**PAC-MAN**

**Main features**

* set maze for each level
* player controls pac-man
* 4 ghosts that chase pac-man
* each empty spot has food in it at start of game that pac-man eats as he passes
* energizers in set places that make ghosts flash, edible by pac-man and run away from pac-man for a set time
* player wins if all food is eaten
* player loses if caught by ghost

**Classes that can be on grid cells**

1. Path
   1. Food
   2. Energizer
   3. Player
   4. Ghost
      1. Frightened: true/false
   5. Empty
   6. Intersection cell (not mutually exclusive to any of above)
2. Wall

**Variables**

1. Score, starts at 0
2. Lives, starts at 3
3. Board array: for each cell designates what kind of class it is at the start – when adding more levels, can have several boards
4. Ghost position array: [current position, previous position] – to keep track of previous cell
5. Ghost options array: all cells around the ghost

**MVP (aim for Tuesday)**

* One level, one ghost
* Colour-coded dots to designate player and ghosts

START BUTTON

* event listener 'click', runs function that contains everything of the game

GENERATE GRID

* generate cells, place them in an array
* designates wall, food, energizer cells
* designates initial player and ghost cells

GHOST MOVEMENT

* start in set positions and in set direction
* movement on setInterval *(figure out timing based on what speed feels right difficulty)*
* continue in direction of movement until there is an intersection
  + looping through forEach cells in ghost options, find cell that is only option available (given can’t go back to previous cell)
* at intersection move to option closest to pac-man
  + looping through forEach cells in ghost options, check distance to pac-man *(figure out maths of that)*
  + go to option that is closest
* after each movement, update ghost position and ghost options, check if player is on current cell
  + GHOST HITS PLAYER (does player stay in same spot, and where do ghosts go?)
    - Remove one from lives
    - check if lives === 0, then game-over, display score
    - Everyone goes back to starting positions

GHOST-FRIGHTENED MOVEMENT

* immediately when changed, move back to previous cell in ghost position
* otherwise all same as normal ghost, but move to cell that is furthest away at intersection
* after each movement, update ghost position and ghost options, check if player is on current cell
  + PLAYER EATS FRIGHTENED GHOST
    - add 100 to score
    - make ghost go back to spawning position (set class back on that original cell)

PLAYER MOVEMENT

* event listener on document for key-up WASD *(figure out maths for movement depending on size of grid)*
* check what kind of cell the player moved towards and do different things depending on that
* PLAYER TRIED TO MOVE INTO WALL
  + do nothing
* PLAYER MOVED ON EMPTY CELL
  + just move there
* PLAYER EATS FOOD
  + add one to score
  + make cell empty
  + if score is XXX game ends (because set number of food on board)
* PLAYER EATS ENERGIZER (Boolean true/false on frightened)
  + change ghosts to ghost-frightened for setTimeout 30 seconds
* PLAYER EATS FRIGHTENED GHOST
  + add 100 to score
  + make ghost go ghost pen
* GHOST HITS PLAYER
  + Remove one from lives
  + check if lives === 0, then game-over, display score
  + Player goes back to starting positions

TO DO (in order)

1. Create movement function that takes into account the edges of the boards
2. Create PLAYER HITS GHOST / GHOST HITS PLAYER mechanism
3. Create Ghost chasing player algorithm
4. Create Ghost pen cell class
5. Create PLAYER HITS ENERGISER mechanism
6. Create PLAYER HITS FIGHTENED GHOST / FRIGHTENED GHOST HITS PLAYER mechanism