CS4303 final game report 160015090 20th.Dec.2019

Overview

This practical becomes chaotic while adding more and more features. As processing doesn't really support packaging, it makes the structure complex and very difficult to check the problems encountered. Most of the features are implemented and I will discuss the details of what I encounter and what I managed to implement. To run the game run game.pde.

Play guide & game design

Background: The game is a rouge game with a set of randomly linked rooms filled with objects and enemies. The player needs to find an emblem to defeat the boss. When a player died, 50% of possessions will loss and 50& chance weapon and cultivation method will lose (random a new one). Then continue game with new generation.

Control: The player is control by w,a,s,d to move around. It attacks towards the place of mouse is placed when mouse right button is clicked. The attacks can be blocked by objects.

Attributes: A Player/enemy has HP, MP, shield, physic attack (AD) and magic attack (AP). HP, MP, shield are default 100 and don't self recover.

Race: A player has a race. Human has more usable (potion) when start game, Wilding deals more AD but less AP, has more HP but less MP, and weaker shield. Spirit is the opposite of wilding. The player choose race for each generation.

Bag: A player will have a bag with three usable slots which storing hp potions, mp potions and bandages. Hp potion restores 20hp and mp potion restores 20mp, bandage will remove bleeding effect. Use 1,2,3 to use an item. A bag has 5 weapon slot and 3 cultivation method slot. When opening a bag, a player can switch weapon or method, this part not implemented.

Profession & weapon: player/enemy has a profession. Different professions have different attack mode. This practical has archer which shot arrow with bow, mage shot magic balls with wand and warrior blast (called swing, a circle area) in front of player location with hammer. Each profession cannot equip other professions' weapon. Bow has a range of 150, wand 200 and its bullets can land anywhere in range, hammer has a range of 30 and blast range of 35. Only hammer can one shot objects (non-enemy). Player can choose profession or 50% inherits from father. Player cannot go bare hand (not implemented, see bag)

Cultivation method: See Wikipedia "wuxing" for rules and information. It has AP and shield boost and decide player's attack and shield type. Player can switch element type with mouse wheel down. Switching only goes along element generation rule.

Attack: Each profession has a frequency of attack (although not obvious). The type of attack is same is current element of cultivation method. Attack has same colour as current type. An attack can be enchanted (press q to trigger), which add AP damage when having MP. 2 MP per attack.

Shield: The player has a shield which strength is affected by race and cultivation method. It can be charged by pressing c when there is MP left. The type of shield is same is current element of cultivation method. Current shield type is represented by the stroke colour. When shield breaks, attack can cause bleeding.

Room: Rooms has random size but always appear at center of screen. They are randomly connected to each other. Except for start room and end room, rooms have one single statue with can cause bleed, reveal hidden room(not implemented), give permanent buff, give emblem or a cultivation method. At most two chests containing usable (can be different) and a chance to drop a weapon, multiple crates which drops usable (same type) with 50% chance and multiple obstacles. Except for statue, objects can be destroyed (and drop no loot). A start room is where the player starts a new map. When player come back to start room, a game is finished map will regenerate. The gate at the center of start room land player at random position of random room. On the left of the gate is the boss statue. It enables boss fight by applying an emblem(not implemented). If boss fight enabled, start room gate connects to boss room. The end room is where player leave the map and back to start room. It is randomly attached to a room and only has a gate in it (land in start room). A room can have a element which in this case all enemies in this room is of that type.

Enemies: They are concerned to be floating, so they pass objects and attack can't be blocked by them. They have a sensing radius of 250. When sensing the player, they speed up into the attack range and slow down and attack player on arrive. They move randomly in attack range to mimic an action of avoiding player attack.

Map: The player can view a map using key m. A map shows the connections between room and player can place rooms in a style they like by dragging them on the map. Current room is highlight by red colour.

Buff: Player can get (de)buff. This practical only has a bleeding effect which last about 30 seconds and take 1 HP per 2 seconds. There are a bunch of permanent buffs that can be pass down to children.

Interact: Use key e to interact with objects, the current interactable is highlight by purple edge.

Teleport: a portal is a glow blue circle in room. When step on it, press tab to go to next room. Can come back use the portal player land on. It should be disabled when room has enemies(since I have problem with hit checking, I did not do this)

Implementation

Elements: create objects of the five elements storing its colour and rules. Only one of each exists in the game. They are pass around when needed. Use an array to store the elements for random an element (by random array index).

Moving of player: use a boolean array to map a hot key set. When opposite direction is press, player won't move. The player won't move if a direction pressed collide object.

Profession, Weapons, Attacks: Each weapon inherits abstract class Weapon giving them different way of attacking and hit checking. Since each profession must have an weapon, when a player attack, it is actually the weapon under the profession firing a bullet (abstract class for different attacts). What a profession does is checking if the attack is enchanted, forcing a frequency of attack using a lock which lock when attack and unlock after some time. This can be achieved by tracking frame counts and frame rate. 1 frame rate is 1 second. I did not use frameCount % framerate because it is not producing the correct value (never get a 0 or an integer). A weapon keeps track of and mov all bullets it fired, so most of its methods loop through the bullets. A bullet is added to weapon when attack and remove when hit or reach range. How bullet directions and positions are calculate is well commented in class arrow.

InterActables: statue, chest, crates, gates inherits interActable. A player can call the interact method to achieve some action. An object has a max it count where decrease by 1 each time it get hit. Meaning that it takes 1 damage at a time. It use a Boolean opened to indicate whether interaction is available (already interact?) There will be magic numbers when generating random stuff but they are explained. It will be better if they are wrapped up in some way but I can't think of a good way to do it. A statue doesn't take damage so there is nothing in take damage function. An obstacle is an interactable but faking not because opened is always true (unavailable).

Potions: inherits potion, each have a consume method, either adding attributes all add/remove effect (buff). The code only implemented hp, mp and bleeding changes.

Buff: play has a list of buff which contains all positive and negative effects. The effect will be apply when the time lock unlock (see buff bleeding in player guide). There is a list of permanent buff which can be passed down to the next generation. These buffs will only be apply once when creating a new player, except from buffs of attacking only take account when dealing damages.

Enemy actions: when player is not is sense range, it will wander around. Otherwise, it calculate a rectangle using the attack range and player position to get an area which it intended to move in (although at corners, it is out of attack range). It picks a random point in that area to move to and attack if in range. A variable turn chance is using to make enemy more likely to turn when moving in a direction for some time. Details see code comment.

Damage counting: There are two types of damage AD and AP. If attack is not enchanted, then only count AD damage. Otherwise total damage will be AD + AP and apply "wuxing" rules. If the element type (decided by cultivation method) overcome the enemy shield, double the damage, if it generates, enemy shield generates the attack amount ,others no change. If weapon type generates the attack element, double the effect of weapon's element strength, otherwise, deduct it. Damage will first dealt to shield, when shield breaks, it takes the excessive damage of that attack. When switch weapon or cultivation method, damage recalculate using calPhysicDamage and calMagicDamage.

Room: a player has a play room. It keeps tracks of all enemies and objects and feed player necessary information such as target object for interaction.

Testing & evaluation

Because of the nature of processing, it becomes hard to do flow control, most of the features I did not implemented is because of this. In bag, map, playground and choosing interface, it all requires different interactions of keys. This made the interaction such as keyPressed() rather complicated and very easy to break if adding new features (I attempt add switch items in bag, but failed as it gets long). Game flow is another problem, I need lots of flags to see which state of the game I am in. I even setup a wrap class called newGeneration which bunch of Boolean in it just to keep track of which step I am at when creating a player, fortunately, it works unlike boss fight, which I failed add to the game flow.

Another problem arise form huge and complex structure is the problem shooting. I have some major problem on detecting if player hit enemy or vice versa. I am confident about them taking damage in a intended way, but I am sure something is incorrect of hit detection especially for mage and warrior enemies to detect hit player. If lucky enough to have a single enemy in room, I can see the mechanism is mostly correct, but there are many enemies, it is hard for me to see if this is the case. More Explained in videos.

Structure

Since processing is incapable in package. Here is a rough structure of game

- Game
- abstract race
    - human
    - wildling
    - spirit
- Abstract profession
    - Archer
    - mage
    - warrior
- abstract weapon
    - bow
    - wand
    - hammer
- abstract bullet
    - arrow
    - magic ball
    - swing
- abstract potion
    - hp potion
    - mp potion
    - bandage
- abstract buff
    - bleeding
- interactables
    - statue
    - chest
    - crate
    - obstacle
    - gate
    - bossStatue
- movingobjects
    - player
    - enemy

Some other wrap up class such as hitpoint manapoint and utility classes of collision checking.

Comparing the existing game

Comparing game “Lovecrafs untold stories”. There are some major difference and similarities::

1. The introduction of “wuxing rules” and the shield system. This is the most outstanding feature of the game and entirely new to the game.
2. The next generation feature. Although the game has similar mechanism, it does not pass down items when player died.
3. My game requires piece of emblem to enable boss fight as well
4. The map system. My game allow player to set up map as they wanted.
5. My game does not have shops.
6. The start room mechanism is new to “Lovecrafs untold stories”
7. Objects are similar
8. The enemies actions. In my game, enemies random move in attack range, while in “Lovecrafs untold stories” they constantly chase player.

Conclusion

Despite the major problems I encountered, I managed to implement the main features and main flow of the game. If the structure is less complex, I am confident that I can solve these problems. If processing support packaging, the structure will be more organised and easier for locating the main problems.