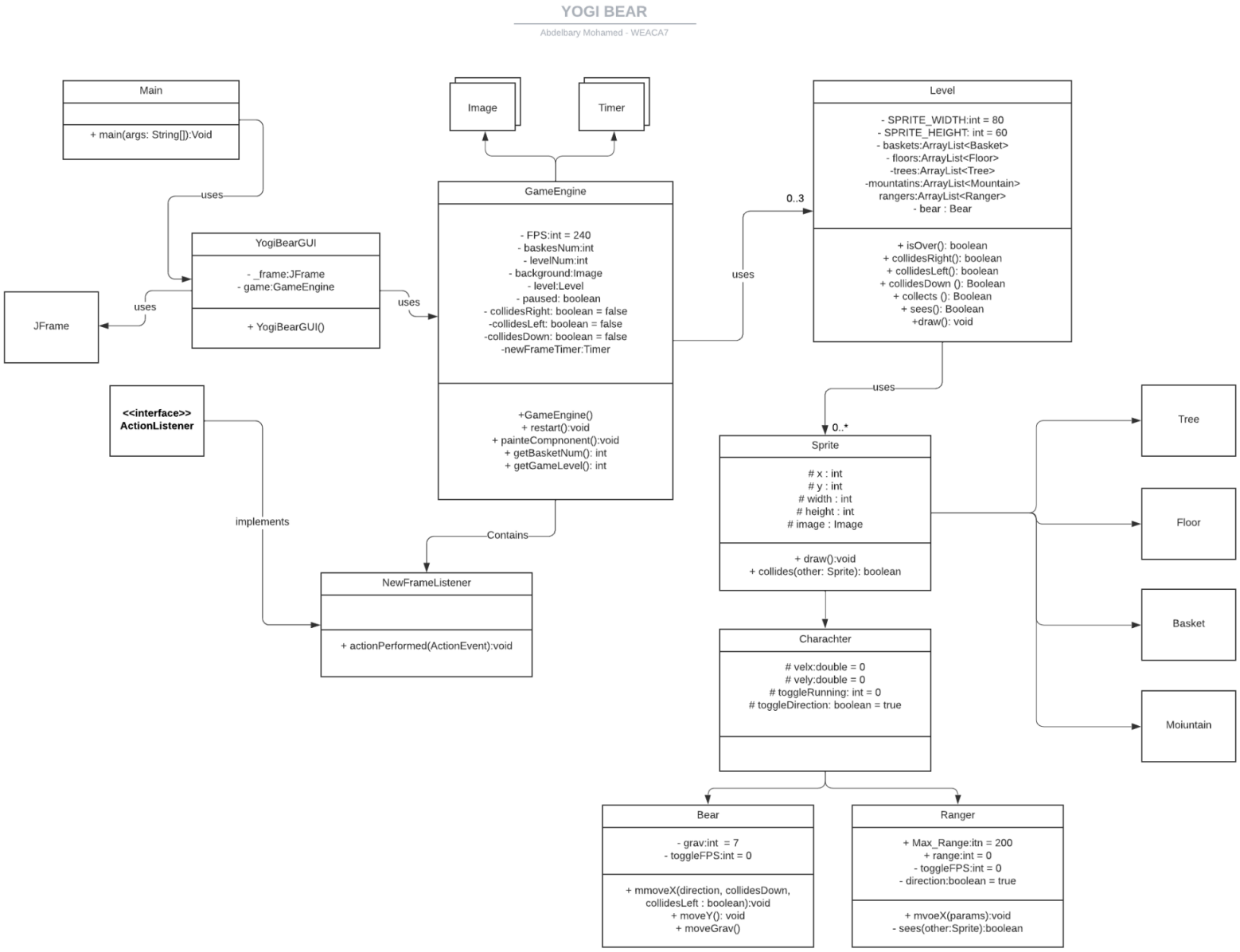
Yogi Bear

Mohamed Abdelbary, WEACA7

Yogi Bear wants to collect all the picnic baskets in the forest of the Yellowstone National Park. This park contains mountains and trees, which are obstacles for Yogi. Besides the obstacles, there are rangers, who make it harder for Yogi to collect the baskets. Rangers can move only horizontally or vertically in the park. If a ranger gets too close (one unit distance) to Yogi, then Yogi loses one life. (It is up to you to define the unit, but it should be at least that wide, as the sprite of Yogi.) If Yogi still has at least one life from the original three, then he spawns at the entrance of the park. During the adventures of Yogi, the game counts the number of picnic baskets that Yogi collected. If all the baskets are collected, then load a new game level, or generate one. If Yogi loses all his lives, then show a popup message box, where the player can type his name and save it to the database. Create a menu item, which displays a highscore table of the players for the 10 best scores. Also, create a menu item which restarts the game.

UML Diagram


YOGI BEAR
Abdelbary Mohamed - WEACA7
Main
+ main(args: String[]):Void
uses
YogiBearGUI
- _frame:JFrame
- game:GameEngine
+ YogiBearGUI()
uses
JFrame
uses
Image
Timer
GameEngine
- FPS:int = 240
- baskesNum:int
- levelNum:int
- background:Image
- level:Level
- paused: boolean
- collidesRight: boolean = false
-collidesLeft: boolean = false
-collidesDown: boolean = false
-newFrameTimer:Timer
+GameEngine()
+ restart():void
+ painteCompnonent():void
+ getBasketNum(): int
+ getGameLevel(): int
0..3
uses
Level
- SPRITE_WIDTH:int = 80
- SPRITE_HEIGHT: int = 60
- baskets:ArrayList<Basket>
- floors:ArrayList<Floor>
-trees:ArrayList<Tree>
-mountatins:ArrayList<Mountain>
rangers:ArrayList<Ranger>
- bear : Bear
+ isOver(): boolean
+ collidesRight(): boolean
+ collidesLeft(): boolean
+ collidesDown (): Boolean
+ collects (): Boolean
+ sees(): Boolean
+draw(): void
uses
0..*
<<interface>>
ActionListener
implements
Contains
NewFrameListener
+ actionPerformed(ActionEvent):void
Sprite
x : int
y : int
width : int
height : int
image : Image
+ draw():void
+ collides(other: Sprite): boolean
Tree
Floor
Basket
Moiuntain
Charachter
velx:double = 0
vely:double = 0
toggleRunning: int = 0
toggleDirection: boolean = true
Bear
- grav:int = 7
- toggleFPS:int = 0
+ mmoveX(direction, collidesDown, collidesLeft : boolean):void
+ moveY(): void
+ moveGrav()
Ranger
+ Max_Range:itn = 200
+ range:int = 0
- toggleFPS:int = 0
- direction:boolean = true
+ mvoeX(params):void
- sees(other:Sprite):boolean

Classes and Methods

1. YogiBear Class
    a. The constructor creates a frame and appends a GameEngine object to it
2. GameEngine Class
    a. Restart: it creates a new level based on the integer levelNum
    b. paintComponent: it draws the background and calls the draw method of level
    c. NewFrameListener Class:
        i. actionPerformed: it checks the boolean paused and then moves the rangers, calls collects function and sits the values for collidesRight Left, Down. Also, it checks for isOver method to decide if the game is over or not. Finally, it repaints.
3. Level Class:
    a. The constructor reads the level files and then creates the object by iterating through each character and create an equivalent object for it.
    b. isOver: returns true if there is not basket
    c. collidesRight: returns true if there is a tree or mountain to the right of the bear
    d. collidesLeft: returns true if there is a tree of a mountains to left of the bear
    e. collidesDown: returns true if there is a floor under the bear
    f. Collects: returns true if there is intersection between the bear and a basket
    g. Sees: returns true if the distance between a ranger and the bear is 40 pixels
    h. Draw: it iterates throw every sprite to call its draw method
4. Sprite

    a. Draw: it draws the object's image according to its x,y, width, height
    b. Collides: checks if the sprite collides with another one
5. Basket
    a. Collects: if it collides with a bear it returns true
6. Bear
    a. MoveX: it checks if the bear collides right or left, if not it moves according to the direction with the defined speed.
    b. moveY: it moves the bear vertically by increasing its Y
    c. moveGrav: whenever there is no colliding with the floor, it moves the bear down according to already defined gravitation speed
7. Ranger
    a. moveX: it changes the x member of the object according to its direction between a certain range.
    b. Sees: returns true if there's 40 pixels between the object and the bear

TESTING

1. Checking the KeyStrokes
    a. LEFT
    b. RIGHT
    c. UP
    d. ESCAPE
2. Checking the baskets when the bear collides
3. Checking the bear when collides with the a mountain or a tree
4. Checking the gravitation when the bear flies and the floor colliding with the bear
5. Checking if there is a 40 pixels between the bear and a ranger
6. Checking what happens if there is no basket in the level
7. Checking animations of the bear when running and when flying
8. The rangers always walk in a certain range
9. The levels are solvable
10. Checking how the game handles if the bear went out of the border