CCT301: Project

Team Subpar’s Game Report:

System Design Report

Table of Contents

CRC Cards

Class name: GameObject

Parent class (if any): N/A

Subclasses (if any): MovableObject, StaticObject

Responsibilities:

* Acts as an object that must be drawn within the game that the player can see

Collaborators:

* Player class

* Bullet class

* Wall class

Class name: MovableObject

Parent class (if any): GameObject

Subclasses (if any): Player class, bullet class

Responsibilities:

* A GameObject that can move and has velocity

Collaborators:

* Player class

* Bullet class

* GameObject

Class name: StaticObject

Parent class (if any): GameObject

Subclasses (if any): wall class

Responsibilities:

* A GameObject that is fixed in space

Collaborators:

* wall class

* GameObject

Class name: Player class

Parent class (if any): GameObject Interface

Subclasses (if any): N/A

Responsibilities:

* shows the player avatar on the screen.

* moves the player avatar.

* notifies the GameObject interface when a game “tick” is updated such as when the player is moving.

Collaborators:
* GameObject interface

* Wall class

Class name: Bullet class

Parent class (if any): MovableObject

Subclasses (if any): N/A

Responsibilities:

* shows the bullet location on the screen.

* bullet moves under specific rules, such as rebounding off of walls.

Collaborators:

* MovableObject

* Wall class

Class name: Wall class

Parent class (if any): StaticObject

Subclasses (if any): N/A

Responsibilities:

* shows the wall location on the game screen.

* walls follow a set number of rules, such as blocking a player or rebounding a shot.

Collaborators:

* StaticObject

* Player class

* Shot class

Class name: KeyHandler

Parent class (if any): KeyAdapter

Subclasses (if any): N/A

Responsibilities:

* Handles all the key presses and notifies all of it's observers that are looking for key strokes.

Collaborators:

* KeyObservable

Class name: KeyObserver

Parent class (if any): Observable

Subclasses (if any): N/A

Responsibilities:

* Keeps track of all of it's observers and notifies when invoked

Collaborators:

* KeyHandler

* Player class

System Interaction

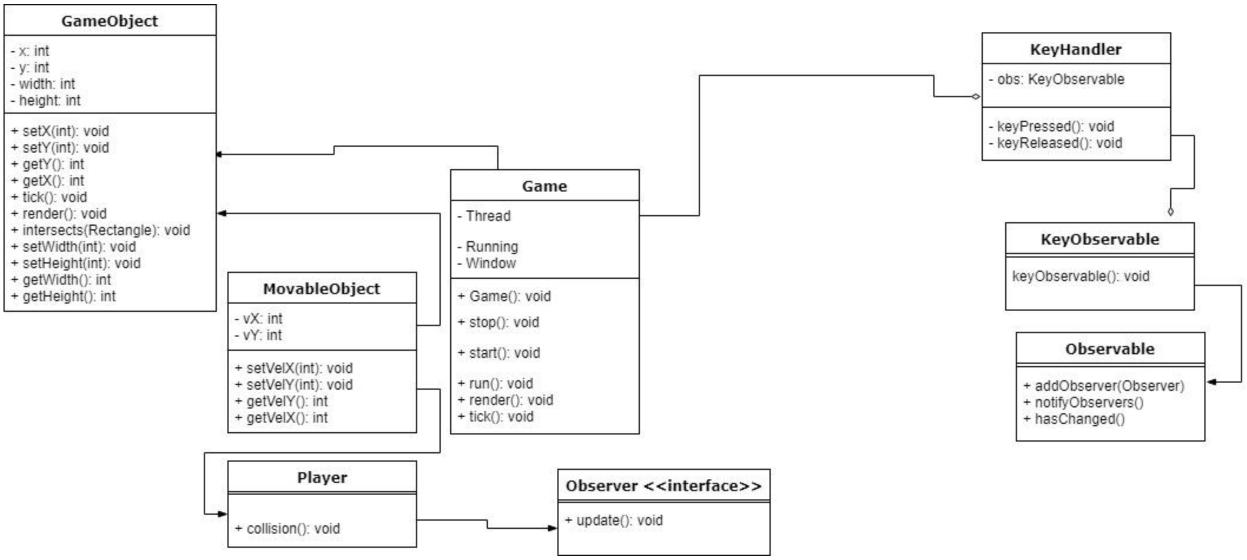
- The system should be able to run a .jar file, so they would need java of a version of 8.0.1110.14 or greater

----

System Decomposition

- Since the user input is whitelisted, the program can only accept user keystrokes that move the player avatar. Therefore, no exception handling should be needed for failure testing.

System Architecture


GameObject
- x: int
- y: int
- width: int
- height: int
+ setX(int): void
+ setY(int): void
+ getY(): int
+ getX(): int
+ tick(): void
+ render(): void
+ intersects(Rectangle): void
+ setWidth(int): void
+ setHeight(int): void
+ getWidth(): int
+ getHeight(): int
MovableObject
- vX: int
- vY: int
+ setVelX(int): void
+ setVelY(int): void
+ getVelY(): int
+ getVelX(): int
Game
- Thread
- Running
- Window
+ Game(): void
+ stop(): void
+ start(): void
+ run(): void
+ render(): void
+ tick(): void
Player
+ collision(): void
Observer <<interface>>
+ update(): void
KeyHandler
- obs: KeyObservable
- keyPressed(): void
- keyReleased(): void
KeyObservable
keyObservable(): void
Observable
+ addObserver(Observer)
+ notifyObservers()
+ hasChanged()