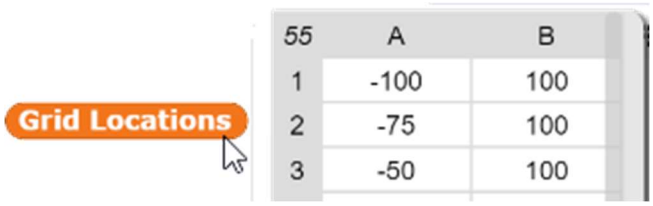
The code used to build the maze is still in the file.

A procedure, **Build Game Board**, uses two subprocedures, **Box** and **Walls**, to draw the game board. The **Box** procedure draws a box around the game board. The **Walls** procedure draws the walls within the box.

A procedure, **Record Grid Positions**, was used to record a list of all the positions on an 9 x 9 grid spaced 25 steps apart.



The result was saved in a reporter block, **Grid Locations**:



The **Wall Locations** were then removed from the **Grid Locations** to create a reporter block, **Path Locations**, with a list of all of the valid path locations:

The Path Locations reporter block is used to determine where the Pacman is allowed to travel.

An extension might generate different mazes using an algorithm rather than manually.