|  |
| --- |
| +: public  -: private  #: protected |

**Class Diagrams**

|  |
| --- |
| **Projectile {abstract}**  #width: int  #height : int  #Vx : int  #Vy : int  #Player player- Instance  #Enemy enemy-Instance  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  #hitObject()- Controls damage infliction and removing projectile  #outOfBounds ()-Checks if instance of a bullet is out of bounds of world.  Abstract- hasHit(); |

|  |
| --- |
| **Enemy {Main Class}** |
| + enemyLives :int  - xSpeed:int  - moving :boolean  - movingLeft :boolean  - movingRight :boolean  - lastDir : String  \* These values are all utilized to determine movement of the enemy, and whether he is dead yet.  - shotDelay:int  - rateOfFire:int  - bulletSpeed:int  -shootRange:int  -searchRange:int  \*These values are utilized to determine the enemy's shooting timing, and to coordinate with the enemyBullet class. |
| +act () - Will continuously move the enemy back and forth on a flat platform during the boss battle. Will also shoot during appropriate time intervals.  -shoot (): void- Will shoot a bullet with appropriate angles as needed and reset shotDelay after each shot.  -findTarget (): boolean- Will determine the location of a player target using Greenfoot built in methods. If found it will be shot at.  -getRadRotation: double- will obtain the angle to rotate the bullet.  +decreaseLives():void- will decrease enemy lives |

|  |
| --- |
| **+-Bullet**  -xSpeed: int  -rightDirection: boolean  #Player player- Instance  Constructor sets bullet’s speed and it will remain constant until contact.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  +act()- Checks if bullet is out of bounds, will also inflict damage on enemy upon contact, and move bullet according to logical speeds.  infliction and removing projectile  +hasHit (): boolean - checks if object is between projectile’s horizontal and vertical velocity. If so, it will hit object and leave. |

|  |
| --- |
| **+-EnemyBullet**  #Vx : int  #Vy : int  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  +act()- Checks if bullet is out of bounds, will also inflict damage on player upon contact, and move bullet according to logical speeds.  infliction and removing projectile  +hasHit (): boolean - checks if object is between projectile’s horizontal and vertical velocity. If so, it will hit object and leave. |

|  |
| --- |
| **ASImgBars** |
| -lives : int  -fullLives : int  -height: int  \*These values are all used for building the box to display the hearts in, and the numerical value for number of hearts. |
| +update (livesModified): int- This method is used to modify the number of lives, and limit them from going above five. Alternatively, it avoids going below zero.  +updateImage():void  This method is used to draw a set number of hearts based on the lives currently available. This method is used in the World class. |

|  |
| --- |
| **Timer** |
| + startTime: long  - currentTime: long  - counter: int  -message: Message  \*Values used to keep track of the different times and the creation of the timer's message in the world (which displays time) |
| +act()-after the first act, adds a message object to the world. Every act, continues to update the message with new times  +getTimeDifference(firstTime):  String-This method is used to find the difference between the given time (start time) and the current time and returns the number as a string. This is used in the World and Timer class. |

|  |
| --- |
| **Messages** |
| -text: String  -color: Color  \*These are used for creating a message and it's greenfoot Image |
| +act()-Every act, continues to update the image of the message with the text  +updateMessage(message, color):void  This method is used to update the message with a different text or foreground colour. |

|  |
| --- |
| **JumpBoost** |
| - counter: int  - actCounter: int  -frameCounter: int  \*Values used to control how long the image stays on the screen and change sprites |
| +act()-Continuously turns the Tigger image. Every few acts, the image will increase in size through a method. Removes the object when the counter reaches 0.  -imageSizeIncrease():void-This method is used to set the image of the jump boost to a larger version |

|  |
| --- |
| **HealthUp** |
| - counter: int  - actCounter: int  -frameCounter: int  \*Values used to control how long the image stays on the screen and change sprites |
| +act()-Continuously turns the Heart image. Every few acts, the image will increase in size through a method. Removes the object when the counter reaches 0.  -imageSizeIncrease():void-This method is used to set the image of the jump boost to a larger version |

|  |
| --- |
| **PowerUp** |
| -imageNum: int  -actNum: int  \*Values used to change the image of the power up |
| +act()-Moves the block towards the left every act. Changes the colour of the power up every few acts. |

|  |
| --- |
| **WorldMap** |
| -counterReset:int  -counter:int  -index:int  -length:int  -pauseTime:long  -lives : int  -fullLives : int  -height: int  -blocks : String[][]  -spawnmap:boolean  - Player player - Instance  -Enemy enemy -Instance  -AsImgBars healthbar - Instance  - Timer timer - instance  -Pause:boolean  -mapscroll:boolean  - Scrollspeed:int  -backgroundmusic:Greenfootsound  \*These values are used to create the map that is shown, it also has instance variables. They help to control the general world. |
| +act() - Checks if game is paused, checks if it is the first time running the world and if so loads the map, checks if it should spawn the map if so spawns the map by columns, checks if the player is dead if so removes him & checks if the enemy is dead if so removes him.  +isPause():boolean - Checks if the game is paused  +isScrolling():boolean- Checks if the game is scrolling  +scrollSpeed ():int - Checks the scroll speed of the map  +unpause() - Unpauses the game.  +loadMap(filename:String, length:int) - Loads the map into a 2d array from a text file  +getBlock(curBlock:int, ySpawn:int) - Gets the block from number and spawns it using the yCoordinate  +spawnObject(height:int, index:int) - Spawns a column of the map into the world |

|  |
| --- |
| **LevelOne** |
| All variables inherited from world |
| +prepare () - Sets run scrollSpeed to 1, level to 1 |

|  |
| --- |
| **LevelTwo** |
| All variables inherited from world |
| + prepare () - Sets run scrollSpeed to 2, level to 2 |

|  |
| --- |
| **LevelThree** |
| All variables inherited from world |
| + prepare () - Sets run scrollSpeed to 4, level to 3 |

|  |
| --- |
| **Player** |
| -hitboxY:int  -hitBoxX:int  \*Values for hitbox  -invincibility:boolean  -invincibilityCounter:int  -playerLives:int  -playerMaxLives:int  -jumpBonus:boolean  -jumpBonusCounter:int  -shootDelay:int  \*Values on player status  -speed:int[]  -ySpeed:int[]  -ySpeedUpgrade:int[]  -jumpIndex:int[]  -speedIndex:int  -jumpUndex:int  \*Values for player movement  -jump:Boolean  -canJump:Boolean  -fall:Boolean  -moving:Boolean  -movingLeft:Boolean  -movingRight:Boolean  -lastDir:String  -climbHeight:int  \*Conditions for movement  -runIndex:int  \*Frames for running  -jumpMarker:Instance  \*For marking movement  -healthUpSound:GreenfootSound  -jumpBoostSound:GreenfootSound  \*Sound effects |
| +act()-Sets player’s image, check for collisions, check for movement in both directions, check for keyboard keys  -hazardCollision():Boolean – checks for collision with hazards  -playerKeys() – check for if player hits keys for movement or shooting  -sideCollision() – move players if they collide with walls so they don’t get stuck as often  -pixelCollisionHR() – grants movement and checks for collision right of the player  -pixelCollisionHL() – grants movement and checks for collision left of the player  -slideRight() – slide right if speed to right is not 0  -slideLeft() – slide left if speed to left is not 0  -jump() – lets player jump and checks for collision above the head  -fall() – lets player fall and checks for collision bellow  -playerDeath() – gives player invincibility and updates health  -powerUp() – gives player random powerup  -tempJumpBonus() – give player higher jump  -tempInvinciblity() – times invincibility  +findLastDir ()– find player’s direction facing  +healthIncrease() – increase player health  +healthDecrease() – decrease player health  +currentHealth() –gets current amount of lives player has  +MaxHealth() – gets max health player can have |