<u>**Homework III Report**</u>

1. Assignment Details:

Goal: Refactor and Review project

Development team: CMPT 276 - Group16 - Sibei Zhou & Rongsheng Qian

Team Members:

    Rongsheng Qian, Sibei Zhou

Development Tools:

    intelliJ IDE with JDK, GitLab, Maven, Junit

Assignment 3 Deadline: April 11th, 2022

2. Parts of the code which is reviewed:

Class GameObject; Class DynamicCharacter; Class StaticChararcter; Class MainCharacter; Class Zombies; Class KindSurvivor; Class BadSurvivor; Class Reward; Class Food; Class Vaccine;

3. The smells we have identified & refactored code:

Fix bug which is caused in code review: `64e057c7 f68d211f c7bb196a`

3.1 unused or useless variables `9cabb41b`

<u>**Delete useless variables:**</u> "private int damage" in Class Zombie; "private int HP_increase" in Class Food; "private String message" in Class StaticCharacter; "private GameMap gm" in Class GameObject;

3.2 code duplication `3b06e009`

1) There are many variables that already declared in abstract class but still declared in subclasses should be deleted:
    a. Delete "private BufferImage up1, up2, up3, up4, down1 down2, down3, down4, left1, left2, left3, left4, right1, right2, right3, right4;" in Class DynamicCharacter;
    b. Delete "private BufferedImage up, left, right, down, message_image, message_image_incomplete;" in Class StaticCharacter;
    c. Add an array of BufferImage "private BufferImage[] images;" to store all the images needed in Class GameObject instead of them.

3.3 low cohesion

1) Change Class DynamicCharacter, StaticCharacter to extend the abstract Class GameObject to avoid their instantiation. `9cabb41b`
2) After doing so, we also removed a large number of variables that had the same function from Class DynamicCharacter and Class StaticCharacter and merged them into this GameObject. (Such as variables x, y, width, height, gf, status, direction, collision) `9cabb41b`
3) There are lots of similar functions in subclass KindSurvivor and BadSurvivor, and we add a function as the same name in their superclass StaticCharacter and add the identical codes in it, and then change the same codes in original functions to super(): `3b06e009`
   a. Add a function named getSurvivorImages() in superclass StaticCharacter;
   b. Move the similar codes in getSurvivorImages() of subclass KindSurvivor and BadSurvivor to superclass's getSurvivorImages()
   c. Change getSurvivorImages() in both of KindSurvivor and BadSurvivor to @Override and add "super.getSurvivorImages();" and leave the different codes behind call the superclass's function.

3.4 poorly structured code

1) Make an abstract function "draw(Graphics2D g2)" in Class GameObject and override it in each subclasses: `cc94018f`
   a. Change drawZombie(Graphics2D g2) in Class Zombies to @Override draw(Graphics2D g2);
   b. Change drawKindSurvivor(Graphics2D g2) in Class KindSurvivor to @Override draw(Graphics2D g2);
   c. Change drawBadSurvivor(Graphics2D g2) in Class BadSurvivor to @Override draw(Graphics2D g2);
   d. Change draw_MC(Graphics2D g2) in Class MainCharacter to @Override draw(Graphics2D g2);
2) Make an abstract function "update()" in Class GameObject and override it in each subclasses: `3b06e009`
   a. Change updateZombie() in Class Zombies to @Override update();
   b. Change speak() in Class KindSurvivor to @Override update();
   c. Change speak() in Class BadSurvivor to @Override update();
   d. Change updateMC() in Class MainCharacter to @Override update();
3) Delete functions "check(Rectangle mc)" in Class Zombies and add a function "check(Rectangle mcHitArea)" in abstract Class GameObject to replace all original check functions in each child Classes. `cc94018f`

3.5 confusing class hierarchy `3b06e009`

1) Change Class DynamicCharacter, StaticCharacter and Reward to abstract class, and must be instantiated by its subclasses (subclasses MainCharacter and Zombie for Class DynamicCharacter, subclasses KindSurvivor and BadSurvivor for Class StaticCharacter).