Justin Friends

CSCI 165

Design Doc

I chose abstraction for two reasons, one is that I’m more familiar with it, and two I think it fits what I’m trying to do better than an interface, based off of this short description: <https://www.guru99.com/interface-vs-abstract-class-java.html>. It’ll allow me to set up a base template from which to create new classes rather easily, needing only to change their move, attack, and observe methods, as well as different character stats.

For class ideas, I have in mind Mario, Luigi, potentially a couple more common main characters that have been brought out in games, as well as Goombas, Koopa Troopas, and Piranha Plants. I don’t have hard numbers in mind yet, depending on the simulation size and how they actually end up playing out once implemented, but essentially:

‘Turns’ will be every second or two or three (not sure if one second is too fast to watch things play out, or really what’s too slow either, and potentially this could be an option via command line argument or something).

Currently, I’m thinking only one of each ‘hero’ character, with an undetermined number of each enemy type. Basically, each Hero character will go towards an enemy to try and attack them unless their health falls below 25% or something, at which point they’ll prioritize searching for a healing item. Each ‘enemy’ type will try to move away from the Hero characters, unless they observe 3 other enemies in their neighborhood, at which point they will move in to attack, or if the Hero character has health below 25% or something. I’d also like to add items that affect this behavior as well (potentially a power star, making the holder invincible and deal more damage for maybe 5 or 10 turns, as well as mushrooms that restore health). ~~Maybe also add a Boolean flag~~ I added a Boolean flag so that if it’s on, other enemies in the neighborhood will then move towards that enemy, until a Hero is detected within their range, at which point they will move in to attack (provided they [still] have enough enemies around).

For color and health, I’m currently thinking a 3 int array for the RGB colors, with each class having its own color, and then the brighter it is means the healthier it is, and the darker it is means the less health it has, which can be a gradient based on its current health compared with its total health. I’m not sure about prioritizing attacks and checking each creature for lowest health and the like or not yet.

I’m not entirely sure how to implement all of this with the drawer map. I’m currently thinking to keep all the creatures in an array, and then adding a Boolean flag for alive or not, and upon 0 health turn it to false and then moving the location to something ‘out of bounds’, since it won’t be checked to move if the alive flag is false, and will avoid being detected in others’ observe checks.

I was thinking something similar for the items, but have a couple questions about how to prevent items from occupying the same space (if using random generation). I’m not entirely sure how I want to do this, but I think having a second item class that extends the prior (either Weapon, Armor, or Item – I’m not sure I’ll include Weapons and Armor in a first pass actually), with location properties added. And then once picked up or equipped, it can be upcast to the parent class which does not have a location anymore, but is still available to be equipped.

If enough time has passed, I would like to seed more items and perhaps monsters into the world as well, based on various states that could be checked, but again that probably won’t make the first cut.

Currently, I have an arraylist of type Creature (which is extended in all of the character classes) in FinalProjectDriver that just has one of each character, and then a for each loop that calls the three abstract methods in Creature on each object. Right now they just print out that the method was called, but will have to be where all the logic eventually goes. Perhaps eventually I can make it a command line parameter and whatever number gets entered for each creature can then be a variable for a loop and iterate it that many times to be added to the array list to be populated on a location.

**Mario**:

Health: 100

Move: Up to 4 spaces a turn.

Attack: Can ‘jump’ on enemies to attack, up to 2 spaces away, moving location afterwards next to the attacked creature. Deals 30 damage.

Observe: Range 4 spaces.

**Luigi:**

Health: 100

Move: Up to 2 spaces a turn.

Attack: Can ‘jump’ on enemies to attack, up to 5 spaces away, moving location afterwards next to the attacked creature. Deals 20 damage.

Observe: Range 5 spaces.

**Toad:**

Health:

Move:

Attack:

Observe:

**Peach:**

Health:

Move:

Attack:

Observe:

**Goomba:**

Health: 50

Move: Up to 3 spaces a turn.

Attack: Must be within one move space to attack. Deals 10 damage.

Observe: Range 3 spaces.

**Koopa Troopa:**

Health: 30

Move: Up to 5 spaces.

Attack: Attack up to 2 spaces away. Deals 10 damage.

Observe: Range 8 spaces.

**Piranha Plant:**

Health: 25

Move: Up to 1 space per turn.

Attack: Shoot a fireball up to 6 spaces away. Deals 15 damage.

Observe: Range 6 spaces.

**Items:**

Power Star: Become immune to damage, double move distance, and double attack power.

Mushroom: Heals up to 50 damage (total starting life can’t be exceeded).

**NEED TO IMPLEMENT**:

Mario:

Observe() method

Item unequip to reset stats (use itemUnequip() once item duration has run out).

FIX BUG MARIO LINE 106 Index 0 out of bounds for length 0 – done.

Add check for pickedupflag() in proximity itemworldcheck.