

# INFO1113 – Gremlins Report & UML diagrams

Yangqing Zheng  
520482401

The design the Gremlins game contains three public classes in the top hierarchy, which are App.java, Frame.java and GameObject.java. App.java is the main class that extends PApplet and responsible for the game to run. Frame.java is responsible for parsing the map from json file and initialize all the game objects for a specific level. GameObject.java is an abstract public class that contains two types of GameObject class (StaticGameObject and MoveGameObject). All the type of game objects is classified to this two abstract GameObject class.

## **Inheritance:**

The reason for creating a hierarchy for game objects is because it allows same attributes and methods to be inherited from the parent class, which is prevent from repeating a large chunk of codes with same logic and making the design from organized. For instance, intersection method can be put directly on GameObject class, so this method can be shared with all game objects, instead of copying this method to every game object class. Moreover, another advantage of having a same parent class is that it allows looping through different type of game objects. Instead of writing similar method for different class, they can be all generalized as GameObject class.

## **Encapsulation:**

In the design of my game, all the attribute inside GameObject and Frame class are set to private. Doing so, this will protect the data and make the code safer because attributes' values cannot be changed outside of the class. In order to control how these attributes can be changed, I also implement getter and setter methods.

## **The design of extension:**

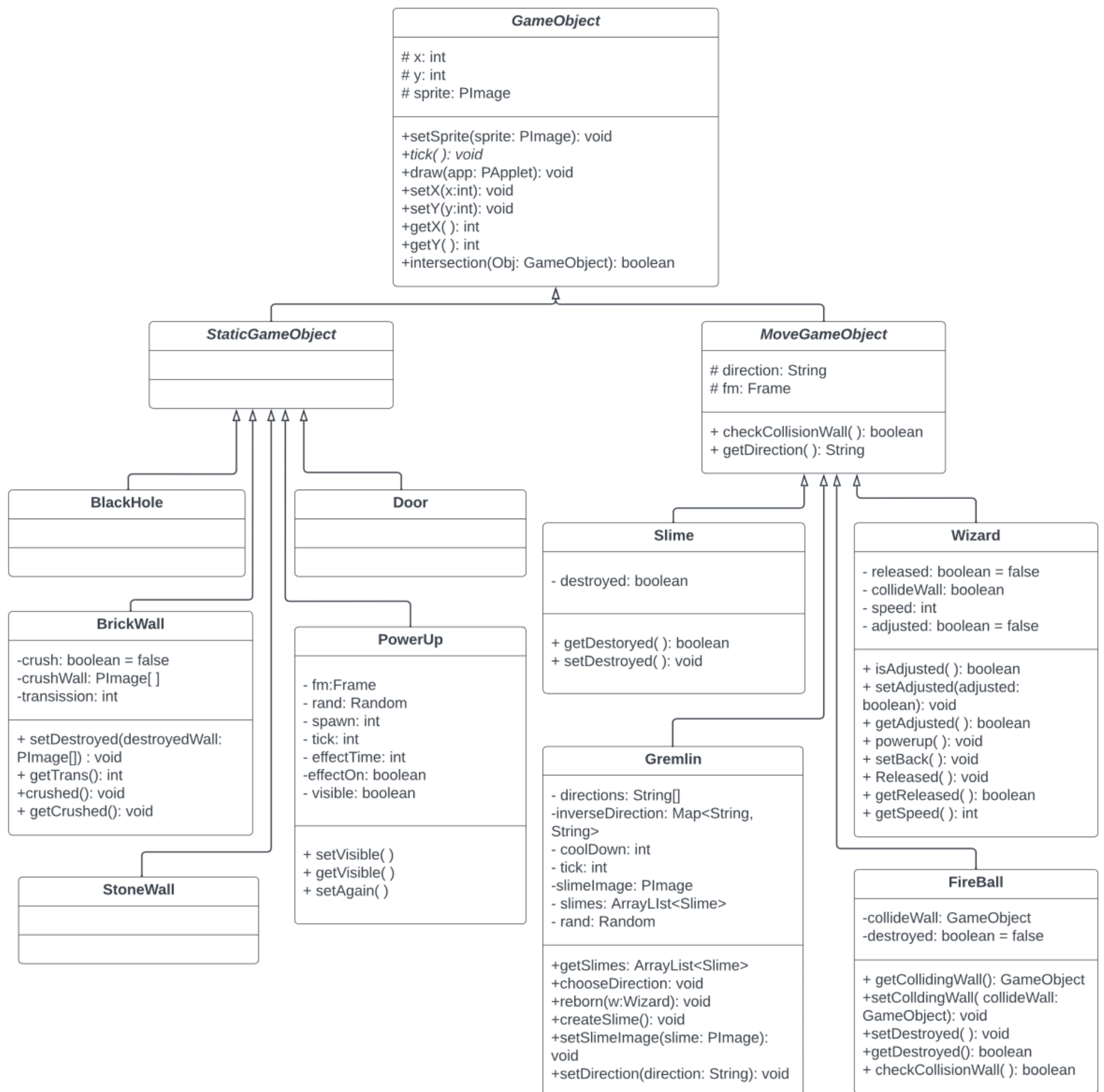
The extension I implemented is a black hole which teleports the player to another black hole. The location of Black Hole will be represented in the text file by the letter 'B,' which Frame will parse at the start of the game. Two Black Holes are typically required on a single map. The player can teleport between the Black holes anytime as they wish. Gremlins, however, cannot pass the black hole.

## **The design of power up:**

The functionality of powerup is to make the player speed up for 3 second. The initial location of power up will be represented by letter 'M' in the map. Around 10 seconds after the game began, it will first appear on the screen and resemble a magic stick. A progress bar named "speed up" will appear at the bottom right of the screen once the player has collected the powerup. The position of the powerup will change every time, and the next time it will appear about 10 seconds later.

(UMLs on next page)

## UML for All main classes & subclasses. (2 pages)



App
<p> <u>+WIDTH: int = 720</u>  <u>+HEIGHT: int = 720</u>  <u>+SPRITESIZE: int = 20</u>  <u>+BOTTOMBAR: int = 60</u>  <u>+FPS: int = 60</u>  <u>+randomGenerator: Random</u>  +configPath: String  +win: boolean  +lose: boolean  +lives: int  +wizardCoolDown: int  +wizardCoolDownCounter: int  +enemyCoolDown: int  +level: int = 0  +totalLevel: int  +magic: PowerUp  +brickwall: PImage  +stonewall: PImage  +wizardUp: PImage  +wizardDown: PImage  +wizardLeft: PImage  +wizardRight: PImage  +gremlin: PImage  +fireball: PImage  +door: PImage  +powerup: PImage  +crushWall: PImage[ ]  +tick: int  +progressbar: PShape  +filledProgressBar: PShape  +fm: Frame  +player: Wizard  +exit: Door  +gremlins: ArrayList&lt;Gremlin&gt;  +fireballs: ArrayList&lt;FireBall&gt;  +slimes: ArrayList&lt;Slime&gt;  +blackholes: ArrayList&lt;BlackHole&gt;  +fireOn: boolean  +transferred: boolean  +originalBlackHole: int = 0 </p>
<p> +settings( ): void  +setup( ): void  +KeyPressed( ): void  +KeyReleased( ): void  +setFrame( ): void  +tick( ): void  +draw( ): void  +checkNextLevel( ): void  +loseLives( ): void  +FireBallsTick( ):void  +GremlinsTick( ): void  +ProgressTick( ): void  +PowerupTick( ): void  +BlackHoleTick( ): void </p>

Frame
<p> - level: JSONObject  - wizardCoolDown: double  - enemyCoolDown: double  - map: GameObject[ ][ ]  - wizard: Wizard  - door: Door  - powerup: GameObject  - gremlins: ArrayList&lt;Gremlin&gt;  - blackholes: ArrayList&lt;BlackHole&gt; </p>
<p> + parseMap( ): void  + createObj( type: char, i: int, j: int):  GameObject  + setSprite(a: App): void  + tick( ): void  + draw( App a): void  + getWizardCoolDown( ): double  + getEnemyCoolDown( ): double  +getPowerup( ): GameObject  +setPowerup(powerup: GameObject ):  void  + getMap( ): GameObject[ ][ ]  + get(x: int, y: int): GameObject  + getWizard( ): Wizard  + getDoor( ): Door  + getBlackHole( ): ArrayList&lt;BlackHole&gt;  + getGremlins( ): ArrayList&lt;Gremlin&gt; </p>