Samantha Staheli

11-22-21

CSE 210 final project

## **Preparation and Planning**

### Decide on the game or interactive simulation you want to create. It's ok if you change the details a little over the next few weeks, but you should have the main idea in place.

**The Office Adventure**

A game based on the Atari adventure game with an office theme. Instead of the character being a square it will be an office character. Instead of finding keys the user will have to find an object that correlates with the character they are using. To win the user has to find all the objects.

Levels:

1. Pam: collect plants, answer phone (first level because just need to collect things)
2. Jim: prank Dwight, propose to Pam
3. Angela: save bandit
4. Michael level: Michael Scott paper company, holly sleeve of cardigan, run away from Jan, save Stanley
5. Dwight level: has to defeat certain things. Roy with pepper spray, Jim with snowballs

### Create a list of the top priority ("must have") requirements for your program to be usable/playable.

1. 1 level
2. Actors:
   1. 1 character from Office
   2. Object to collect
   3. Inventory to show collected objects
3. Arrow controls move character
4. Once character is near object. The object gets collected to inventory
5. Collisions with walls makes actor not able to move through walls
6. World seen on whole screen

### Create a list of the lower priority ("nice to have") requirements that could take your program to the next level.

1. Explore world/not all of world shows on screen at once
2. Gif shows on screen before level starts
3. The Office set layout
4. Multiple levels
5. Main menu
   1. Start
6. Sub menu
   1. exit game
   2. restart game
   3. change level
7. Once character is near object message shows up that user can press g key to grab object (g for grab)

### Create a list of all the classes you anticipate in your program. For each class, please state the things that class is responsible for.

1. Basic classes for functionality or parent classes that are like batter programs:
   1. Director
      1. imports constants
      2. starts game
      3. cues actions
   2. Actor
      1. keeps track of visible things in the game
      2. keeps track of their position, size, image path, etc.
   3. Action
      1. parent class to classes that do an action
      2. execute() is its abstract function
   4. Point
      1. Represents distance from an origin (0, 0)
      2. parent class to input\_service
   5. Constants
      1. Not a class but has games basic variables.
2. Basic child classes like batter programs:
   1. Output service
      1. draws the games state in terminal
   2. Input service
      1. gets which direction user wants to go
      2. gets keys pressed
   3. Physics service
      1. child to Actor
      2. responsible for actions that are movements (physics related)
      3. looks for collisions
   4. Audio service
      1. plays sounds using pyray.play\_sound()
3. Other child classes for more customization of game:
   1. Character
      1. A class that controls and prepares the character that is used and uses the characteristics defined in the cast constants.
   2. Object
      1. A class that controls and prepares objects that go with what character is used.
      2. It will also import cast constants.
   3. CastConstants
      1. Not a class but will contain the specific characteristics for each character that the character class will use.