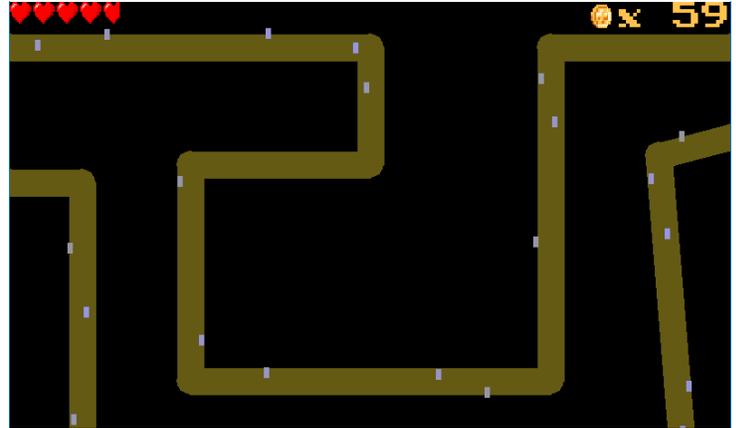


1. Application class

- a. (-5 for bad structure, code that won't run)
- b. Attributes (include at least these):
 - i. **mAttackers**: a list of Dudes (below)
 - ii. **mPathSize**: the width of the path
 - iii. **mClock**: a pygame clock object
 - iv. **mPaths**: a list of points (the first element is the spawn point, the last is the point at which the attacker causes damage) – Note the bottom path in the video is using a bonus feature.
- c. Methods:
 - i. (4 points) constructor: see the main program for what arguments will be passed to this method.
 - ii. (1 point) **isDone(self)**: return True if the user has indicated they want to quit (by pressing escape or the close button)
 - iii. (1 point) **createLane(self, path)**: take a list of points from the caller and make a new path.
 - iv. (4 points) **processEvents(self)**: handle all input here (and only here).
 - v. (8 points) **update(self)**:
 1. calculate the elapsed time
 2. move all attackers (calling their method) if the game isn't over
 3. remove attackers that have reached the end of the path (and remove a fractional amount of health).
 4. spawn new attackers every few seconds (make this rate frame-rate independent) if the game isn't over.
 - vi. (5 points) **render(self, surf)**:
 1. draw the paths (as a line with width `self.mPathSize`) with circles at the junction points.
 2. draw the attackers
 3. if the game isn't over, draw a health bar using a single heart image (see the video for an example). Allow fractions of a heart.
 4. if the game is over, display a game-over message.



2. Dude class

- a. Attributes (include at least these):
 - i. **mPos**: the current position
 - ii. **mSpeed**: the speed (in pixels / s) that the dude moves.
 - iii. **mPath**: the path this dude is following
 - iv. **mColor**: make each dude slightly different colors.
- b. Methods (include at least these):
 - i. (4 points) constructor: should take only a path list (and a path-width if doing the bonus)
 - ii. (10 points) **update(self, dt)**: move the dude along the path
 - iii. (2 points) **render(self, surf)**: draw the dude.
 - iv. (1 point) **finished(self)**: return True if the dude has reached the end of the path

3. Optional Bonus Features

- a. Note: I'm planning to make extending this lab an option for our last lab (Lab9) so you may want to ask before attempting bonus features not listed here.
 - b. (5 points) "Jitter" the dudes (i.e. don't make them all walk directly along the path; make them a few pixels above / left / right / down [at random] around the center of the line).
 - c. (4 points) Draw a gold counter that gradually increases over time and a spinny-coin icon like seen in the video.
 - d. (8 points) Allow path points to be in any position (the bottom path in the video illustrates this)
4. Sample video: <https://youtu.be/grGN3ATNmzg>