

The Legend of Python

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	actor.aquamentus.Aquamentus Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	__init__()	8
4.1.3	Member Function Documentation	8
4.1.3.1	attack()	9
4.1.3.2	bossLogic()	9
4.1.3.3	checkState()	9
4.1.3.4	swapDirection()	9
4.2	actor.boomerang.Boomerang Class Reference	9
4.2.1	Detailed Description	10
4.2.2	Constructor & Destructor Documentation	10
4.2.2.1	__init__()	10
4.3	actor.boss.Boss Class Reference	11
4.3.1	Detailed Description	12

4.3.2	Constructor & Destructor Documentation	12
4.3.2.1	__init__()	12
4.3.3	Member Function Documentation	12
4.3.3.1	hit()	12
4.3.3.2	move()	13
4.3.3.3	update()	13
4.4	collision.door.Door Class Reference	13
4.4.1	Detailed Description	14
4.4.2	Constructor & Destructor Documentation	14
4.4.2.1	__init__()	14
4.5	actor.enemy.Enemy Class Reference	14
4.5.1	Detailed Description	15
4.5.2	Constructor & Destructor Documentation	15
4.5.2.1	__init__()	15
4.5.3	Member Function Documentation	16
4.5.3.1	hit()	16
4.5.3.2	move()	16
4.5.3.3	update()	16
4.5.4	Member Data Documentation	16
4.5.4.1	hitCount	17
4.5.4.2	hitdir	17
4.6	actor.fireball.Fireball Class Reference	17
4.6.1	Detailed Description	18
4.6.2	Constructor & Destructor Documentation	18
4.6.2.1	__init__()	18
4.6.3	Member Function Documentation	18
4.6.3.1	checkState()	19
4.6.3.2	end()	19
4.6.3.3	hit()	19
4.6.3.4	logic()	19

4.6.3.5	move()	19
4.6.3.6	start()	19
4.6.3.7	update()	20
4.7	actor.healthbar.Health_Bar Class Reference	20
4.7.1	Detailed Description	21
4.7.2	Constructor & Destructor Documentation	21
4.7.2.1	__init__()	21
4.7.3	Member Function Documentation	21
4.7.3.1	health()	21
4.7.4	Member Data Documentation	22
4.7.4.1	h_sprite_sheet	22
4.7.4.2	image	22
4.7.4.3	rect	22
4.8	actor.item.Item Class Reference	22
4.8.1	Detailed Description	23
4.8.2	Constructor & Destructor Documentation	23
4.8.2.1	__init__()	23
4.8.3	Member Function Documentation	24
4.8.3.1	collision()	24
4.9	actor.keese.Keese Class Reference	24
4.9.1	Detailed Description	25
4.9.2	Constructor & Destructor Documentation	25
4.9.2.1	__init__()	25
4.9.3	Member Function Documentation	25
4.9.3.1	checkState()	26
4.9.3.2	enemyLogic()	26
4.9.3.3	genRestLength()	26
4.9.3.4	genTravelPoint()	26
4.9.3.5	setMoveSpeed()	26
4.9.3.6	stop()	27

4.9.3.7	switchSprite()	27
4.10	collision.level.Level Class Reference	27
4.10.1	Detailed Description	28
4.11	collision.levelmanager.LevelManager Class Reference	28
4.11.1	Detailed Description	28
4.11.2	Member Function Documentation	28
4.11.2.1	transition()	28
4.12	actor.player.Player Class Reference	29
4.12.1	Detailed Description	31
4.12.2	Constructor & Destructor Documentation	31
4.12.2.1	__init__()	31
4.12.3	Member Function Documentation	31
4.12.3.1	attackupdate()	31
4.12.3.2	move()	31
4.12.3.3	moveupdate()	32
4.12.4	Member Data Documentation	32
4.12.4.1	attacksword	32
4.12.4.2	dir	32
4.13	actor.rupee.Rupee_Bar Class Reference	32
4.13.1	Detailed Description	33
4.13.2	Constructor & Destructor Documentation	33
4.13.2.1	__init__()	33
4.13.3	Member Data Documentation	33
4.13.3.1	image	33
4.13.3.2	rect	34
4.14	actor.spritesheet.SpriteSheet Class Reference	34
4.14.1	Detailed Description	34
4.14.2	Constructor & Destructor Documentation	35
4.14.2.1	__init__()	35
4.14.3	Member Function Documentation	35

4.14.3.1	get_image()	35
4.14.3.2	get_imageNT()	36
4.14.4	Member Data Documentation	36
4.14.4.1	sprite_sheet	36
4.15	actor.stalfos.Stalfos Class Reference	36
4.15.1	Detailed Description	38
4.15.2	Constructor & Destructor Documentation	38
4.15.2.1	__init__()	38
4.15.3	Member Function Documentation	38
4.15.3.1	checkState()	38
4.15.3.2	enemyLogic()	38
4.15.3.3	genTravelPath()	39
4.15.3.4	setWalkSpeed()	39
4.15.3.5	stop()	39
4.15.4	Member Data Documentation	39
4.15.4.1	direction	39
4.15.4.2	dmg	39
4.15.4.3	HP	40
4.15.4.4	image	40
4.15.4.5	isHit	40
4.15.4.6	isMoving	40
4.15.4.7	maxHP	40
4.15.4.8	obj	40
4.15.4.9	oldx	41
4.15.4.10	oldy	41
4.15.4.11	previousDirection	41
4.15.4.12	spriteIndex	41
4.15.4.13	sprites	41
4.15.4.14	walkFrames	41
4.15.4.15	walkStartFrame	42

4.15.4.16 xSpeed	42
4.15.4.17 ySpeed	42
4.16 actor.sword.Sword Class Reference	42
4.16.1 Detailed Description	43
4.16.2 Constructor & Destructor Documentation	43
4.16.2.1 __init__()	43
4.16.3 Member Data Documentation	43
4.16.3.1 image	43
4.17 collision.wall.Wall Class Reference	44
4.17.1 Detailed Description	44
4.17.2 Constructor & Destructor Documentation	44
4.17.2.1 __init__()	44
4.17.3 Member Function Documentation	45
4.17.3.1 collision()	45
4.17.4 Member Data Documentation	45
4.17.4.1 id	45
4.17.4.2 image	45
4.17.4.3 rect	46
5 File Documentation	47
5.1 src/actor/aquamentus.py File Reference	47
5.1.1 Detailed Description	47
5.1.2 Function Documentation	48
5.1.2.1 populateSprites()	48
5.2 src/actor/boomerang.py File Reference	48
5.2.1 Detailed Description	48
5.3 src/actor/boss.py File Reference	48
5.3.1 Detailed Description	49
5.4 src/actor/constants.py File Reference	49
5.4.1 Detailed Description	50
5.5 src/actor/enemy.py File Reference	51

5.5.1	Detailed Description	51
5.6	src/actor/fireball.py File Reference	51
5.6.1	Detailed Description	52
5.6.2	Function Documentation	52
5.6.2.1	populateSprites()	52
5.7	src/actor/healthbar.py File Reference	52
5.7.1	Detailed Description	52
5.8	src/actor/item.py File Reference	53
5.8.1	Detailed Description	53
5.9	src/actor/keese.py File Reference	53
5.9.1	Detailed Description	54
5.9.2	Function Documentation	54
5.9.2.1	populateSprites()	54
5.10	src/actor/player.py File Reference	54
5.10.1	Detailed Description	54
5.11	src/actor/rupee.py File Reference	55
5.11.1	Detailed Description	55
5.12	src/actor/spritesheet.py File Reference	55
5.12.1	Detailed Description	55
5.13	src/actor/stalfos.py File Reference	55
5.13.1	Detailed Description	56
5.13.2	Function Documentation	56
5.13.2.1	populateSprites()	56
5.14	src/actor/sword.py File Reference	56
5.14.1	Detailed Description	57
5.15	src/collision/door.py File Reference	57
5.15.1	Detailed Description	57
5.16	src/collision/level.py File Reference	57
5.16.1	Detailed Description	58
5.17	src/collision/levelmanager.py File Reference	58
5.17.1	Detailed Description	58
5.18	src/collision/wall.py File Reference	58
5.18.1	Detailed Description	59
5.19	src/config/colour.py File Reference	59
5.19.1	Detailed Description	59
5.20	src/config/window.py File Reference	59
5.20.1	Detailed Description	60

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

collision.levelmanager.LevelManager	28
object	
actor.spritesheet.SpriteSheet	34
Sprite	
actor.boomerang.Boomerang	9
actor.boss.Boss	11
actor.aquamentus.Aquamentus	7
actor.enemy.Enemy	14
actor.keese.Keese	24
actor.stalfos.Stalfos	36
actor.fireball.Fireball	17
actor.healthbar.Health_Bar	20
actor.item.Item	22
actor.player.Player	29
actor.rupee.Rupee_Bar	32
actor.sword.Sword	42
collision.door.Door	13
collision.level.Level	27
collision.wall.Wall	44

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

actor.aquamentus.Aquamentus		
	This class represents the Aquamentus Boss	7
actor.boomerang.Boomerang		
	Boomerang Weapon Class The class holding the creation, behaviour, and collision effects of the player's boomerang weapon	9
actor.boss.Boss		
	Superclass for representing a Boss	11
collision.door.Door		
	Dungeon Door Class This class is used to place a door on one of four places of a dungeon room, to allow the player to walk through and traverse to other rooms within the dungeon	13
actor.enemy.Enemy		
	Superclass for representing an Enemy	14
actor.fireball.Fireball		
	This class represents the Fireball object	17
actor.healthbar.Health_Bar		
	This class represents the HealthBar for the user controlled player character	20
actor.item.Item		
	Consumable Item Class This class is used for the creation of a random consumable item, spawned once an enemy has been defeated on their last position before despawn	22
actor.keese.Keese		
	This class represents the Keese enemy	24
collision.level.Level		
	Dungeon level background class Creates the background of a dungeon, can be modified in the future to change colour based on sprite sheet and new arg	27
collision.levelmanager.LevelManager		
	Dungeon Level Creation Class Class used as a master constructor for every dungeon levels, getting pre-written data and loading it when the game starts and when a transition occurs	28
actor.player.Player		
	Player Class A pygame sprite subclass for defining the creation of the game's playable character, as well as its interactions with both the user and other entities within the game	29
actor.rupee.Rupee_Bar		
	This class represents the RupeeBar object	32
actor.spritesheet.SpriteSheet		
	This class represents the SpriteSheet object, allowing sprites to be loaded and processed	34
actor.stalfos.Stalfos		
	This class represents the Stalfos enemy	36

[actor.sword.Sword](#)

Player [Sword](#) Class Class for the creation and deletion of the sword sprite object, made when the player attacks 42

[collision.wall.Wall](#)

This class represents the [Wall](#) class for collision for objects in the environment 44

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

src/actor/ aquamentus.py	
Aquamentus Boss	47
src/actor/ boomerang.py	
Boomerang Weapon	48
src/actor/ boss.py	
Boss Template	48
src/actor/ constants.py	
Actor Constants	49
src/actor/ enemy.py	
Enemy Template	51
src/actor/ fireball.py	
Fireball object	51
src/actor/ healthbar.py	
HealthBar Class	52
src/actor/ item.py	
Consumable Items	53
src/actor/ keese.py	
Keese Enemy	53
src/actor/ player.py	
Playable Characer	54
src/actor/ rupee.py	
Rupee_Bar Class	55
src/actor/ spritesheet.py	
SpriteSheet Class	55
src/actor/ stalfos.py	
Stalfos Enemy	55
src/actor/ sword.py	
Player Sword	56
src/collision/ door.py	
Dungeon Door	57
src/collision/ level.py	
Dungeon Background	57
src/collision/ levelmanager.py	
Dungeon Level Master Creator	58
src/collision/ wall.py	
Wall Class	58

src/config/ colour.py	
Colour Constants	59
src/config/ window.py	
Window Constants	59

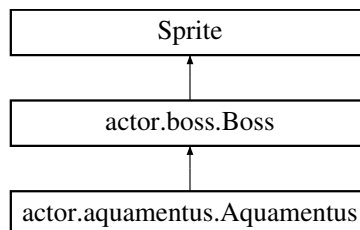
Chapter 4

Class Documentation

4.1 actor.aquamentus.Aquamentus Class Reference

This class represents the [Aquamentus](#) Boss.

Inheritance diagram for actor.aquamentus.Aquamentus:



Public Member Functions

- `def __init__ (self, x, y)`
Constructor for [Aquamentus](#).
- `def checkState (self)`
Evaluates the state of the [Aquamentus](#).
- `def swapDirection (self)`
Swaps [Aquamentus](#)' direction.
- `def attack (self)`
Allows for [Aquamentus](#) to attack.
- `def bossLogic (self)`
Controls [Aquamentus](#) logic.

Public Attributes

- **isAttacking**
- **attackStartFrame**
- **fireballs**
- **xSpeed**
- **oldspeed**
- **maxHP**
- **HP**
- **dmg**
- **image**
- **obj**
- **sprites**
- **spriteIndex**
- **hitCount**
- **isHit**
- **oldx**
- **oldy**

4.1.1 Detailed Description

This class represents the [Aquamentus](#) Boss.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `__init__()`

```
def actor.aquamentus.Aquamentus.__init__ (
    self,
    x,
    y )
```

Constructor for [Aquamentus](#).

Constructor takes two parameters, the x and y coordinates

Parameters

<i>x</i>	X coordinate of the starting postion of the Aquamentus
<i>y</i>	Y coordinate of the starting postion of the Aquamentus

4.1.3 Member Function Documentation

4.1.3.1 attack()

```
def actor.aquamentus.Aquamentus.attack (
    self )
```

Allows for [Aquamentus](#) to attack.

Spawns fireballs at Aquamenuts' mouth and sets their move speed

4.1.3.2 bossLogic()

```
def actor.aquamentus.Aquamentus.bossLogic (
    self )
```

Controls [Aquamentus](#) logic.

Uses the states to control the [Aquamentus](#)

4.1.3.3 checkState()

```
def actor.aquamentus.Aquamentus.checkState (
    self )
```

Evaluates the state of the [Aquamentus](#).

Evalautes if [Aquamentus](#) can stop, if it is in iframes, if it collides with something, and if it has died

4.1.3.4 swapDirection()

```
def actor.aquamentus.Aquamentus.swapDirection (
    self )
```

Swaps [Aquamentus](#)' direction.

Multiplies the speed in the x direction by -1

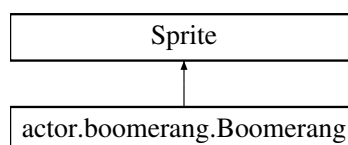
The documentation for this class was generated from the following file:

- [src/actor/aquamentus.py](#)

4.2 actor.boomerang.Boomerang Class Reference

[Boomerang](#) Weapon Class The class holding the creation, behaviour, and collision effects of the player's boomerang weapon.

Inheritance diagram for actor.boomerang.Boomerang:



Public Member Functions

- `def __init__ (self, x, y, direction, obj, player)`
[Boomerang](#) constructor A sprite subclass constructor that takes an x and y position (the players), a direction for the boomerang's trajectory, a list of collidable objects, as well as the player object (to add to the collision list)
- `def moveupdate (self)`
[Boomerang](#) position updating function, updating position based on changing trajectory speed.
- `def collisionupdate (self)`
[Boomerang](#) collision updating function, constantly checking for collisions and acting accordingly.
- `def update (self)`
[Boomerang](#) updating function, repeatedly running [moveupdate](#) and [collisionupdate](#).
- `def collision (self)`
Default collision function to satisfy other class collision calls to this object.

Public Attributes

- [image](#)
[Boomerang](#) sprite image.
- [rect](#)
Position of boomerang.
- [dir](#)
Initial travel direction of boomerang.
- [speed](#)
[Boomerang](#) initial update speed.
- [obj](#)
List of objects the boomerang could collide with.
- [killable](#)
Boolean to tell when the boomerang should be deleted.

4.2.1 Detailed Description

[Boomerang](#) Weapon Class The class holding the creation, behaviour, and collision effects of the player's boomerang weapon.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 `__init__()`

```
def actor.boomerang.Boomerang.__init__ (
    self,
    x,
    y,
    direction,
    obj,
    player )
```

[Boomerang](#) constructor A sprite subclass constructor that takes an x and y position (the players), a direction for the boomerang's trajectory, a list of collidable objects, as well as the player object (to add to the collision list)

Parameters

<i>x</i>	X coordinate of boomerang spawn
<i>y</i>	Y coordinate of boomerang spawn
<i>direction</i>	Direction of boomerang path
<i>obj</i>	List of objects to check for collision with the boomerang
<i>player</i>	Player object to also check collision for

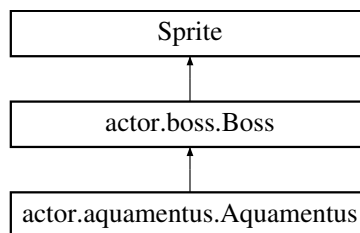
The documentation for this class was generated from the following file:

- `src/actor/boomerang.py`

4.3 actor.boss.Boss Class Reference

Superclass for representing a [Boss](#).

Inheritance diagram for actor.boss.Boss:



Public Member Functions

- `def __init__(self, x, y)`
Constructor for [Boss](#).
- `def checkState(self)`
Empty function for evaluating [Boss](#) state.
- `def move(self)`
Moves the [Boss](#).
- `def bossLogic(self)`
Empty function for logic of [Boss](#).
- `def update(self)`
Update loop for a [Boss](#).
- `def hit(self, dir)`
Hit detection for [Boss](#).

Public Attributes

- **image**
- **rect**
- **id**
- **isHit**
- **stuncount**
- **maxHP**
- **HP**
- **dmg**
- **hitCount**
- **xSpeed**
- **ySpeed**
- **frameCounter**

4.3.1 Detailed Description

Superclass for representing a [Boss](#).

4.3.2 Constructor & Destructor Documentation

4.3.2.1 `__init__()`

```
def actor.boss.Boss.__init__ (
    self,
    x,
    y )
```

Constructor for [Boss](#).

Constructor takes two parameters, the x and y coordinates

Parameters

<i>x</i>	X coordinate of the starting postion of the Boss
<i>y</i>	Y coordinate of the starting postion of the Boss

4.3.3 Member Function Documentation

4.3.3.1 `hit()`

```
def actor.boss.Boss.hit (
    self,
    dir )
```

Hit detection for [Boss](#).

Handles health and iframes

4.3.3.2 move()

```
def actor.boss.Boss.move (
    self )
```

Moves the [Boss](#).

Adds the x speed and y speed to the x and y position of the [Boss](#)

4.3.3.3 update()

```
def actor.boss.Boss.update (
    self )
```

Update loop for a [Boss](#).

Checks state, does the logic, and then moves [Boss](#)

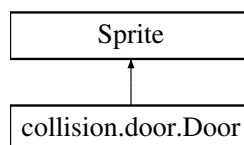
The documentation for this class was generated from the following file:

- src/actor/[boss.py](#)

4.4 collision.door.Door Class Reference

Dungeon [Door](#) Class This class is used to place a door on one of four places of a dungeon room, to allow the player to walk through and traverse to other rooms within the dungeon.

Inheritance diagram for collision.door.Door:



Public Member Functions

- def `__init__` (self, direction)
[Door](#) constructor, making a door on the specified side of the dungeon.
- def `collision` (self, i)

Public Attributes

- **image**
- **rect**
- **id**
- **state**

DOOR STATE 0 = open 1 = locked 2 = blocked (objective door)

4.4.1 Detailed Description

Dungeon [Door](#) Class This class is used to place a door on one of four places of a dungeon room, to allow the player to walk through and traverse to other rooms within the dungeon.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 `__init__()`

```
def collision.door.Door.__init__ (
    self,
    direction )
```

[Door](#) constructor, making a door on the specified side of the dungeon.

Parameters

<i>direction</i>	Integer value of the wall the door will be on (direction from centre of room)
------------------	-------------------------------------------------------------------------------

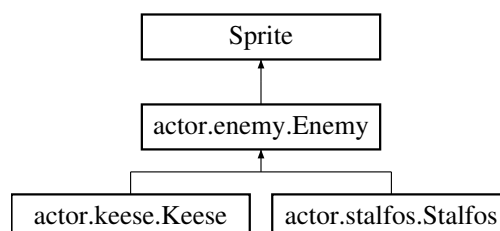
The documentation for this class was generated from the following file:

- [src/collision/door.py](#)

4.5 `actor.enemy.Enemy` Class Reference

Superclass for representing an [Enemy](#).

Inheritance diagram for `actor.enemy.Enemy`:



Public Member Functions

- `def __init__ (self, x, y)`
Constructor for [Enemy](#).
- `def checkState (self)`
Empty function for evaluating [Enemy](#) state.
- `def move (self)`
Moves the [Enemy](#).
- `def enemyLogic (self)`
Empty function for logic of [Enemy](#).
- `def update (self)`
Update loop for a [Enemy](#).
- `def hit (self, direc)`
Hit detection for [Enemy](#).

Public Attributes

- **image**
- **rect**
- **id**
- **isHit**
- **stuncount**
- **maxHP**
- **HP**
- **dmg**
- **hitCount**
This represents the buffer for the number of hits for the [Enemy](#) after being hit by player character.
- **hitdir**
Represents the direction [Enemy](#) is hit in by player character attack.
- **xSpeed**
- **ySpeed**
- **frameCounter**

4.5.1 Detailed Description

Superclass for representing an [Enemy](#).

4.5.2 Constructor & Destructor Documentation

4.5.2.1 __init__()

```
def actor.enemy.Enemy.__init__ (
    self,
    x,
    y )
```

Constructor for [Enemy](#).

Constructor takes two parameters, the x and y coordinates

Parameters

<i>x</i>	X coordinate of the starting postion of the Enemy
<i>y</i>	Y coordinate of the starting postion of the Enemy

4.5.3 Member Function Documentation

4.5.3.1 hit()

```
def actor.enemy.Enemy.hit (
    self,
    direc )
```

Hit dectetion for [Enemy](#).

Handles health, iframes and knockback direction

Parameters

<i>direc</i>	Direction of the knockback
--------------	----------------------------

4.5.3.2 move()

```
def actor.enemy.Enemy.move (
    self )
```

Moves the [Enemy](#).

Adds the x speed and y speed to the x and y postion of the [Enemy](#)

4.5.3.3 update()

```
def actor.enemy.Enemy.update (
    self )
```

Update loop for a [Enemy](#).

Checks state, does the logic, and then moves [Enemy](#)

4.5.4 Member Data Documentation

4.5.4.1 hitCount

```
actor.enemy.Enemy.hitCount
```

This represents the buffer for the number of hits for the [Enemy](#) after being hit by player character.

4.5.4.2 hitdir

```
actor.enemy.Enemy.hitdir
```

Represents the direction [Enemy](#) is hit in by player character attack.

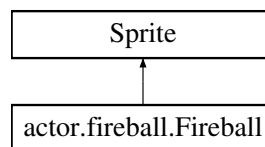
The documentation for this class was generated from the following file:

- [src/actor/enemy.py](#)

4.6 actor.fireball.Fireball Class Reference

This class represents the [Fireball](#) object.

Inheritance diagram for actor.fireball.Fireball:



Public Member Functions

- `def __init__ (self, x, y, xSpeed, ySpeed)`
Constructor for [Fireball](#).
- `def start (self, x, y, xs, ys)`
Starts the [Fireball](#)'s movement.
- `def end (self)`
Ends the [Fireball](#)'s movement.
- `def hit (self, dir)`
Empty function for being hit by player.
- `def checkState (self)`
Evaluates the state of the [Fireball](#).
- `def move (self)`
Moves the [Fireball](#).
- `def logic (self)`
Updates the [Fireball](#) sprite.
- `def update (self)`
Updates the [Fireball](#) every frame

Public Attributes

- **image**
- **rect**
- **id**
- **isHit**
- **dmg**
- **hitCount**
- **xSpeed**
- **ySpeed**
- **frameCounter**
- **obj**
- **sprites**
- **spriteIndex**

4.6.1 Detailed Description

This class represents the [Fireball](#) object.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 `__init__()`

```
def actor.fireball.Fireball.__init__ (
    self,
    x,
    y,
    xSpeed,
    ySpeed )
```

Constructor for [Fireball](#).

Constructor takes four parameters, the x and y coordinates and the x and y speeds

Parameters

<i>x</i>	X coordinate of the starting postion of the Fireball
<i>y</i>	Y coordinate of the starting postion of the Fireball
<i>xSpeed</i>	The speed in the x direction of the Fireball
<i>ySpeed</i>	The speed in the y direction of the Fireball

4.6.3 Member Function Documentation

4.6.3.1 checkState()

```
def actor.fireball.Fireball.checkState (
    self )
```

Evaluates the state of the [Fireball](#).

Checks for collision with walls, doors, and players

4.6.3.2 end()

```
def actor.fireball.Fireball.end (
    self )
```

Ends the [Fireball](#)'s movement.

Sets the x and y speed to 0 and places the [Fireball](#) of screen

4.6.3.3 hit()

```
def actor.fireball.Fireball.hit (
    self,
    dir )
```

Empty function for being hit by player.

Needs to exist for when player sword collides with [Fireball](#)

4.6.3.4 logic()

```
def actor.fireball.Fireball.logic (
    self )
```

Updates the [Fireball](#) sprite.

Swaps between the 2 sprites every 15 frames

4.6.3.5 move()

```
def actor.fireball.Fireball.move (
    self )
```

Moves the [Fireball](#).

Adds the x speed and y speed to the x and y position of the [Fireball](#)

4.6.3.6 start()

```
def actor.fireball.Fireball.start (
    self,
    x,
    y,
    xs,
    ys )
```

Starts the [Fireball](#)'s movement.

Sets the x and y positions and x and y speeds for [Fireball](#)

Parameters

<code>x</code>	X coordinate of where the Fireball is placed
<code>y</code>	Y coordinate of where the Fireball is placed
<code>xs</code>	The speed in the x direction of the Fireball
<code>ys</code>	The speed in the y direction of the Fireball

4.6.3.7 update()

```
def actor.fireball.Fireball.update (
    self )
```

Updates the [Fireball](#) every frame

Checks the state, applies the logic, and then moves the [Fireball](#)

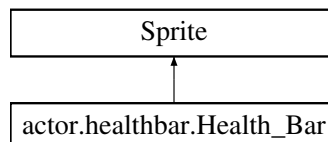
The documentation for this class was generated from the following file:

- [src/actor/fireball.py](#)

4.7 actor.healthbar.Health_Bar Class Reference

This class represents the HealthBar for the user controlled player character.

Inheritance diagram for actor.healthbar.Health_Bar:

**Public Member Functions**

- `def __init__ (self, x, y)`
Constructor HealthBar class.
- `def health (self, i)`
Method to update the current sprite image for the healthbar.

Public Attributes

- `image`
This represents the sprite image for the [Health_Bar](#) object.
- `h_sprite_sheet`
This represents the spritesheet for the image for the [Health_Bar](#) object, containing all sprites associated with the health bar.
- `rect`
This represents rectangle for position for the sprite image of the [Health_Bar](#) object.

4.7.1 Detailed Description

This class represents the HealthBar for the user controlled player character.

The HealthBar class uses the base class for visible game objects from Pygame library.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 __init__()

```
def actor.healthbar.Health_Bar.__init__ (
    self,
    x,
    y )
```

Constructor HealthBar class.

Constructor for class initializes the x and y location of the HealthBar object.

Parameters

<i>x</i>	this represents the x-coordinate at which the HealthBar object will be drawn.
<i>y</i>	this represents the y-coordinate at which the HealthBar object will be drawn.

4.7.3 Member Function Documentation

4.7.3.1 health()

```
def actor.healthbar.Health_Bar.health (
    self,
    i )
```

Method to update the current sprite image for the healthbar.

This method will allow the HealthBar to update as soon as the player character is damaged/receives health.

Parameters

<i>i</i>	value represening the number of heart sprites to render to screen.
----------	--------------------------------------------------------------------

4.7.4 Member Data Documentation

4.7.4.1 h_sprite_sheet

```
actor.healthbar.Health_Bar.h_sprite_sheet
```

This represents the spritesheet for the image for the [Health_Bar](#) object, containing all sprites associated with the health bar.

4.7.4.2 image

```
actor.healthbar.Health_Bar.image
```

This represents the sprite image for the [Health_Bar](#) object.

4.7.4.3 rect

```
actor.healthbar.Health_Bar.rect
```

This represents rectangle for position for the sprite image of the [Health_Bar](#) object.

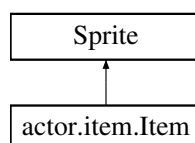
The documentation for this class was generated from the following file:

- [src/actor/healthbar.py](#)

4.8 actor.item.Item Class Reference

Consumable [Item](#) Class This class is used for the creation of a random consumable item, spawned once an enemy has been defeated on their last position before despawn.

Inheritance diagram for actor.item.Item:



Public Member Functions

- `def __init__ (self, x, y, typ)`
Item constructor A sprite subclass constructor which takes a pair of x-y coordinates, commonly those of the killed enemy, and an item type, in the form of a randomly generated integer from 0 to the # of possible items.
- `def collision (self, p)`
Collision handler for the player and the consumable item, depending on the item's type.

Public Attributes

- `image`
Item sprite image.
- `itemsprite`
List of possible item sprite images.
- `rect`
Item x and y position.
- `id`
Collision ID (how other items tell what this item is)
- `type`
Integer value to determine what kind of item this is (rupee, heart, etc)

4.8.1 Detailed Description

Consumable *Item* Class This class is used for the creation of a random consumable item, spawned once an enemy has been defeated on their last position before despawn.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 __init__()

```
def actor.item.Item.__init__ (
    self,
    x,
    y,
    typ )
```

Item constructor A sprite subclass constructor which takes a pair of x-y coordinates, commonly those of the killed enemy, and an item type, in the form of a randomly generated integer from 0 to the # of possible items.

Parameters

<i>x</i>	X coordinate of the spawned item
<i>y</i>	Y coordinate of the spawned item
<i>typ</i>	Integer value to specify the type of consumable item

4.8.3 Member Function Documentation

4.8.3.1 collision()

```
def actor.item.Item.collission (
    self,
    p )
```

Collision handler for the player and the consumable item, depending on the item's type.

Parameters

<i>p</i>	Player object
----------	---------------

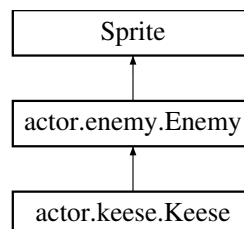
The documentation for this class was generated from the following file:

- [src/actor/item.py](#)

4.9 actor.keese.Keese Class Reference

This class represents the [Keese](#) enemy.

Inheritance diagram for actor.keese.Keese:



Public Member Functions

- `def __init__ (self, x, y)`
Constructor for [Keese](#).
- `def genRestLength (self)`
Sets the [Keese](#) rest length.
- `def genTravelPoint (self)`
Creates a new travel point for [Keese](#).
- `def switchSprite (self)`
Iterates through the sprite list.
- `def stop (self)`
Stops the [Keese](#).
- `def setMoveSpeed (self)`
Sets the [Keese](#) x and y movement speed.
- `def checkState (self)`
Evaluates the state of the [Keese](#).
- `def enemyLogic (self)`
Controls [Keese](#) logic.

Public Attributes

- **isMoving**
- **canStop**
- **isResting**
- **travelPoint**
- **restTime**
- **restStartFrame**
- **flyStartFrame**
- **spriteIndex**
- **maxHP**
- **HP**
- **dmg**
- **image**
- **sprites**
- **xSpeed**
- **ySpeed**
- **hitCount**
- **isHit**

4.9.1 Detailed Description

This class represents the [Keese](#) enemy.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 `__init__()`

```
def actor.keese.Keese.__init__ (
    self,
    x,
    y )
```

Constructor for [Keese](#).

Constructor takes two parameters, the x and y coordinates

Parameters

x	X coordinate of the starting postion of the Keese
y	Y coordinate of the starting postion of the Keese

4.9.3 Member Function Documentation

4.9.3.1 checkState()

```
def actor.keese.Keese.checkState (
    self )
```

Evaluates the state of the [Keese](#).

Evaluates if the [Keese](#) is moving, if it can stop, if it is in iframes, and if it has died

4.9.3.2 enemyLogic()

```
def actor.keese.Keese.enemyLogic (
    self )
```

Controls [Keese](#) logic.

Uses the states to control the [Keese](#)

4.9.3.3 genRestLength()

```
def actor.keese.Keese.genRestLength (
    self )
```

Sets the [Keese](#) rest length.

Generates a random number between [1,2] inclusive as the rest time between movements and sets the rest start frame to the current frame

4.9.3.4 genTravelPoint()

```
def actor.keese.Keese.genTravelPoint (
    self )
```

Creates a new travel point for [Keese](#).

Generate a random point to move to between 0 and the width - 30 of the screen for the x coordinate and between 0 and the height - 30 of the screen for the y coordinate

4.9.3.5 setMoveSpeed()

```
def actor.keese.Keese.setMoveSpeed (
    self )
```

Sets the [Keese](#) x and y movement speed.

Compares the two lengths of travel (x and y) and sets the speed in which ever direction is longer to the max speed and the other to a scalar multiple of the max speed based on the ratio of the two lengths

4.9.3.6 stop()

```
def actor.keese.Keese.stop (
    self )
```

Stops the [Keese](#).

Sets the [Keese](#) speed in the x direction and the y direction to zero

4.9.3.7 switchSprite()

```
def actor.keese.Keese.switchSprite (
    self )
```

Iterates through the sprite list.

Swaps between the two sprites that are available for the keese

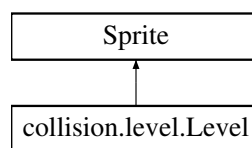
The documentation for this class was generated from the following file:

- [src/actor/keese.py](#)

4.10 collision.level.Level Class Reference

Dungeon level background class Creates the background of a dungeon, can be modified in the future to change colour based on sprite sheet and new arg.

Inheritance diagram for collision.level.Level:



Public Member Functions

- `def __init__ (self)`
Background initializer Initialize a background to constantly be printed as game background.

Public Attributes

- `image`
- `dungeon_sprite`
- `rect`

4.10.1 Detailed Description

Dungeon level background class Creates the background of a dungeon, can be modified in the future to change colour based on sprite sheet and new arg.

The documentation for this class was generated from the following file:

- [src/collision/level.py](#)

4.11 collision.levelmanager.LevelManager Class Reference

Dungeon Level Creation Class Class used as a master constructor for every dungeon levels, getting pre-written data and loading it when the game starts and when a transition occurs.

Public Member Functions

- `def __init__ (self, spritelist, collidlist, updatelist)`
- `def make (self, x, y)`
- `def transition (self, xchange, ychange)`

Dungeon level transition function, used to go to an adjacent level, clear all current data, and load new room data.

Public Attributes

- `sl`
- `cl`
- `ul`
- `x`
- `y`
- `level`

4.11.1 Detailed Description

Dungeon Level Creation Class Class used as a master constructor for every dungeon levels, getting pre-written data and loading it when the game starts and when a transition occurs.

4.11.2 Member Function Documentation

4.11.2.1 transition()

```
def collision.levelmanager.LevelManager.transition (
    self,
    xchange,
    ychange )
```

Dungeon level transition function, used to go to an adjacent level, clear all current data, and load new room data.

Parameters

<i>j</i>	Value to add to the current index to get to new room
----------	------------------------------------------------------

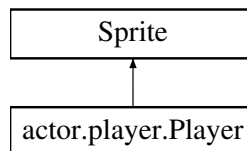
The documentation for this class was generated from the following file:

- [src/collision/levelmanager.py](#)

4.12 actor.player.Player Class Reference

Player Class A pygame sprite subclass for defining the creation of the game's playable character, as well as its interactions with both the user and other entities within the game.

Inheritance diagram for actor.player.Player:



Public Member Functions

- `def __init__ (self, x, y, hbar)`
Player constructor The constructor for the player object, used within the initialization of the game.
- `def movechar (self)`
Function to move the player in a specified direction.
- `def move (self, d, b)`
Function to store which movement buttons are being pressed and let go of by the user.
- `def attack (self)`
Function to react to the user pressing the attack button, putting the player into an attack animation and spawning a sword for the duration.
- `def useitem (self)`
Function to react to the user pressing the use item button, putting the player into an attack animation and spawning a boomerang for the duration.
- `def moveupdate (self)`
UPDATE LOOP FUNCTIONS.
- `def attackupdate (self)`
Update function for attack and boomerang conditions, as well as handing user collision with enemies (damage taken) Checks sword collision on enemies, and checks the timeout on both the sword and boomerang attack animations, telling the player when they can move again.
- `def collisionupdate (self)`
Update function to check collisions between the player and any specified collidable objects, and call their respective collision events.
- `def update (self)`
Player master update function, running all update functions repeatedly.

Public Attributes

- [image](#)
Player sprite image.
- [id](#)
Collision ID (To tell other objects what this object is)
- [leveltrans](#)
Boolean value to tell the game when the player has left the room, and a room transition is needed.
- [rect](#)
Player x and y position.
- [walkLeft](#)
Animation sprites for player walking left.
- [walkRight](#)
Animation sprites for player walking right.
- [walkUp](#)
Animation sprites for player walking up.
- [walkDown](#)
Animation sprites for player walking down.
- [attacksprite](#)
Animation sprites for player attacking.
- [obj](#)
List of objects that the player can collide with.
- [hbar](#)
Health bar corresponding to the player's health.
- [totalhp](#)
Player's maximum HP amount.
- [hp](#)
Player's current HP amount (starts at max)
- [rupes](#)
Player's rupee counter.
- [keys](#)
Player's key counter.
- [moveable](#)
Boolean allowing and stoping the player from moving.
- [attackbool](#)
Boolean to tell whether the player is attacking or not.
- [attackcount](#)
Attack counter, to control how long the attack lasts.
- [attacksword](#)
Holder for a sword object when attacking.
- [item](#)
Holder for a boomerang object when using item.
- [itembool](#)
Boolean to tell whether the player is using their boomerang or not.
- [oldx](#)
Player's x value one frame ago.
- [oldy](#)
Player's y value one frame ago.
- [dirbool](#)
Boolean array to tell which directional buttons are pressed or not (starting left going clockwise, dirbool[0] is left, dirbool[1] is up, etc)

- [dir](#)
Current direction the player is facing.
- [hit](#)
Boolean on whether the player was recently hit.
- [hitcount](#)
How long a hit counts for on the player (how long until self.hit turns off)

4.12.1 Detailed Description

[Player](#) Class A pygame sprite subclass for defining the creation of the game's playable character, as well as its interactions with both the user and other entities within the game.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 `__init__()`

```
def actor.player.Player.__init__ (
    self,
    x,
    y,
    hbar )
```

[Player](#) constructor The constructor for the player object, used within the initialization of the game.

The only arguments passed are an initial x and y position, as well as heads-up-display (HUD) elements. param x Initial player x coordinate param y Initial player y coordinate param hbar HUD health bar, showing the current health of the player

4.12.3 Member Function Documentation

4.12.3.1 `attackupdate()`

```
def actor.player.Player.attackupdate (
    self )
```

Update function for attack and boomerang conditions, as well as handing user collision with enemies (damage taken) Checks sword collision on enemies, and checks the timeout on both the sword and boomerang attack animations, telling the player when they can move again.

Also updates the player and reacts accordingly to receiving damage.

4.12.3.2 `move()`

```
def actor.player.Player.move (
    self,
    d,
    b )
```

Function to store which movement buttons are being pressed and let go of by the user.

Parameters

<i>d</i>	Direction related to the button pressed (integer value from 0 to 3, starting from the left and going clockwise)
<i>b</i>	Boolean value on whether the button is pressed or not

4.12.3.3 moveupdate()

```
def actor.player.Player.moveupdate (
    self )
```

UPDATE LOOP FUNCTIONS.

Update function for player movement and according sprite animation

4.12.4 Member Data Documentation

4.12.4.1 attacksword

```
actor.player.Player.attacksword
```

Holder for a sword object when attacking.

Sprite function to delete a sprite object and all relations of the object in the project.

4.12.4.2 dir

```
actor.player.Player.dir
```

Current direction the player is facing.

Render Appropriate Sprites According to movement.

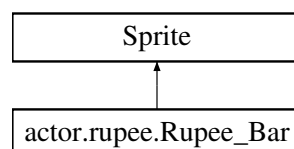
The documentation for this class was generated from the following file:

- [src/actor/player.py](#)

4.13 actor.rupee.Rupee_Bar Class Reference

This class represents the RupeeBar object.

Inheritance diagram for actor.rupee.Rupee_Bar:



Public Member Functions

- `def __init__ (self, x, y)`
Constructor for the RupeeBar object.

Public Attributes

- `image`
This represents the sprite image of the [Rupee_Bar](#) object.
- `rect`
This represents rectangle for collision and position for the sprite image of the [Rupee_Bar](#) object.

4.13.1 Detailed Description

This class represents the RupeeBar object.

The Rupee class uses the Pygame library and SpriteSheet module to create an image for the

4.13.2 Constructor & Destructor Documentation

4.13.2.1 __init__()

```
def actor.rupee.Rupee_Bar.__init__ (
    self,
    x,
    y )
```

Constructor for the RupeeBar object.

Constructor for class initializes the x and y location of the RupeeBar object.

Parameters

<code>x</code>	this represents the x-coordinate at which the RupeeBar object will be drawn.
<code>y</code>	this represents the y-coordinate at which the RupeeBar object will be drawn.

4.13.3 Member Data Documentation

4.13.3.1 image

```
actor.rupee.Rupee_Bar.image
```

This represents the sprite image of the [Rupee_Bar](#) object.

4.13.3.2 `rect`

```
actor.rupee.Rupee_Bar.rect
```

This represents rectangle for collision and position for the sprite image of the [Rupee_Bar](#) object.

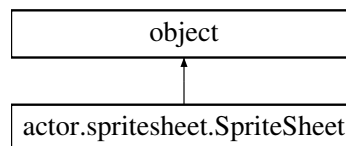
The documentation for this class was generated from the following file:

- [src/actor/rupee.py](#)

4.14 `actor.spritesheet.SpriteSheet` Class Reference

This class represents the [SpriteSheet](#) object, allowing sprites to be loaded and processed.

Inheritance diagram for `actor.spritesheet.SpriteSheet`:



Public Member Functions

- `def __init__(self, file_name)`
Constructor for the [SpriteSheet](#) class.
- `def get_image(self, x, y, width, height)`
Accessor to return an image splice from the loaded [SpriteSheet](#).
- `def get_imageNT(self, x, y, width, height)`
Accessor to return an image splice from the loaded [SpriteSheet](#), WITHOUT removing the image background.

Public Attributes

- `sprite_sheet`
This represents the current spritesheet loaded from the specific `file_name` path.

4.14.1 Detailed Description

This class represents the [SpriteSheet](#) object, allowing sprites to be loaded and processed.

The [SpriteSheet](#) class uses the Pygame library to load images and change the transparency on those images.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 `__init__()`

```
def actor.spritesheet.SpriteSheet.__init__ (
    self,
    file_name )
```

Constructor for the [SpriteSheet](#) class.

This constructor initializes an image file using the Pygame library.

Parameters

<i>file_name</i>	This is the string representing the path to the image file.
------------------	-------------------------------------------------------------

4.14.3 Member Function Documentation

4.14.3.1 `get_image()`

```
def actor.spritesheet.SpriteSheet.get_image (
    self,
    x,
    y,
    width,
    height )
```

Accessor to return an image splice from the loaded [SpriteSheet](#).

This accessor returns an image splice based on the x,y location and the height and width of the image.

Parameters

<i>x</i>	The x-coordinate for starting point of the splice on the SpriteSheet .
<i>y</i>	The y-coordinate for starting point of the splice on the SpriteSheet .
<i>width</i>	The width of the splice on the SpriteSheet .
<i>height</i>	The height of the splice on the SpriteSheet .

Returns

image This returns the newly spliced image after removing the image background transparency.

4.14.3.2 `get_imageNT()`

```
def actor.spritesheet.SpriteSheet.get_imageNT (
    self,
    x,
    y,
    width,
    height )
```

Accessor to return an image splice from the loaded [SpriteSheet](#), WITHOUT removing the image background.

This accessor returns an image splice based on the x,y location and the height and width of the image.

Parameters

<i>x</i>	The x-coordinate for starting point of the splice on the SpriteSheet .
<i>y</i>	The y-coordinate for starting point of the splice on the SpriteSheet .
<i>width</i>	The width of the splice on the SpriteSheet .
<i>height</i>	The height of the splice on the SpriteSheet .

Returns

image This returns the newly spliced image WITHOUT removing the image background transparency.

4.14.4 Member Data Documentation

4.14.4.1 `sprite_sheet`

```
actor.spritesheet.SpriteSheet.sprite_sheet
```

This represents the current spritesheet loaded from the specific `file_name` path.

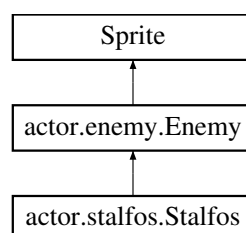
The documentation for this class was generated from the following file:

- `src/actor/spritesheet.py`

4.15 `actor.stalfos.Stalfos` Class Reference

This class represents the [Stalfos](#) enemy.

Inheritance diagram for `actor.stalfos.Stalfos`:



Public Member Functions

- def `__init__` (self, x, y)
Constructor for [Stalfos](#).
- def `checkState` (self)
Evaluates the state of the [Stalfos](#).
- def `enemyLogic` (self)
Controls [Stalfos](#) logic.
- def `genTravelPath` (self)
Creates a new travel point for [Stalfos](#).
- def `setWalkSpeed` (self)
Sets the walk speed of the [Stalfos](#).
- def `stop` (self)
Stops the [Stalfos](#).

Public Attributes

- `isMoving`
Call superclass constructor.
- `previousDirection`
Movement value for previous direction of movement for [Stalfos](#) state.
- `direction`
Movement value for current direction of movement for [Stalfos](#) state.
- `walkFrames`
Number representing the frames walked by [Stalfos](#) in movement state.
- `walkStartFrame`
Movement value for starting frame for [Stalfos](#) in movement state.
- `oldx`
This represents the previous x-location of [Stalfos](#) in movement/stationary state.
- `oldy`
This represents the previous y-location of [Stalfos](#) in movement/stationary state.
- `obj`
This represents the list of objects [Stalfos](#) can collide with, in movement/stationary state.
- `maxHP`
Set [Stalfos](#) health to maximum health.
- `HP`
Current [Stalfos](#) health.
- `dmg`
The current damage value that the [Stalfos](#) object has received from player character.
- `image`
Load the sprite sheet.
- `sprites`
Creates a list of sprites list for [Stalfos](#).
- `spriteIndex`
Index for the current index in the sprite list for [Stalfos](#).
- `hitdir`
- `hitCount`
- `isHit`
Represents the current state if [Stalfos](#) has collided with player character attack.
- `xSpeed`
The current x-directional speed for [Stalfos](#) in movement/stationary state.
- `ySpeed`
The current y-directional speed for [Stalfos](#) in movement/stationary state.

4.15.1 Detailed Description

This class represents the [Stalfos](#) enemy.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 `__init__()`

```
def actor.stalfos.Stalfos.__init__ (
    self,
    x,
    y )
```

Constructor for [Stalfos](#).

Constructor takes two parameters, the x and y coordinates

Parameters

x	X coordinate of the starting postion of the Keese
y	Y coordinate of the starting postion of the Keese

4.15.3 Member Function Documentation

4.15.3.1 `checkState()`

```
def actor.stalfos.Stalfos.checkState (
    self )
```

Evaluates the state of the [Stalfos](#).

Evalautes if the [Stalfos](#) if moving, if it can stop, if it is in iframes, if it collides with something, and if it has died

4.15.3.2 `enemyLogic()`

```
def actor.stalfos.Stalfos.enemyLogic (
    self )
```

Controls [Stalfos](#) logic.

Uses the states to control the [Stalfos](#)

4.15.3.3 genTravelPath()

```
def actor.stalfos.Stalfos.genTravelPath (
    self )
```

Creates a new travel point for [Stalfos](#).

Generates a direction to walk and the distance to move

4.15.3.4 setWalkSpeed()

```
def actor.stalfos.Stalfos.setWalkSpeed (
    self )
```

Sets the walk speed of the [Stalfos](#).

Sets speed based on the direction

4.15.3.5 stop()

```
def actor.stalfos.Stalfos.stop (
    self )
```

Stops the [Stalfos](#).

Sets the [Stalfos](#) speed in the x direction and the y direction to zero

4.15.4 Member Data Documentation

4.15.4.1 direction

```
actor.stalfos.Stalfos.direction
```

Movement value for current direction of movement for [Stalfos](#) state.

4.15.4.2 dmg

```
actor.stalfos.Stalfos.dmg
```

The current damage value that the [Stalfos](#) object has received from player character.

4.15.4.3 HP

```
actor.stalfos.Stalfos.HP
```

Current [Stalfos](#) health.

4.15.4.4 image

```
actor.stalfos.Stalfos.image
```

Load the sprite sheet.

This represents the sprite image for the [Stalfos](#) object.

4.15.4.5 isHit

```
actor.stalfos.Stalfos.isHit
```

Represents the current state if [Stalfos](#) has collided with player character attack.

4.15.4.6 isMoving

```
actor.stalfos.Stalfos.isMoving
```

Call superclass constructor.

Check for movement.

Boolean values for keeping track of the [Stalfos](#) state.

4.15.4.7 maxHP

```
actor.stalfos.Stalfos.maxHP
```

Set [Stalfos](#) health to maximum health.

4.15.4.8 obj

```
actor.stalfos.Stalfos.obj
```

This represents the list of objects [Stalfos](#) can collide with, in movement/stationary state.

4.15.4.9 oldx

```
actor.stalfos.Stalfos.oldx
```

This represents the previous x-location of [Stalfos](#) in movement/stationary state.

4.15.4.10 oldy

```
actor.stalfos.Stalfos.oldy
```

This represents the previous y-location of [Stalfos](#) in movement/stationary state.

4.15.4.11 previousDirection

```
actor.stalfos.Stalfos.previousDirection
```

Movement value for previous direction of movement for [Stalfos](#) state.

4.15.4.12 spriteIndex

```
actor.stalfos.Stalfos.spriteIndex
```

Index for the current index in the sprite list for [Stalfos](#).

4.15.4.13 sprites

```
actor.stalfos.Stalfos.sprites
```

Creates a list of sprites list for [Stalfos](#).

4.15.4.14 walkFrames

```
actor.stalfos.Stalfos.walkFrames
```

Number representing the frames walked by Stalfos in movement state.

4.15.4.15 walkStartFrame

```
actor.stalfos.Stalfos.walkStartFrame
```

Movement value for starting frame for [Stalfos](#) in movement state.

4.15.4.16 xSpeed

```
actor.stalfos.Stalfos.xSpeed
```

The current x-directional speed for [Stalfos](#) in movement/stationary state.

4.15.4.17 ySpeed

```
actor.stalfos.Stalfos.ySpeed
```

The current y-directional speed for [Stalfos](#) in movement/stationary state.

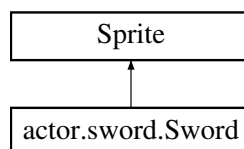
The documentation for this class was generated from the following file:

- [src/actor/stalfos.py](#)

4.16 actor.sword.Sword Class Reference

Player [Sword](#) Class Class for the creation and deletion of the sword sprite object, made when the player attacks.

Inheritance diagram for actor.sword.Sword:



Public Member Functions

- `def __init__(self, x, y, direction)`
[Sword](#) constructor, taking an x and y coordinate, and the direction for the sword to be pointing (player direction)

Public Attributes

- [sprite](#)
Array of all possible sword directions (following usual directional standards, 0-3, starting at left going clockwise)
- [image](#)
Sprite image of the sword.
- [rect](#)
X and Y position of the sword.

4.16.1 Detailed Description

Player [Sword](#) Class Class for the creation and deletion of the sword sprite object, made when the player attacks.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 `__init__()`

```
def actor.sword.Sword.__init__ (
    self,
    x,
    y,
    direction )
```

[Sword](#) constructor, taking an x and y coordinate, and the direction for the sword to be pointing (player direction)

Parameters

<i>x</i>	Sword 's x coordinate
<i>y</i>	Sword 's y coordinate
<i>direction</i>	Sword 's direction (integer from 0 to 3, starting left, going clockwise)

4.16.3 Member Data Documentation

4.16.3.1 `image`

```
actor.sword.Sword.image
```

Sprite image of the sword.

Below is for testing `self.image = pygame.Surface([26, 14]) self.image.fill((200, 0, 0))`

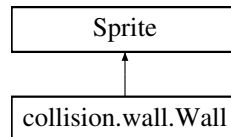
The documentation for this class was generated from the following file:

- `src/actor/sword.py`

4.17 collision.wall.Wall Class Reference

This class represents the [Wall](#) class for collision for objects in the environment.

Inheritance diagram for collision.wall.Wall:



Public Member Functions

- `def __init__ (self, x, y, w, h, sprite)`
Constructor for the [Wall](#) class.
- `def collision (self, i)`
This method checks for collision with the [Wall](#) object and other sprite object.

Public Attributes

- [image](#)
This represents the sprite image for [Wall](#) object.
- [rect](#)
This represents rectangle for position for the sprite image of the [Wall](#) object.
- [id](#)
*This represents the ID for the [Wall](#) object, ised for collision in **main.py**.*

4.17.1 Detailed Description

This class represents the [Wall](#) class for collision for objects in the environment.

The [Wall](#) class uses the base class for visible game objects from Pygame library.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 `__init__()`

```
def collision.wall.Wall.__init__ (
    self,
    x,
    y,
    w,
    h,
    sprite )
```

Constructor for the [Wall](#) class.

Constructor for [Wall](#) class initializes a [Wall](#) object based on the x/y location and the width/height of the wall, and the sprite the collision for the wall will be created upon.

Parameters

<i>x</i>	This represents the x-location for the Wall object to be created.
<i>y</i>	This represents the y-location for the Wall object to be created.
<i>w</i>	This represents the width of the Wall object when created.
<i>h</i>	This represents the height of the Wall object when created.
<i>sprite</i>	This represents the sprite that the collision for a Wall will be present upon at all times.

4.17.3 Member Function Documentation

4.17.3.1 collision()

```
def collision.wall.Wall.collision (
    self,
    i )
```

This method checks for collision with the [Wall](#) object and other sprite object.

The collision will be checked with the [Wall](#) and other sprite object as the object collides with the wall.

Parameters

<i>i</i>	This is the sprite object that is passed into the method, and checks if the object is colliding with the Wall object, resetting the sprite objects location accordingly.
----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.17.4 Member Data Documentation

4.17.4.1 id

```
collision.wall.Wall.id
```

This represents the ID for the [Wall](#) object, used for collision in **main.py**.

4.17.4.2 image

```
collision.wall.Wall.image
```

This represents the sprite image for [Wall](#) object.

4.17.4.3 rect

`collision.wall.Wall.rect`

This represents rectangle for position for the sprite image of the [Wall](#) object.

The documentation for this class was generated from the following file:

- [src/collision/wall.py](#)

Chapter 5

File Documentation

5.1 src/actor/aquamentus.py File Reference

Aquamentus Boss

Classes

- class `actor.aquamentus.Aquamentus`
This class represents the [Aquamentus](#) Boss.

Functions

- def `actor.aquamentus.populateSprites ()`
Creates the sprite list for [Aquamentus](#).

Variables

- string `actor.aquamentus.SPRITE_MAP` = 'src/actor/sprites/aquamentus.png'

5.1.1 Detailed Description

Aquamentus Boss

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 8 2018

5.1.2 Function Documentation

5.1.2.1 populateSprites()

```
def actor.aquamentus.populateSprites ( )
```

Creates the sprite list for Aquamentus.

Iterates through a sprite sheet to pull images for the sprite array

Returns

sprites Array if the sprites that represent the Aquamentus

5.2 src/actor/boomerang.py File Reference

Boomerang Weapon

Classes

- class [actor.boomerang.Boomerang](#)
Boomerang Weapon Class The class holding the creation, behaviour, and collision effects of the player's boomerang weapon.

5.2.1 Detailed Description

Boomerang Weapon

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 6 2018

5.3 src/actor/boss.py File Reference

Boss Template

Classes

- class [actor.boss.Boss](#)
Superclass for representing a Boss.

5.3.1 Detailed Description

Boss Template

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.4 src/actor/constants.py File Reference

Actor Constants

Variables

- tuple `actor.constants.BLACK` = (0,0,0)
Defines the colour black.
- tuple `actor.constants.WHITE` = (255, 255, 255)
Defines the colour white.
- tuple `actor.constants.PURPLE` = (255, 50, 255)
Defines the colour purple.
- tuple `actor.constants.RED` = (255, 50, 50)
Defines the colour red.
- int `actor.constants.PLAYER_WIDTH` = 15
Defines the player's width.
- int `actor.constants.PLAYER_HEIGHT` = 15
Defines the player's height.
- int `actor.constants.PLAYER_SPEED` = 3
Defines the player's speed.
- int `actor.constants.PLAYER_MAX_HP` = 3
Defines the player's max hp.
- int `actor.constants.HIT_SPEED` = 12
Defines the speed of the knock back applied per frame.
- int `actor.constants.HIT_TIME` = 5
Defines the number of frames hit speed is applied for.
- int `actor.constants.HIT_IFRAME` = 30
Defines the number of iframes.
- int `actor.constants.ATK_WIDTH` = 15
Defines the width of the attack hitbox.
- int `actor.constants.ATK_HEIGHT` = 15
Defines the height of the attack hitbox.
- int `actor.constants.ATK_LENGTH` = 10
Defines the number of frames the hit stays out for.
- int `actor.constants.ATK_BUFFER` = 20
Defines the number of frames before the next attack can start.

- int `actor.constants.BOOM_SPEED` = 9
Defines the bommerang speed.
- float `actor.constants.BOOM_RETURN` = 0.3
Defines the acceleration in the opposite direction of the bommerang's travel path.
- int `actor.constants.GLOBAL_FRAME_BUFFER` = 12
Coordinate Divisor that determines range in which next sprite should render.
- int `actor.constants.KEESE_MAX_HP` = 1
Defines the keese's max hp.
- float `actor.constants.KEESE_DMG` = 0.5
Defines the keese's damage.
- int `actor.constants.KEESE_MAX_SPEED` = 2
Defines the keese's max speed.
- int `actor.constants.KEESE_MIN_SPEED` = 1
Defines the keese's min speed.
- int `actor.constants.ACCEPTABLE_RADIUS` = 20
Defines the keese's minimum magnitude from its determined travel point before stopping.
- int `actor.constants.KEESE_MAGNITUDE_MIN` = 200
Defines the keese's minimum magnitude for selecting its next travel point.
- int `actor.constants.STALFOS_MAX_HP` = 2
Defines the stalfos' max hp.
- int `actor.constants.STALFOS_DMG` = 1
Defines the stalfos' damage.
- int `actor.constants.STALFOS_SPEED` = 1
Defines the stalfos' speed.
- int `actor.constants.STALFOS_HIT_SPEED` = 10
Defines the stalfos' speed during knockback.
- int `actor.constants.ENEMY_STUNCOUNT` = 30
Length of frames an enemy will be stunned for when hit by a boomerang.
- int `actor.constants.AQUAMENTUS_MAX_HP` = 10
Defines the aquamentus' max hp.
- int `actor.constants.AQUAMENTUS_DMG` = 2
Defines the aquamentus' damage.
- int `actor.constants.AQUAMENTUS_SPEED` = 1
Defines the aquamentus' speed.
- int `actor.constants.FIREBALL_DMG` = 1
Defines the fireballs damage.

5.4.1 Detailed Description

Actor Constants

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.5 src/actor/enemy.py File Reference

Enemy Template

Classes

- class [actor.enemy.Enemy](#)
Superclass for representing an [Enemy](#).

5.5.1 Detailed Description

Enemy Template

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.6 src/actor/fireball.py File Reference

Fireball object

Classes

- class [actor.fireball.Fireball](#)
This class represents the [Fireball](#) object.

Functions

- def [actor.fireball.populateSprites](#) ()
Creates the sprite list for [Fireball](#).

Variables

- string **actor.fireball.SPRITE_MAP** = 'src/actor/sprites/fireball.png'

5.6.1 Detailed Description

Fireball object

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.6.2 Function Documentation

5.6.2.1 populateSprites()

```
def actor.fireball.populateSprites ( )
```

Creates the sprite list for Fireball.

Iterates through a sprite sheet to pull images for the sprite array

Returns

sprites Array if the sprites that represent the Fireball

5.7 src/actor/healthbar.py File Reference

HealthBar Class

Classes

- class [actor.healthbar.Health_Bar](#)

This class represents the HealthBar for the user controlled player character.

5.7.1 Detailed Description

HealthBar Class

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.8 src/actor/item.py File Reference

Consumable Items

Classes

- class [actor.item.Item](#)

Consumable [Item](#) Class This class is used for the creation of a random consumable item, spawned once an enemy has been defeated on their last position before despawn.

5.8.1 Detailed Description

Consumable Items

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 6 2018

5.9 src/actor/keese.py File Reference

Keese Enemy

Classes

- class [actor.keese.Keese](#)

This class represents the [Keese](#) enemy.

Functions

- def [actor.keese.populateSprites](#) ()

Creates the sprite list for [Keese](#).

Variables

- string [actor.keese.SPRITE_MAP](#) = 'src/actor/sprites/keese.png'

5.9.1 Detailed Description

Keese Enemy

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.9.2 Function Documentation

5.9.2.1 populateSprites()

```
def actor.keese.populateSprites ( )
```

Creates the sprite list for Keese.

Iterates through a sprite sheet to pull images for the sprite array

Returns

sprites Array if the sprites that represent the Keese

5.10 src/actor/player.py File Reference

Playable Characer

Classes

- class [actor.player.Player](#)

Player Class A pygame sprite subclass for defining the creation of the game's playable character, as well as its interactions with both the user and other entities within the game.

5.10.1 Detailed Description

Playable Characer

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 6 2018

5.11 src/actor/rupee.py File Reference

Rupee_Bar Class

Classes

- class [actor.rupee.Rupee_Bar](#)
This class represents the RupeeBar object.

5.11.1 Detailed Description

Rupee_Bar Class

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.12 src/actor/spritesheet.py File Reference

SpriteSheet Class

Classes

- class [actor.spritesheet.SpriteSheet](#)
This class represents the [SpriteSheet](#) object, allowing sprites to be loaded and processed.

5.12.1 Detailed Description

SpriteSheet Class

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.13 src/actor/stalfos.py File Reference

Stalfos Enemy

Classes

- class `actor.stalfos.Stalfos`
This class represents the [Stalfos](#) enemy.

Functions

- def `actor.stalfos.populateSprites ()`
Creates the sprite list for [Stalfos](#).

Variables

- string `actor.stalfos.SPRITE_MAP` = 'src/actor/sprites/stalfos.png'

5.13.1 Detailed Description

Stalfos Enemy

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.13.2 Function Documentation

5.13.2.1 `populateSprites()`

```
def actor.stalfos.populateSprites ( )
```

Creates the sprite list for Stalfos.

Iterates through a sprite sheet to pull images for the sprite array

Returns

sprites Array if the sprites that represent the Stalfos

5.14 `src/actor/sword.py` File Reference

Player Sword

Classes

- class [actor.sword.Sword](#)

Player [Sword](#) Class Class for the creation and deletion of the sword sprite object, made when the player attacks.

5.14.1 Detailed Description

Player Sword

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 8 2018

5.15 src/collision/door.py File Reference

Dungeon Door

Classes

- class [collision.door.Door](#)

Dungeon [Door](#) Class This class is used to place a door on one of four places of a dungeon room, to allow the player to walk through and traverse to other rooms within the dungeon.

5.15.1 Detailed Description

Dungeon Door

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.16 src/collision/level.py File Reference

Dungeon Background

Classes

- class [collision.level.Level](#)

Dungeon level background class Creates the background of a dungeon, can be modified in the future to change colour based on sprite sheet and new arg.

5.16.1 Detailed Description

Dungeon Background

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.17 [src/collision/levelmanager.py](#) File Reference

Dungeon Level Master Creator

Classes

- class [collision.levelmanager.LevelManager](#)

Dungeon Level Creation Class Class used as a master constructor for every dungeon levels, getting pre-written data and loading it when the game starts and when a transition occurs.

5.17.1 Detailed Description

Dungeon Level Master Creator

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.18 [src/collision/wall.py](#) File Reference

Wall Class

Classes

- class [collision.wall.Wall](#)

This class represents the [Wall](#) class for collision for objects in the environment.

5.18.1 Detailed Description

Wall Class

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.19 src/config/colour.py File Reference

Colour Constants

Variables

- tuple [config.colour.WHITE](#) = (255, 255, 255)

Defines the colour white.

5.19.1 Detailed Description

Colour Constants

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

5.20 src/config/window.py File Reference

Window Constants

Variables

- int `config.window.Y_OFFSET` = 56
Y offset for HUD.
- int `config.window.Wwidth` = 480
Width of the window.
- int `config.window.Wheight` = 320 + `Y_OFFSET`
Height of the window.

5.20.1 Detailed Description

Window Constants

Author

Giacomo Loparco, Bilal Jaffry, Lucas Zacharewicz

Date

November 7 2018

Index

- `__init__`
 - `actor::aquamentus::Aquamentus`, 8
 - `actor::boomerang::Boomerang`, 10
 - `actor::boss::Boss`, 12
 - `actor::enemy::Enemy`, 15
 - `actor::fireball::Fireball`, 18
 - `actor::healthbar::Health_Bar`, 21
 - `actor::item::Item`, 23
 - `actor::keese::Keese`, 25
 - `actor::player::Player`, 31
 - `actor::rupee::Rupee_Bar`, 33
 - `actor::spritesheet::SpriteSheet`, 35
 - `actor::stalfos::Stalfos`, 38
 - `actor::sword::Sword`, 43
 - `collision::door::Door`, 14
 - `collision::wall::Wall`, 44
- `actor.aquamentus.Aquamentus`, 7
- `actor.boomerang.Boomerang`, 9
- `actor.boss.Boss`, 11
- `actor.enemy.Enemy`, 14
- `actor.fireball.Fireball`, 17
- `actor.healthbar.Health_Bar`, 20
- `actor.item.Item`, 22
- `actor.keese.Keese`, 24
- `actor.player.Player`, 29
- `actor.rupee.Rupee_Bar`, 32
- `actor.spritesheet.SpriteSheet`, 34
- `actor.stalfos.Stalfos`, 36
- `actor.sword.Sword`, 42
- `actor::aquamentus::Aquamentus`
 - `__init__`, 8
 - `attack`, 8
 - `bossLogic`, 9
 - `checkState`, 9
 - `swapDirection`, 9
- `actor::boomerang::Boomerang`
 - `__init__`, 10
- `actor::boss::Boss`
 - `__init__`, 12
 - `hit`, 12
 - `move`, 13
 - `update`, 13
- `actor::enemy::Enemy`
 - `__init__`, 15
 - `hit`, 16
 - `hitCount`, 16
 - `hitdir`, 17
 - `move`, 16
 - `update`, 16
- `actor::fireball::Fireball`
 - `__init__`, 18
 - `checkState`, 18
 - `end`, 19
 - `hit`, 19
 - `logic`, 19
 - `move`, 19
 - `start`, 19
 - `update`, 20
- `actor::healthbar::Health_Bar`
 - `__init__`, 21
 - `h_sprite_sheet`, 22
 - `health`, 21
 - `image`, 22
 - `rect`, 22
- `actor::item::Item`
 - `__init__`, 23
 - `collision`, 24
- `actor::keese::Keese`
 - `__init__`, 25
 - `checkState`, 25
 - `enemyLogic`, 26
 - `genRestLength`, 26
 - `genTravelPoint`, 26
 - `setMoveSpeed`, 26
 - `stop`, 26
 - `switchSprite`, 27
- `actor::player::Player`
 - `__init__`, 31
 - `attacksword`, 32
 - `attackupdate`, 31
 - `dir`, 32
 - `move`, 31
 - `moveupdate`, 32
- `actor::rupee::Rupee_Bar`
 - `__init__`, 33
 - `image`, 33
 - `rect`, 34
- `actor::spritesheet::SpriteSheet`
 - `__init__`, 35
 - `get_image`, 35
 - `get_imageNT`, 35
 - `sprite_sheet`, 36
- `actor::stalfos::Stalfos`
 - `__init__`, 38
 - `checkState`, 38
 - `direction`, 39
 - `dmg`, 39
 - `enemyLogic`, 38

- genTravelPath, 38
- HP, 39
- image, 40
- isHit, 40
- isMoving, 40
- maxHP, 40
- obj, 40
- oldx, 40
- oldy, 41
- previousDirection, 41
- setWalkSpeed, 39
- spriteIndex, 41
- sprites, 41
- stop, 39
- walkFrames, 41
- walkStartFrame, 41
- xSpeed, 42
- ySpeed, 42
- actor::sword::Sword
 - __init__, 43
 - image, 43
- aquamentus.py
 - populateSprites, 48
- attack
 - actor::aquamentus::Aquamentus, 8
- attacksword
 - actor::player::Player, 32
- attackupdate
 - actor::player::Player, 31
- bossLogic
 - actor::aquamentus::Aquamentus, 9
- checkState
 - actor::aquamentus::Aquamentus, 9
 - actor::fireball::Fireball, 18
 - actor::keese::Keese, 25
 - actor::stalfos::Stalfos, 38
- collision
 - actor::item::Item, 24
 - collision::wall::Wall, 45
- collision.door.Door, 13
- collision.level.Level, 27
- collision.levelmanager.LevelManager, 28
- collision.wall.Wall, 44
- collision::door::Door
 - __init__, 14
- collision::levelmanager::LevelManager
 - transition, 28
- collision::wall::Wall
 - __init__, 44
 - collision, 45
 - id, 45
 - image, 45
 - rect, 45
- dir
 - actor::player::Player, 32
- direction
 - actor::stalfos::Stalfos, 39
- dmg
 - actor::stalfos::Stalfos, 39
- end
 - actor::fireball::Fireball, 19
- enemyLogic
 - actor::keese::Keese, 26
 - actor::stalfos::Stalfos, 38
- fireball.py
 - populateSprites, 52
- genRestLength
 - actor::keese::Keese, 26
- genTravelPath
 - actor::stalfos::Stalfos, 38
- genTravelPoint
 - actor::keese::Keese, 26
- get_image
 - actor::spritesheet::SpriteSheet, 35
- get_imageNT
 - actor::spritesheet::SpriteSheet, 35
- h_sprite_sheet
 - actor::healthbar::Health_Bar, 22
- health
 - actor::healthbar::Health_Bar, 21
- hit
 - actor::boss::Boss, 12
 - actor::enemy::Enemy, 16
 - actor::fireball::Fireball, 19
- hitCount
 - actor::enemy::Enemy, 16
- hitdir
 - actor::enemy::Enemy, 17
- HP
 - actor::stalfos::Stalfos, 39
- id
 - collision::wall::Wall, 45
- image
 - actor::healthbar::Health_Bar, 22
 - actor::rupee::Rupee_Bar, 33
 - actor::stalfos::Stalfos, 40
 - actor::sword::Sword, 43
 - collision::wall::Wall, 45
- isHit
 - actor::stalfos::Stalfos, 40
- isMoving
 - actor::stalfos::Stalfos, 40
- keese.py
 - populateSprites, 54
- logic
 - actor::fireball::Fireball, 19
- maxHP
 - actor::stalfos::Stalfos, 40

- move
 - actor::boss::Boss, 13
 - actor::enemy::Enemy, 16
 - actor::fireball::Fireball, 19
 - actor::player::Player, 31
- moveupdate
 - actor::player::Player, 32
- obj
 - actor::stalfos::Stalfos, 40
- oldx
 - actor::stalfos::Stalfos, 40
- oldy
 - actor::stalfos::Stalfos, 41
- populateSprites
 - aquamentus.py, 48
 - fireball.py, 52
 - keese.py, 54
 - stalfos.py, 56
- previousDirection
 - actor::stalfos::Stalfos, 41
- rect
 - actor::healthbar::Health_Bar, 22
 - actor::rupee::Rupee_Bar, 34
 - collision::wall::Wall, 45
- setMoveSpeed
 - actor::keese::Keese, 26
- setWalkSpeed
 - actor::stalfos::Stalfos, 39
- sprite_sheet
 - actor::spritesheet::SpriteSheet, 36
- spriteIndex
 - actor::stalfos::Stalfos, 41
- sprites
 - actor::stalfos::Stalfos, 41
- src/actor/aquamentus.py, 47
- src/actor/boomerang.py, 48
- src/actor/boss.py, 48
- src/actor/constants.py, 49
- src/actor/enemy.py, 51
- src/actor/fireball.py, 51
- src/actor/healthbar.py, 52
- src/actor/item.py, 53
- src/actor/keese.py, 53
- src/actor/player.py, 54
- src/actor/rupee.py, 55
- src/actor/spritesheet.py, 55
- src/actor/stalfos.py, 55
- src/actor/sword.py, 56
- src/collision/door.py, 57
- src/collision/level.py, 57
- src/collision/levelmanager.py, 58
- src/collision/wall.py, 58
- src/config/colour.py, 59
- src/config/window.py, 59
- stalfos.py
 - populateSprites, 56
- start
 - actor::fireball::Fireball, 19
- stop
 - actor::keese::Keese, 26
 - actor::stalfos::Stalfos, 39
- swapDirection
 - actor::aquamentus::Aquamentus, 9
- switchSprite
 - actor::keese::Keese, 27
- transition
 - collision::levelmanager::LevelManager, 28
- update
 - actor::boss::Boss, 13
 - actor::enemy::Enemy, 16
 - actor::fireball::Fireball, 20
- walkFrames
 - actor::stalfos::Stalfos, 41
- walkStartFrame
 - actor::stalfos::Stalfos, 41
- xSpeed
 - actor::stalfos::Stalfos, 42
- ySpeed
 - actor::stalfos::Stalfos, 42