Entity Classes

`abstract class IndividualCharacter`

Responsibilities

- Health: Attribute, int, denotes the total amount of health the individual has
- Position: Attribute, array with x, y, where on the map the IC is currently located
- Inventory: Attribute, ArrayList, items currently being held
- Level(?): Attribute, the level of the enemy
- Range(?): Attribute, how far the character can attack
- Hitbox(?): Attribute, from how far the character can be hit from
- updatePosition: Method, updates the x, y position coordinates
- depleteHealth: Method, updates health after being attacked

Collaborators

Player, Enemy, Defender

`class Player`

Responsibilities

- Spawnable_defenders: Attribute, dictionary of level of defenders and their corresponding integer of number of defenders the player can spawn.
- Build_wall(position): Method to use inventory to build wall at a certain position on the map
- Spawn_defender(position): Method to spawn defender at a certain position on the map

Collaborators

Player, Enemy, Defender, Weaponable, Collidable, IndividualCharacter `class Enemy`

Responsibilities

- Path: Attribute, ArrayList, the path the enemy is meant to follow
- Attack: Method, attacks defense on the map

Collaborators

IndividualCharacter, Player, Defender, Map, Weaponable, Collidable

`class Defender`

Responsibilities

- Attack: Method, attacks enemies on the map

Collaborators

IndividualCharacter, Player, Enemy, Map, Weaponable, Collidable

`class Tower (?)`

Responsibilities

- Position: Attribute, array with x, y, where on the map the Tower is currently located
- Health: Attribute, total health of the tower
- updateHealth: method, updates the health of the tower to reflect damage taken

Collaborators

Enemy, Map, Collidable

`class Map`

Responsibilities

- All_positions: Attribute, list of all objects and their positions

Collaborators

Player, Enemy, Defender, Tower

`interface Weaponable`

Responsibilities

- Hit(attacker, victim, weapon): Method that depletes victim health by weapon’s attackPower

Collaborators

Player, Enemy, Defender

`interface Collidable`

Responsibilities

- Collide: Method that collides two objects whenever in the same position and depletes health according to objects clashed.

Collaborators

Player, Enemy, Defender, Tower

`abstract class Weapon`

Responsibilities

- Grade: Attribute, essentially the level of the weapon
- attackPower: The amount of damage the weapon does

Collaborators

Player, Enemy, Defender, Collidable

Use Case Classes

`class CharacterManager`

Responsibilities

- moveCharacter: method, takes directional commands from keyboard input and moves character
- attackObject: method, attacks nearest object when the range <= distance to that object (for NPCs)
- takeDamage/depleteHealth: method, takes health away from the character when hit

Collaborators

Player, Enemy, Defender, Collidable

`class gameManager (?)`

Responsibilities

- Spawn: Method, spawns enemies at position on the map
- deleteCharacter: removes a character from the map when health = 0

Collaborators

Player, Enemy, Defender, Map

Controller Classes

`class InputHandler`

Responsibilities

- - keyLeft(input), move player left
- - keyRight(input), move player right
- - keyDown(input), move player down
- - keyUp(input), move player up
- - keyOpenInventory, browse inventory
- - keyChooseInventoryItem, pick inventory item
- - keyPlaceItem(input), place inventory item
- - keyLevelUpDefender(?), level up the defender
- - keyAttack(input), attack with weapon

Collaborators

Player, CharacterManager

`class SpawnController`

Responsibilities

- spawnLocation, attribute, (x,y) coordinate, different for player and enemy
- spawn(spawnLocation), method, spawn player/enemy at spawn location

Collaborators

Player, Enemy, Map

Other Classes

`class MainMenu`

Responsibilities

- startButton, attribute, position of start button
- helpButton, attribute, position of help button
- quitButton, attribute, position of quit button
- start_game, method, start game by generating map, spawning player (w/ SpawnController?)

Collaborators

Player, Map