CS246E—Assignment 4, Project (Fall 2021)

B. Lushman

Due Date 1: Friday, November 27, 5pm
Due Date 2: Tuesday, December 15, 11:59pm

ARLG

One of the two games you are required to make is called ARLG. ARLG's main gameplay loop is centered around room based combat. There are six levels in the game: to beat the game in its entirety one must beat all the enemies in the first five levels and then defeat the final boss in the sixth level. Beating the game displays a credits screen with your name, your group partner's name, and (optionally), your Twitch VIP statuses. Running out of health displays a failure screen informing the player that they lost. Any input to the failure screen will result in the program terminating.

An Example ARLG screen

```
+------------------------------------------------------------------------------+
|                                                                              |
|                                                  X                           |
|        X                0                                                    |
|                                                                              |
|                                                                    x         |
|                         0                                                    |
|                              X                Zzzz                           |
|              aa                                                              |
|              aa         0                            e                       |
|                                                                              |
|                                                                              |
|     0 x                 0                                    zzzZ            |
|                                       x             bb                       |
|                                                     bb                       |
|0    D    0              p                                                    |
|                                                                              |
|                                      Z                                       |
|     0                                z                x                      |
|      h                          c    z                                       |
|                                      z                                       |
+------------------------------------------------------------------------------+
Level 5/6
Health: 5
```

Status Bar 3

Entities

Player

The player character is represented with the character p. The player character can move around using the arrow keys, and shoot up, left, down, and right with the W, A, S, and D keys respectively. The projectiles shot by the player will appear as O characters, which travel in the direction they are shot in until they collide with an another entity at which point the projectile is destroyed. If a projectile collides with an entity that has health, the entity should take 1 hit of damage. The player starts with 5 health, and their health state is always reflected within the first line of the status bar. If a player walks into an enemy the two entities should bounce off of each other with the player taking 1 hit of damage.

Fire

A fire is represented by an x; the fire should animate between lowercase and uppercase. If an entity with health walks into a fire the entity should stop and take 1 hit of damage. Fires can be put out by shooting them with projectiles, whereupon they have a chance to drop a health pickup.

Health Pickups

A health pickup is represented by a lowercase h. If an entity with health walks into a health pickup, the health pickup should disappear and the entity's health should be increased by one.

Exit

The level exit should be represented with an e/E. If there are enemies on the screen, the exit should appear in lowercase and at a lower height than all other entities. Once all enemies for the level have been cleared, the exit should appear in uppercase. If the player walks into the exit when the level is cleared, then the next level is loaded.

Enemies

Walkers

Walkers are represented by a 2x2 box of the character a. Walkers have a basic movement pattern in which they choose a random cardinal direction to walk in and proceed in that direction for a few ticks. They then choose a new direction and repeat the process over again. Walkers start with three health by default.

Stalkers

Stalkers are represented by a 2x2 box of the character b. Stalkers always move towards the player (note that if a fire is in the way, they do not need to pathfind around it, and may just take repeated damage as a result). Stalkers start with two health by default. When a stalker is killed, it has a 25% chance to produce a walker in its place.

Pop-Ups

Pop-ups are stationary enemies with an invulnerable and vulnerable state. If a pop-up is in its invulnerable state then it cannot be damaged by projectiles. and it is represented with the character c. If a pop-up is in its vulnerable state then it shoots projectiles in all four cardinal directions and is represented by the character D. Consider using ncurses color options to distinguish between player projectiles and enemy projectiles. Pop-ups should alternate between their vulnerable and invulnerable state on a frequent but semi-random basis. Pop-ups have four health by default.

Snakes

Snakes are represented by a 4x1 box, with the uppercase character Z representing the snake's head, and the three lowercase zs representing the snake's tail. Snakes should move in a straight line path, turning around when they reach the solid border area. Snake heads are invulnerable to projectiles; one can only damage a snake by attacking its tail. Snakes have three health by default.

Level Generation

Each level should place the player and the exit in a random position on the game board. A random number of fires should be generated depending on the desired difficulty of the game. Finally, depending on the level number, somewhere between 5 and 10 of the following enemy types must be placed on the game board:

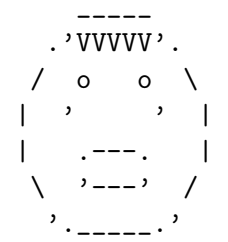
1. Walkers
2. Walkers and Stalkers
3. Walkers and Pop-Ups
4. Stalkers and Snakes
5. All enemies

The Boss

The Boss of the game appears in level 6. Its movement pattern should be in a diagonal line, bouncing when it collides with wall boundaries à la https://www.bouncingdvdlogo.com. The boss's health should be displayed in the third line of the status bar, with its peaceful phase starting at 20 health. During the peaceful phase, the boss has the following appearance:

```
    -----
  .'     '.
 /  o   o  \
|           |
|  \     /  |
 \  '---'  /
  '.-----.'
```

While the peaceful phase is ongoing the boss should intermittingly spawn walkers and stalkers. Once the boss has lost all of its health once, its health bar refills to 30 health, and the boss enters its rage phase. During the rage phase, the boss has the following appearance:



The boss should intermittingly spawn all four enemy types throughout the rage phase. Additionally, the boss should continue its movement pattern but at an increased speed. Slaying the final boss in its rage phase causes a win condition and should launch the credits screen.

Command Line Arguments

ARLG takes one optional command line argument, which is a number from one to six. The command line argument allows a player to skip to a certain level without playing the prior onces - for example, one could run ARLG with the command line six to skip straight to the final boss.

Game Two

The second game is up to you! You should ensure that your second game is of a similar level of complexity to ARLG. You may also want to consider trying a different genre of video game to demonstrate the power and versatility of your AGE implementation. For example,

- Programming a game like Mario could demonstrate your engines handling of gravity.
- A pinball simulation could demonstrate the complexity of collision interactions.
- Files could be used to save and load a game mid execution.
- ARLG does not make extensive use of AGE's height mechanic so one could design a puzzle game around height.

The possibilities are endless and you are encouraged to be creative and apply what you have learned about object oriented design.