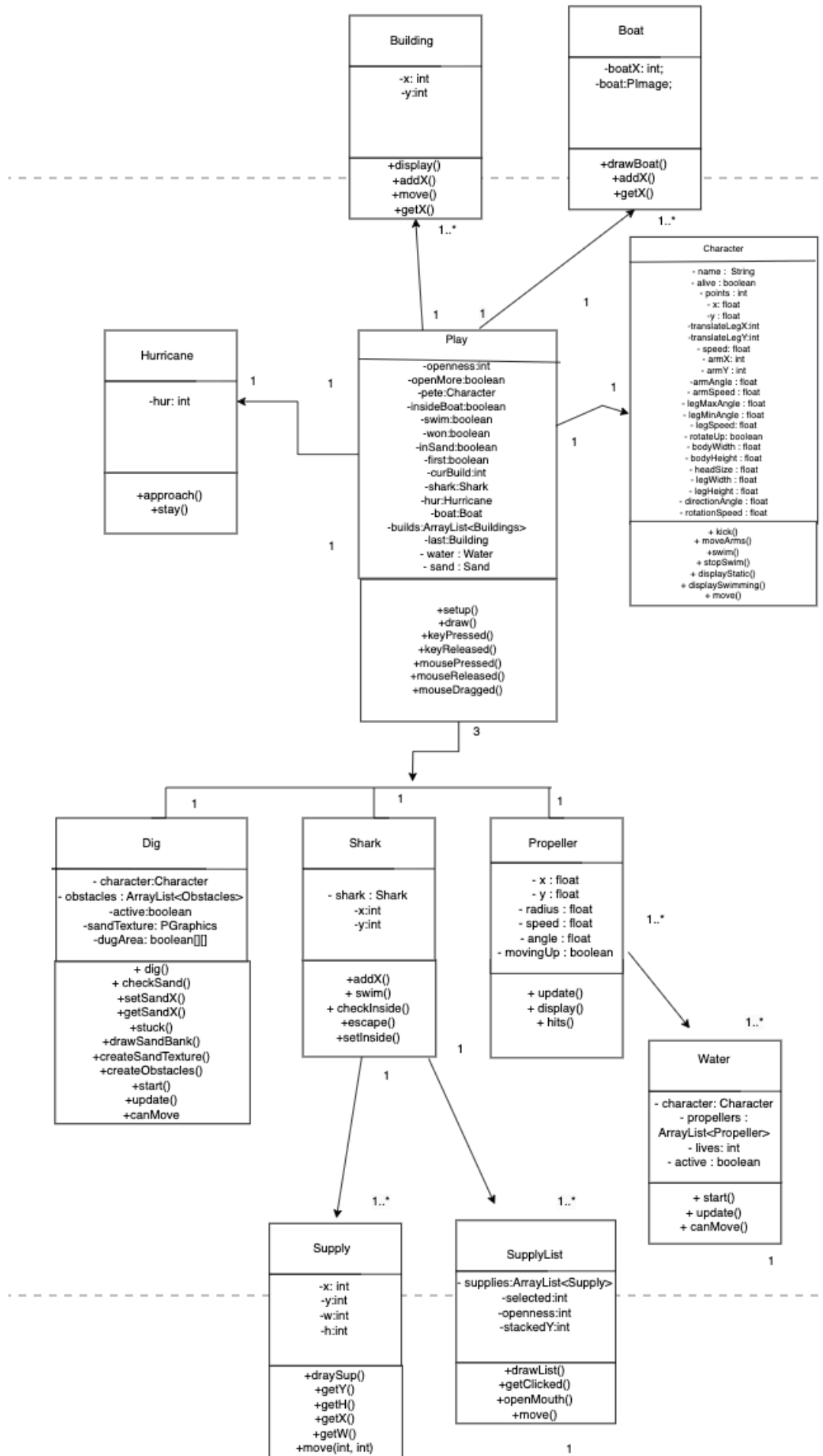Angela Mashuryan and Ellen Wright

Period 7

Project Title: Hurricane Hoboken

Group Name: The Adventurers

We are creating a game similar to the Oregon Trail called Hurricane Hoboken. The user will navigate through the Hudson River, trying to get from New Jersey to Stuy. They will encounter obstacles that will take them into mini-games. They must reach the end before the hurricane approaches; otherwise, they die. The mini-games will include a game where they have to dig out of a sand bank, one where they escape a shark, and one where they avoid the propellers of a boat. We are not planning on using libraries for this project.

Building

-x: int
-y:int

+display()
+addX()
+move()
+getX()

1..*

Boat

-boatX: int;
-boat:PImage;

+drawBoat()
+addX()
+getX()

1..*

Character

- name : String
- alive : boolean
- points : int
- x : float
- y : float
- translateLegX:int
- translateLegY:int
- speed: float
- armX : int
- armY : int
- armAngle : float
- armSpeed : float
- legMaxAngle : float
- legMinAngle : float
- legSpeed : float
- rotateUp: boolean
- bodyWidth : float
- bodyHeight : float
- headSize : float
- legWidth : float
- legHeight : float
- directionAngle : float
- rotationSpeed : float

+ kick()
+ moveArms()
+swim()
+ stopSwim()
+ displayStatic()
+ displaySwimming()
+ move()

Hurricane

-hur: int

+approach()
+stay()

Play

-openness:int
-openMore:boolean
-pete:Character
-insideBoat:boolean
-swim:boolean
-won:boolean
-inSand:boolean
-first:boolean
-curBuild:int
-shark:Shark
-hur:Hurricane
-boat:Boat
-builds:ArrayList<Buildings>
-last:Building
- water : Water
- sand : Sand

+setup()
+draw()
+keyPressed()
+keyReleased()
+mousePressed()
+mouseReleased()
+mouseDragged()

3

1

1

1

1

1

1

Dig

- character:Character
- obstacles : ArrayList<Obstacles>
-active:boolean
-sandTexture: PGraphics
-dugArea: boolean[][]

+ dig()
+ checkSand()
+setSandX()
+getSandX()
+stuck()
+drawSandBank()
+createSandTexture()
+createObstacles()
+start()
+update()
+canMove

Shark

- shark : Shark
-x:int
-y:int

+addX()
+ swim()
+ checkInside()
+escape()
+setInside()

Propeller

- x : float
- y : float
- radius : float
- speed : float
- angle : float
- movingUp : boolean

+ update()
+ display()
+ hits()

1..*

Water

- character: Character
- propellers :
ArrayList<Propeller>
- lives: int
- active : boolean

+ start()
+ update()
+ canMove()

1..*

1

1

1

1..*

1..*

Supply

-x: int
-y:int
-w:int
-h:int

+draySup()
+getY()
+getH()
+getX()
+getW()
+move(int, int)

SupplyList

- supplies:ArrayList<Supply>
-selected:int
-openness:int
-stackedY:int

+drawList()
+getClicked()
+openMouth()
+move()

1

1

Directions:

To swim, the user will use the arrow keys (e.g., the right arrow key to move forward). Directions will be displayed on the screen in each of the mini-games. In the shark game, they will have to stack objects with their mouse. In the sandbank, they will have to use their mouse to dig and then use their arrow keys to move. In the propeller game, they must use the arrow keys to navigate and evade the propellers.

Functionalities/Issues:

We currently have our three mini-games, dig, shark, and propeller, and the character class, which provides functionality for the user to advance through the game. We have the background set up to display a cityscape and the water that the user will navigate through. We have the swim method, which causes the background to move backward and the character to swim forward. The character is transported into different settings when they encounter different obstacles.

Log:

Angela created the propeller and dig mini-games. Ellen created the background and swimmer and the shark mini-game.