My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

MacLunky Source Code

The folders and files for this project are as follows:

...

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

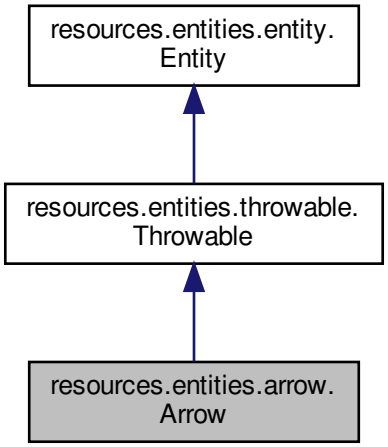
Chapter 5

Class Documentation

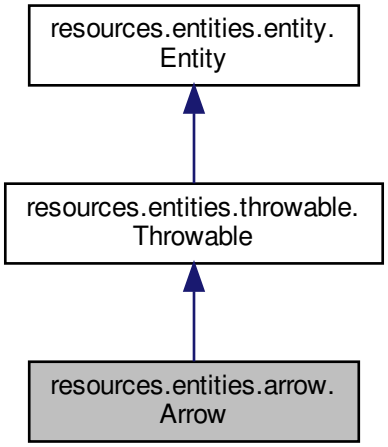
5.1 resources.entities.arrow.Arrow Class Reference

Arrow is an object of type Throwable.

Inheritance diagram for resources.entities.arrow.Arrow:



Collaboration diagram for resources.entities.arrow.Arrow:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Arrow.
- def pickup (self)

  Set held and playerOwned to True.
- def putDown (self)

  Set held and playerOwned to False.
- def tick (self, gameinfo=None, player=None)

  animation movement of the arrow object that is in attack mode

Public Attributes

- **ARROWDAMAGE**
- **playerOwned**
- **xs**

5.1.1 Detailed Description

Arrow is an object of type Throwable.

Extends from throwable.py and implements the secrets of M4.15

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
def resources.entities.arrow.Arrow.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Arrow.

Parameters

x	an integer element indicating the x position of the arrow object
y	an integer element indicating the y position of the arrow object
hp	an integer element indicating the health points of the arrow object
mat	an image representing the arrow object on the game screen
name	a string representing the name of the image of the object
width	an integer element indicating the width of the arrow object
height	an integer element indicating the height of the arrow object

5.1.3 Member Function Documentation

5.1.3.1 tick()

```
def resources.entities.arrow.Arrow.tick (
              self,
              gameinfo = None,
              player = None )
```

animation movement of the arrow object that is in attack mode

Parameters

gameinfo	a ReadMap object element indicating the map where the arrow object is on
player	a Mover object element representing the player of the game that is within range of arrow attack

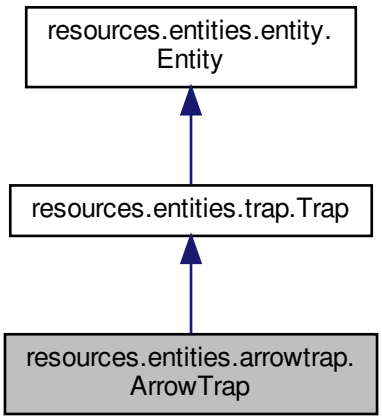
The documentation for this class was generated from the following file:

- resources/entities/arrow.py

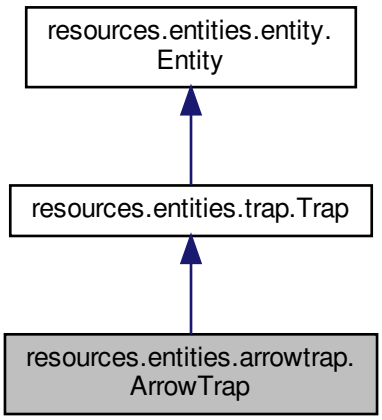
5.2 resources.entities.arrowtrap.ArrowTrap Class Reference

ArrowTrap is an object of type Trap.

Inheritance diagram for resources.entities.arrowtrap.ArrowTrap:



Collaboration diagram for resources.entities.arrowtrap.ArrowTrap:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for ArrowTrap.
- def **setDirection** (self, d)
- def sense (self, gameinfo=None, player=None)

  check if the player is within the sensing range of the arrow trap object
- def tick (self, gameinfo=None, player=None)

  animation movement of the arrow trap object attack mechanism to shoot an arrow

Public Attributes

- **arrowSense**
- **direction**
- **arrowMade**

5.2.1 Detailed Description

ArrowTrap is an object of type Trap.

Extends from trap.py and implements the secrets of M4.14

5.2.2 Constructor & Destructor Documentation

5.2.2.1 __init__()

```
def resources.entities.arrowtrap.ArrowTrap.__init__ (
                self,
                x = None,
                y = None,
                name = None,
                height = None,
                width = None,
                hp = None,
                mat = None )
```

Constructor method for ArrowTrap.

Parameters

x	an integer element indicating the x position of the arrow trap object
y	an integer element indicating the y position of the arrow trap object
hp	an integer element indicating the health points of the arrow trap object
mat	an image representing the arrow trap object on the game screen
name	a string representing the name of the image of the object
width	an integer element indicating the width of the arrow trap object
height	an integer element indicating the height of the arrow trap object

5.2.3 Member Function Documentation

5.2.3.1 sense()

```
def resources.entities.arrowtrap.ArrowTrap.sense (
                self,
```

```
                gameinfo = None,
                player = None )
```

check if the player is within the sensing range of the arrow trap object

Parameters

gameinfo	a ReadMap object element indicating the map where the arrow trap object is on
player	a Mover object element representing the player of the game that is within range

Returns

a Boolean value indicating if the player was sensed or not, True is returned if the player is within range of the trap

5.2.3.2 tick()

```
def resources.entities.arrowtrap.ArrowTrap.tick (
                self,
                gameinfo = None,
                player = None )
```

animation movement of the arrow trap object attack mechanism to shoot an arrow

Parameters

gameinfo	a ReadMap object element indicating the map where the arrow trap object is on
player	a Mover object element representing the player of the game that is within range of arrow trap attack

Returns

a Boolean value indicating that animation for the frame update is done

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

- resources/entities/arrowtrap.py

5.3 resources.blocks.bits.bits.Bitloader Class Reference

Public Member Functions

- def **__init__** (self)
- def **setoverlay** (self, tile=None, mapinfo=None, bitdict=None, bittype=None)
- def **__init__** (self)
- def **setoverlay** (self, tile=None, mapinfo=None, bitdict=None, bittype=None)

Public Attributes

- **images**

5.3.1 Detailed Description

```
Loading 'bits' (the extra polished parts) to map tiles
```

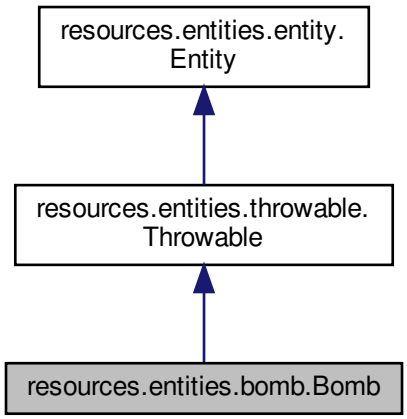
The documentation for this class was generated from the following file:

- Pylunky/pylunky-master/resources/blocks/bits/bits.py

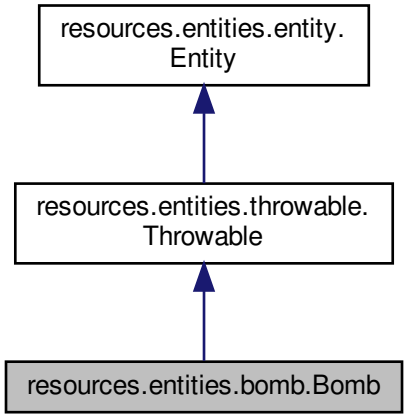
5.4 resources.entities.bomb.Bomb Class Reference

Bomb is an object of type Throwable.

Inheritance diagram for resources.entities.bomb.Bomb:



Collaboration diagram for resources.entities.bomb.Bomb:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Bomb.
- def tick (self, gameinfo=None, player=None)

  Update the bomb with 1 unit of time.
- def explode (self, gameinfo=None, player=None)

  Explode the bomb.

Public Attributes

- **BOMBTIME**
- **time**
- **xs**

5.4.1 Detailed Description

Bomb is an object of type Throwable.

Extends from throwable.py and implements the secrets of M4.16

5.4.2 Constructor & Destructor Documentation

5.4.2.1 __init__()

```
def resources.entities.bomb.Bomb.__init__ (
                self,
                x = None,
                y = None,
                name = None,
                height = None,
                width = None,
                hp = None,
                mat = None )
```

Constructor method for Bomb.

Create the bomb using super(). Set the time to BOMBTIME.

Parameters

x	int representing the x position of the bomb
y	int representing the y position of the bomb
hp	int representing the health points of the bomb
mat	image representing the bomb on the game screen
name	string representing the name of the image of the object
width	int representing the width of the bomb
height	int representing the height of the bomb

5.4.3 Member Function Documentation

5.4.3.1 explode()

```
def resources.entities.bomb.Bomb.explode (
                self,
                gameinfo = None,
                player = None )
```

Explode the bomb.

Create an explosion concentric with the bomb. Add it to the game and remove the bomb.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.4.3.2 tick()

```
def resources.entities.bomb.Bomb.tick (
              self,
              gameinfo = None,
              player = None )
```

Update the bomb with 1 unit of time.

Reduce the time. Call super.tick(). Explode if time reaches 0, the bomb stops on a wall, or is destroyed.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

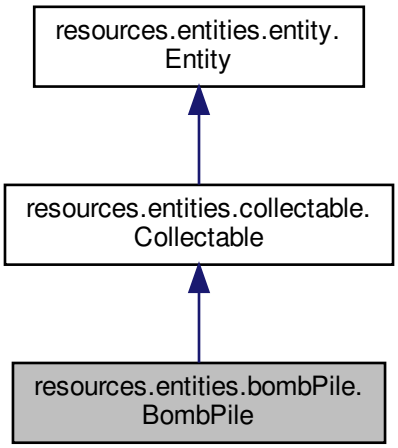
The documentation for this class was generated from the following file:

- resources/entities/bomb.py

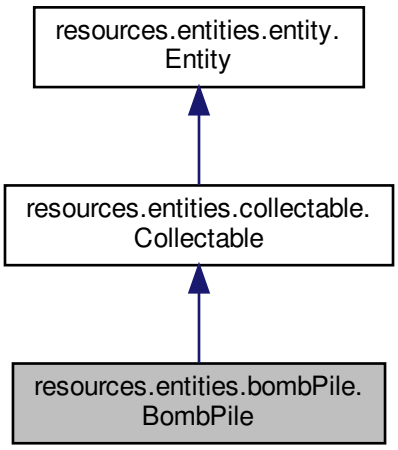
5.5 resources.entities.bombPile.BombPile Class Reference

BombPile is an object of type Collectable.

Inheritance diagram for resources.entities.bombPile.BombPile:



Collaboration diagram for resources.entities.bombPile.BombPile:



Public Member Functions

- def use (self, gameinfo=None, player=None)

  Use the bombPile.

Additional Inherited Members

5.5.1 Detailed Description

BombPile is an object of type Collectable.

Extends from collectable.py and implements the secrets of M4.9

5.5.2 Member Function Documentation

5.5.2.1 use()

```
def resources.entities.bombPile.BombPile.use (
              self,
              gameinfo = None,
              player = None )
```

Use the bombPile.

Give the player bombs equal to the bombPile's value.

Parameters

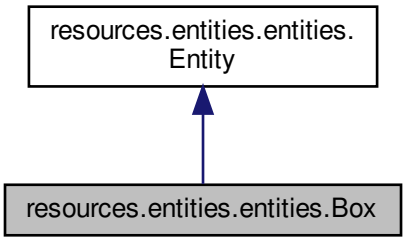
gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

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

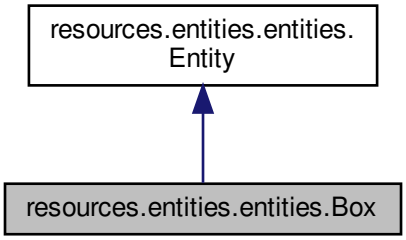
- resources/entities/bombPile.py

5.6 resources.entities.entities.Box Class Reference

Inheritance diagram for resources.entities.entities.Box:



Collaboration diagram for resources.entities.entities.Box:



Public Member Functions

- def **use** (self, gameinfo=None, player=None)

Additional Inherited Members

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

- Pylunky/pylunky-master/resources/entities/entities.py

5.7 camera.cam Class Reference

Update the camera showing the game based on player movement.

Public Member Functions

- def __init__ (self, x=0, y=0, width=0, height=0, mapw=0, maph=0)
  Constructor to create the camera type.
- def move (self, x, y, adj)
  Move the camera.
- def **__init__** (self, x=0, y=0, width=0, height=0, mapw=0, maph=0)
- def move (self, x, y)

Public Attributes

- **x**
- **y**
- **width**
- **height**
- **mapw**
- **maph**
- **adj**

5.7.1 Detailed Description

Update the camera showing the game based on player movement.

Implements the secrets of M1. Show only the portion of the map applicable to the player's motion by the user

5.7.2 Constructor & Destructor Documentation

5.7.2.1 __init__()

```
def camera.cam.__init__ (
              self,
              x = 0,
              y = 0,
              width = 0,
              height = 0,
              mapw = 0,
              maph = 0 )
```

Constructor to create the camera type.

Parameters

x	an integer element indicating the x position of the camera object
y	an integer element indicating the y position of the camera object
height	an integer element indicating the height of the camera object
mapw	an integer element indicating the width of the camera map
maph	an integer element indicating the height of the camera map

5.7.3 Member Function Documentation

5.7.3.1 move() [1/2]

```
def camera.cam.move (
              self,
              x,
              y )
```

```
Move the camera to where the player is, lock to the corners of the map
```

5.7.3.2 move() [2/2]

```
def camera.cam.move (
              self,
              x,
              y,
              adj )
```

Move the camera.

Move the camera to where the player is, lock to the corners of the map

Parameters

x	x axis coordinate position of the player wrt to the camera coordinates
y	y axis coordinate position of the player wrt to the camera coordinates
adj	int representing how much to move the camera

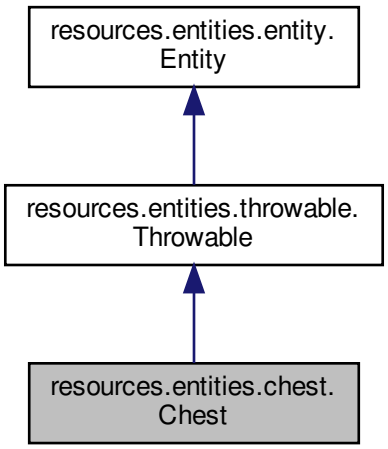
The documentation for this class was generated from the following file:

- camera.py

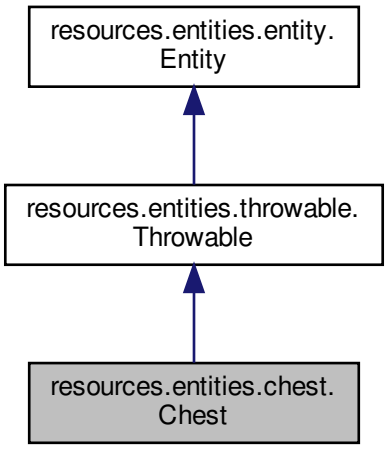
5.8 resources.entities.chest.Chest Class Reference

Chest is an object of type Throwable.

Inheritance diagram for resources.entities.chest.Chest:



Collaboration diagram for resources.entities.chest.Chest:



Public Member Functions

- def useable (self)

Constructor method for Chest.

- def use (self, gameinfo=None, player=None)

  Use the throwable.

Additional Inherited Members

5.8.1 Detailed Description

Chest is an object of type Throwable.

Extends from throwable.py and implements the secrets of M4.15

5.8.2 Member Function Documentation

5.8.2.1 use()

```
def resources.entities.chest.Chest.use (
              self,
              gameinfo = None,
              player = None )
```

Use the throwable.

Call super(). If not open create a random Treasure, place it centred with the chest, and add it to the game.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.8.2.2 useable()

```
def resources.entities.chest.Chest.useable (
              self )
```

Constructor method for Chest.

Create the chest using super(). Set open to False.

Parameters

x	int representing the x position of the chest
y	int representing the y position of the chest

Parameters

hp	int representing the health points of the chest
mat	image representing the chest on the game screen
name	string representing the name of the image of the object
text	string representing addditional object information
width	int representing the width of the chest
height	int representing the height of the chest def **init**(self, x=None, y=None, hp=None, mat=None, name=None, text=None, width=None, height=None): super().__init__(x, y, hp, mat, name, text, width, height) self.open = False Determine if a chest is useable

Returns

True

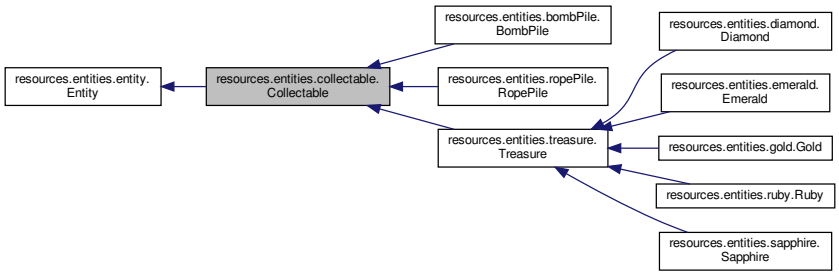
The documentation for this class was generated from the following file:

- resources/entities/chest.py

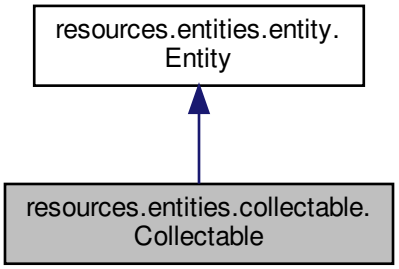
5.9 resources.entities.collectable.Collectable Class Reference

Collectable is an object of type Entity.

Inheritance diagram for resources.entities.collectable.Collectable:



Collaboration diagram for resources.entities.collectable.Collectable:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Collectable.
- def useable (self)

  Determine if a collectable is useable.
- def use (self, gameinfo=None, player=None)

  Use the collectable.

Public Attributes

- **val**

5.9.1 Detailed Description

Collectable is an object of type Entity.

Extends from entity.py and implements the secrets of M4.2

5.9.2 Constructor & Destructor Documentation

5.9.2.1 __init__()

```
def resources.entities.collectable.Collectable.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Collectable.

Create the collectable using super(). Set the value.

Parameters

x	int representing the x position of the collectable
y	int representing the y position of the collectable
hp	int representing the health points of the collectable
mat	image representing the collectable on the game screen
name	string representing the name of the image of the object
width	int representing the width of the collectable
height	int representing the height of the collectable

5.9.3 Member Function Documentation

5.9.3.1 use()

```
def resources.entities.collectable.Collectable.use (
              self,
              gameinfo = None,
              player = None )
```

Use the collectable.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.9.3.2 useable()

```
def resources.entities.collectable.Collectable.useable (
              self )
```

Determine if a collectable is useable.

Returns

True

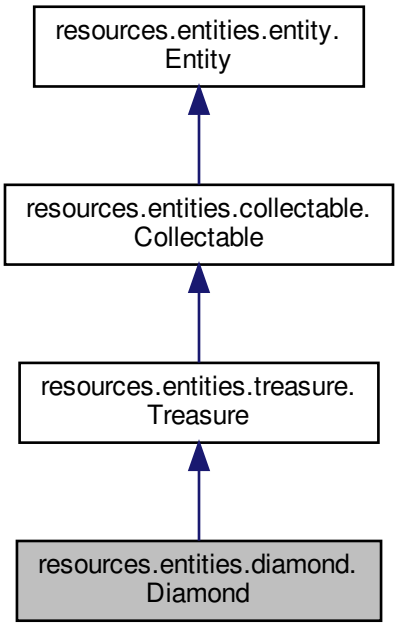
The documentation for this class was generated from the following file:

- resources/entities/collectable.py

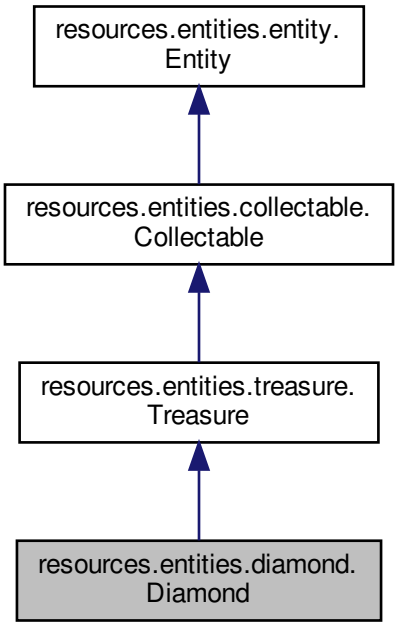
5.10 resources.entities.diamond.Diamond Class Reference

Diamond is an object of type Treasure.

Inheritance diagram for resources.entities.diamond.Diamond:



Collaboration diagram for resources.entities.diamond.Diamond:



Additional Inherited Members

5.10.1 Detailed Description

Diamond is an object of type Treasure.

Extends from treasure.py and implements the secrets of M4.8

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

- resources/entities/diamond.py

5.11 display.display Class Reference

Display is a class that implements the game user interface screen placements and changes and implements the secrets of M1.

Public Member Functions

- def __init__ (self)

  constructor method for class display, intializes text font for game screen information to be displayed
- def ui (self, screen=None, camera=None, player=None, gameinfo=None)

  method for updating the game display on the screen
- def showent (self, screen=None, cam=None, gameinfo=None)

  method for updating the game display for showing entities objects on the scren
- def sign (self, screen=None, cam=None, gameinfo=None, player=None)

  method for updating the game display for showing the sign text information as a player passes the entity on the screen
- def resourceDisp (self, screen=None, player=None)

  method for updating the game display for showing the player state information on the screen
- def goldDisp (self, screen=None, player=None)

  method for updating the game display for showing the gold to be collected on the screen
- def gameover (self, screen=None, player=None)

  method for updating the game screen when the win condition is met or the lose condition is met or exit condition met
- def addScore (self, screen=None, player=None)

  method for saving the player is score in a highscore file that can be accessed by the player outside the game
- def **location** (self, screen=None, player=None)
- def **__init__** (self)
- def **ui** (self, screen=None, camera=None, player=None, gameinfo=None)
- def **showent** (self, screen=None, cam=None, gameinfo=None)
- def **sign** (self, screen=None, cam=None, gameinfo=None, player=None)
- def **hpdisp** (self, screen=None, player=None)
- def **golddisp** (self, screen=None, player=None)
- def **gameover** (self, screen=None, player=None)

Public Attributes

- **textfont**
- **c**

5.11.1 Detailed Description

Display is a class that implements the game user interface screen placements and changes and implements the secrets of M1.

Implements the secrets of M1. The class contains screen changes information and the displayed content of the game

5.11.2 Member Function Documentation

5.11.2.1 addScore()

```
def display.display.addScore (
              self,
              screen = None,
              player = None )
```

method for saving the player is score in a highscore file that can be accessed by the player outside the game

Parameters

screen	is a pygame display surface representing the game screen
player	is a Mover element indicating the player playing the game and interacting with the screen

5.11.2.2 gameover()

```
def display.display.gameover (
              self,
              screen = None,
              player = None )
```

method for updating the game screen when the win condition is met or the lose condition is met or exit condition met

Parameters

screen	is a pygame display surface representing the game screen
player	is a Mover element indicating the player playing the game and interacting with the screen

5.11.2.3 goldDisp()

```
def display.display.goldDisp (
              self,
              screen = None,
              player = None )
```

method for updating the game display for showing the gold to be collected on the screen

Parameters

screen	is a pygame display surface representing the game screen
player	is a Mover element indicating the player playing the game and interacting with the screen

5.11.2.4 resourceDisp()

```
def display.display.resourceDisp (
              self,
              screen = None,
              player = None )
```

method for updating the game display for showing the player state information on the screen

updates the information displayed about the player health and resources to be used to aid the player in the game

Parameters

screen	is a pygame display surface representing the game screen
player	is a Mover element indicating the player playing the game and interacting with the screen

5.11.2.5 showent()

```
def display.display.showent (
              self,
              screen = None,
              cam = None,
              gameinfo = None )
```

method for updating the game display for showing entities objects on the scren

Parameters

screen	is a pygame display surface representing the game screen
cam	is a camera object indicating the player screen view
gameinfo	a ReadMap object element indicating the map where the Mover object is on

5.11.2.6 sign()

```
def display.display.sign (
              self,
              screen = None,
              cam = None,
              gameinfo = None,
              player = None )
```

method for updating the game display for showing the sign text information as a player passes the entity on the screen

Parameters

screen	is a pygame display surface representing the game screen
cam	is a camera object indicating the player screen view
player	is a Mover element indicating the player playing the game and interacting with the screen
gameinfo	a ReadMap object element indicating the map where the Mover object is on

5.11.2.7 ui()

```
def display.display.ui (
```

```
self,
screen = None,
camera = None,
player = None,
gameinfo = None )
```

method for updating the game display on the screen

Parameters

screen	is a pygame display surface representing the game screen
camera	is a camera object indicating the player screen view
player	is a Mover element indicating the player playing the game and interacting with the screen
gameinfo	a ReadMap object element indicating the map where the Mover object is on

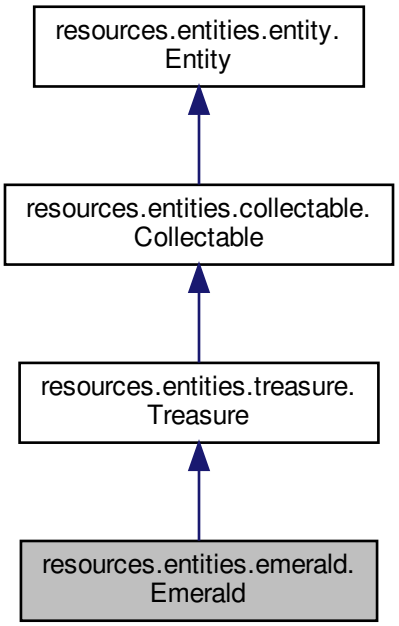
The documentation for this class was generated from the following file:

- display.py

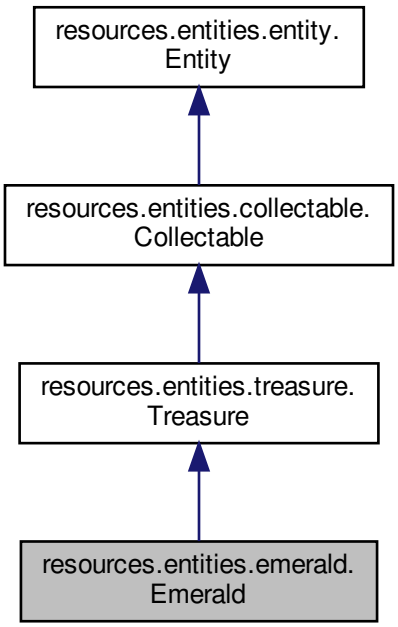
5.12 resources.entities.emerald.Emerald Class Reference

Emerald is an object of type Treasure.

Inheritance diagram for resources.entities.emerald.Emerald:



Collaboration diagram for resources.entities.emerald.Emerald:



Additional Inherited Members

5.12.1 Detailed Description

Emerald is an object of type Treasure.

Extends from treasure.py and implements the secrets of M4.6

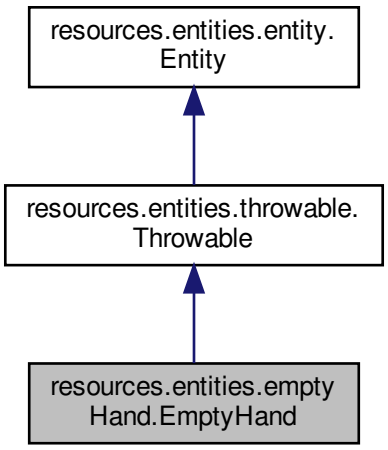
The documentation for this class was generated from the following file:

- resources/entities/emerald.py

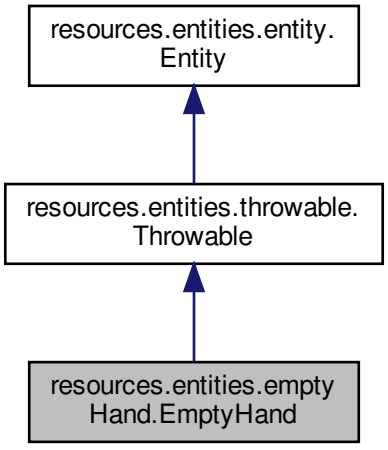
5.13 resources.entities.emptyHand.EmptyHand Class Reference

EmptyHand is an object of type Throwable.

Inheritance diagram for resources.entities.emptyHand.EmptyHand:



Collaboration diagram for resources.entities.emptyHand.EmptyHand:



Additional Inherited Members

5.13.1 Detailed Description

EmptyHand is an object of type Throwable.

Extends from throwable.py and implements the secrets of M4.17

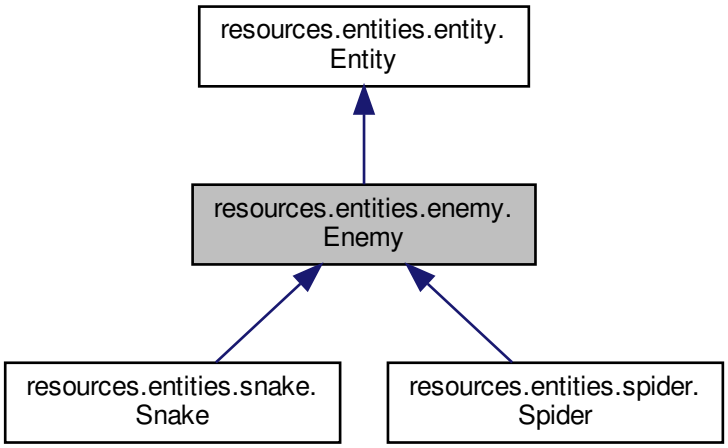
The documentation for this class was generated from the following file:

- resources/entities/emptyHand.py

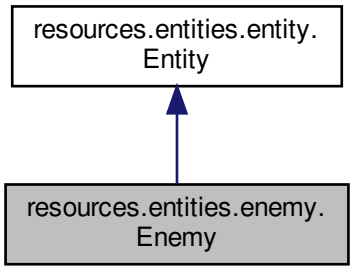
5.14 resources.entities.enemy.Enemy Class Reference

Enemy is an object of type Entity.

Inheritance diagram for resources.entities.enemy.Enemy:



Collaboration diagram for resources.entities.enemy.Enemy:



Public Member Functions

- def **__init__** (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)
- def tick (self, gameinfo=None, player=None)

Update the enemy with 1 unit of time.

- def damage (self, d=0, gameinfo=None)

  Take damage.

- def gravity (self)

  Gravity effect.

- def move (self, gameinfo=None, x=None, y=None)

  Move.

Public Attributes

- **xs**
- **ys**
- **GRAVITY**
- **attack**

5.14.1 Detailed Description

Enemy is an object of type Entity.

Extends from entity.py and implements the secrets of M4.18

5.14.2 Member Function Documentation

5.14.2.1 damage()

```
def resources.entities.enemy.Enemy.damage (
              self,
              d = 0,
              gameinfo = None )
```

Take damage.

Take damage. Remove from game if dead.

Parameters

d	int representing the damage taken
gameinfo	ReadMap object representing the level environment

5.14.2.2 move()

```
def resources.entities.enemy.Enemy.move (
              self,
```

```
                gameinfo = None,
                x = None,
                y = None )
```

Move.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.14.2.3 tick()

```
def resources.entities.enemy.Enemy.tick (
                self,
                gameinfo = None,
                player = None )
```

Update the enemy with 1 unit of time.

Gravity effect, move, damage player.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

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

- resources/entities/enemy.py

5.15 resources.entities.entities.Entities Class Reference

Entities is a class that makes entities.

Public Member Functions

- def makeEnt (name, x, y)

  Create an entity.

5.15.1 Detailed Description

Entities is a class that makes entities.

Implements the secrets of M4

5.15.2 Member Function Documentation

5.15.2.1 makeEnt()

```
def resources.entities.entities.Entities.makeEnt (
              name,
              x,
              y )
```

Create an entity.

Parameters

x	an integer element indicating the x position of the entity object
y	an integer element indicating the y position of the entity object
name	a string representing the name of the entity

Returns

Entity object representing the created entity

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

- resources/entities/entities.py

5.16 readmap.Entity Class Reference

Public Member Functions

- def **__init__** (self, x=None, y=None, hp=None, mat=None)
- def **move** (self, x=None, y=None)

Public Attributes

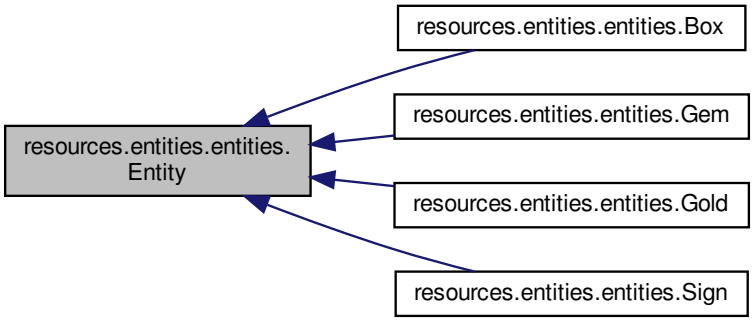
- **x**
- **y**
- **hp**
- **mat**

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

- Pylunky/pylunky-master/readmap.py

5.17 resources.entities.entities.Entity Class Reference

Inheritance diagram for resources.entities.entities.Entity:



Public Member Functions

- def **__init__** (self, x=None, y=None, hp=None, mat=None, name=None, text=None, width=None, height=None)
- def **move** (self, x=None, y=None)
- def **position** (self)
- def **debuginfo** (self)

Public Attributes

- **x**
- **y**
- **height**
- **width**
- **hp**
- **mat**
- **name**
- **target**
- **text**

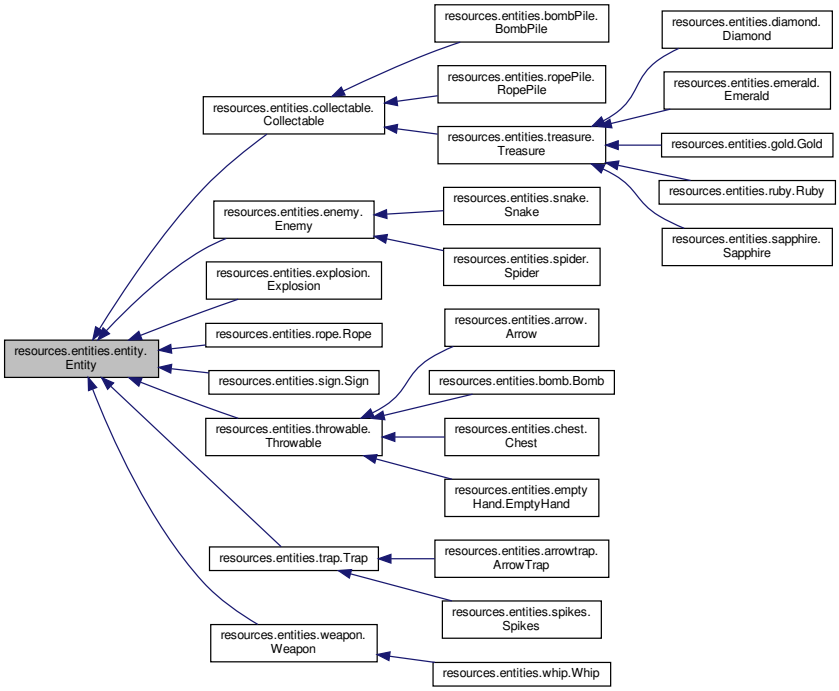
The documentation for this class was generated from the following file:

- Pylunky/pylunky-master/resources/entities/entities.py

5.18 resources.entities.entity.Entity Class Reference

Entity is an object.

Inheritance diagram for resources.entities.entity.Entity:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Entity.
- def move (self, x=None, y=None)

  Change the position.
- def set (self, x=None, y=None)

  Set the position.
- def position (self)

  Get the position.
- def dimension (self)

  Get the dimensions.
- def damage (self, d=0, gameinfo=None)

  Take damage.
- def overlap (self, ent=None)

  Determine if an entity overlaps with self.
- def tick (self, gameinfo=None, player=None)

  Update the Entity with 1 unit of time.
- def useable (self)

  Determine if an entity is useable.
- def use (self, gameinfo=None, player=None)

  Use the entity.
- def flipImage (self, x=None, y=None)

  Flip mat.

Public Attributes

- **x**
- **y**
- **height**
- **width**
- **hp**
- **mat**
- **name**

5.18.1 Detailed Description

Entity is an object.

Implements the secrets of M4.1

5.18.2 Constructor & Destructor Documentation

5.18.2.1 __init__()

```
def resources.entities.entity.Entity.__init__ (
                self,
                x = None,
                y = None,
                name = None,
                height = None,
                width = None,
                hp = None,
                mat = None )
```

Constructor method for Entity.

Set the state.

Parameters

x	int representing the x position of the entity
y	int representing the y position of the entity
hp	int representing the health points of the entity
mat	image representing the entity on the game screen
name	string representing the name of the image of the object
width	int representing the width of the entity
height	int representing the height of the entity

5.18.3 Member Function Documentation

5.18.3.1 damage()

```
def resources.entities.entity.Entity.damage (
              self,
              d = 0,
              gameinfo = None )
```

Take damage.

Take damage

Parameters

d	int representing the damage taken
gameinfo	ReadMap object representing the level environment

5.18.3.2 dimension()

```
def resources.entities.entity.Entity.dimension (
              self )
```

Get the dimensions.

Returns

int, int representing the height and width

5.18.3.3 flipImage()

```
def resources.entities.entity.Entity.flipImage (
              self,
              x = None,
              y = None )
```

Flip mat.

Parameters

x	boolean representing vertical mirroring
y	boolean representing horizontal mirroring

5.18.3.4 move()

```
def resources.entities.entity.Entity.move (
              self,
              x = None,
              y = None )
```

Change the position.

Parameters

x	int representing the change in the x axis
y	int representing the change in the y axis

5.18.3.5 overlap()

```
def resources.entities.entity.Entity.overlap (
              self,
              ent = None )
```

Determine if an entity overlaps with self.

Check corners

Parameters

ent	Entity object representing another Entity

5.18.3.6 position()

```
def resources.entities.entity.Entity.position (
              self )
```

Get the position.

Returns

int, int representing the position in the x and y axis

5.18.3.7 set()

```
def resources.entities.entity.Entity.set (
              self,
              x = None,
              y = None )
```

Set the position.

Parameters

x	int representing the position in the x axis
y	int representing the position in the y axis

5.18.3.8 tick()

```
def resources.entities.entity.Entity.tick (
              self,
              gameinfo = None,
              player = None )
```

Update the Entity with 1 unit of time.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.18.3.9 use()

```
def resources.entities.entity.Entity.use (
              self,
              gameinfo = None,
              player = None )
```

Use the entity.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.18.3.10 useable()

```
def resources.entities.entity.Entity.useable (
              self )
```

Determine if an entity is useable.

Returns

False

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

- resources/entities/entity.py

5.19 readmap.EntMap Class Reference

Map of all entities.

Public Member Functions

- def **__init__** (self, text=None)
- def **count** (self)
- def **add** (self, entry=None)
- def **rem** (self, entry=None)
- def __init__ (self, text=None)
  Create the entities.
- def count (self)
  Get number of entities.
- def add (self, entry=None)
  Add an entity.
- def rem (self, entry=None)
  Remove an entity.

Public Attributes

- **entlist**
- **enemylist**

5.19.1 Detailed Description

Map of all entities.

Implements the secrets of M5

5.19.2 Constructor & Destructor Documentation

5.19.2.1 __init__()

```
def readmap.EntMap.__init__ (
              self,
              text = None )
```

Create the entities.

Create the entities from the map data

Parameters

text	File containing map data

5.19.3 Member Function Documentation

5.19.3.1 add()

```
def readmap.EntMap.add (
              self,
              entry = None )
```

Add an entity.

Parameters

entry	Entity object representing the entity to add

5.19.3.2 count()

```
def readmap.EntMap.count (
              self )
```

Get number of entities.

Returns

int representing the number of entities

5.19.3.3 rem()

```
def readmap.EntMap.rem (
              self,
              entry = None )
```

Remove an entity.

Parameters

entry	Entity object representing the entity to remove

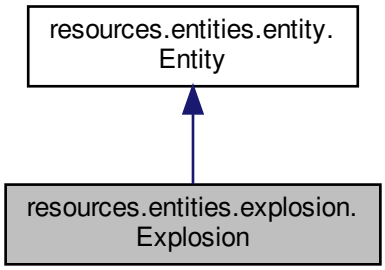
The documentation for this class was generated from the following file:

- Pylunky/pylunky-master/readmap.py

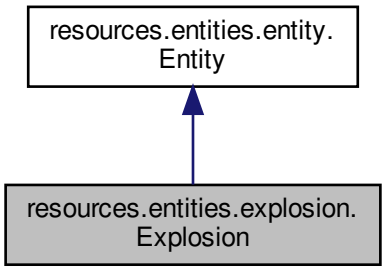
5.20 resources.entities.explosion.Explosion Class Reference

Explosion is an object of type Entity.

Inheritance diagram for resources.entities.explosion.Explosion:



Collaboration diagram for resources.entities.explosion.Explosion:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Explosion.
- def tick (self, gameinfo=None, player=None)

  Update the explosion with 1 unit of time.

- def explode (self, gameinfo=None, player=None)

  Destroy blocks, player, and entities.
- def destroyBlock (self, gameinfo=None)

  Destroy nearby blocks.
- def destroyEnt (self, gameinfo=None)

  Damage nearby entities.
- def destroyPlayer (self, player=None)

  Damage the nearby player.

Public Attributes

- **DURATION**
- **time**
- **DAMAGE**

5.20.1 Detailed Description

Explosion is an object of type Entity.

Extends from entity.py and implements the secrets of M4.12

5.20.2 Constructor & Destructor Documentation

5.20.2.1 __init__()

```
def resources.entities.explosion.Explosion.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Explosion.

Create the explosion using super(). Set the time to DURATION.

Parameters

x	int representing the x position of the explosion
y	int representing the y position of the explosion
hp	int representing the health points of the explosion
mat	image representing the explosion on the game screen
name	string representing the name of the image of the object
width	int representing the width of the explosion
height	int representing the height of the explosion

5.20.3 Member Function Documentation

5.20.3.1 destroyBlock()

```
def resources.entities.explosion.Explosion.destroyBlock (
              self,
              gameinfo = None )
```

Destroy nearby blocks.

Parameters

gameinfo	ReadMap object representing the level environment

5.20.3.2 destroyEnt()

```
def resources.entities.explosion.Explosion.destroyEnt (
              self,
              gameinfo = None )
```

Damage nearby entities.

Parameters

gameinfo	ReadMap object representing the level environment

5.20.3.3 destroyPlayer()

```
def resources.entities.explosion.Explosion.destroyPlayer (
              self,
              player = None )
```

Damage the nearby player.

Parameters

player	Mover object representing the player

5.20.3.4 explode()

```
def resources.entities.explosion.Explosion.explode (
              self,
              gameinfo = None,
              player = None )
```

Destroy blocks, player, and entities.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.20.3.5 tick()

```
def resources.entities.explosion.Explosion.tick (
              self,
              gameinfo = None,
              player = None )
```

Update the explosion with 1 unit of time.

Reduce the time. If time reaches 0, remove the explosion from the game.

Parameters

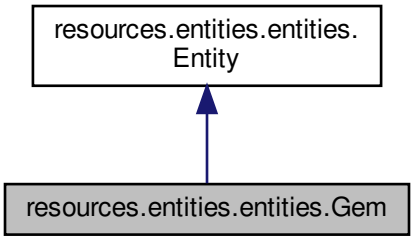
gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

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

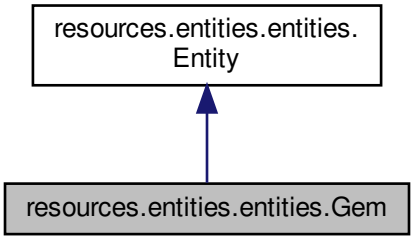
- resources/entities/explosion.py

5.21 resources.entities.entities.Gem Class Reference

Inheritance diagram for resources.entities.entities.Gem:



Collaboration diagram for resources.entities.entities.Gem:



Public Member Functions

- def **use** (self, gameinfo=None, player=None)

Additional Inherited Members

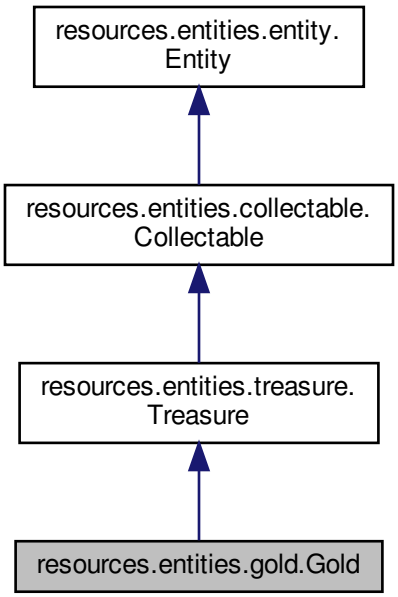
The documentation for this class was generated from the following file:

- Pylunky/pylunky-master/resources/entities/entities.py

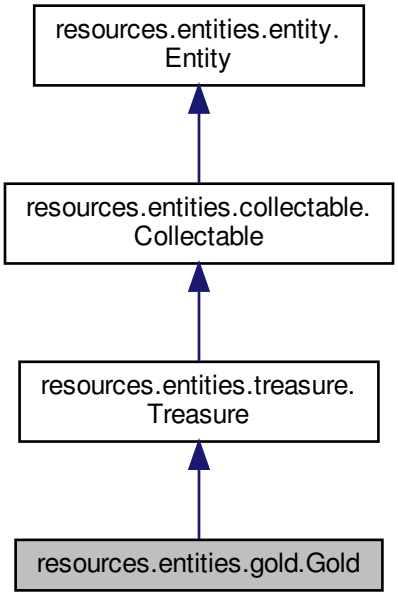
5.22 resources.entities.gold.Gold Class Reference

Gold is an object of type Treasure.

Inheritance diagram for resources.entities.gold.Gold:



Collaboration diagram for resources.entities.gold.Gold:



Additional Inherited Members

5.22.1 Detailed Description

Gold is an object of type Treasure.

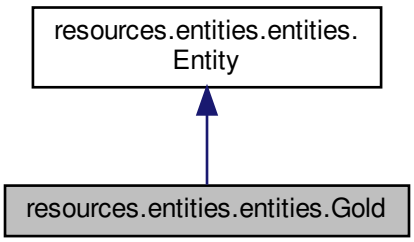
Extends from treasure.py and implements the secrets of M4.4

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

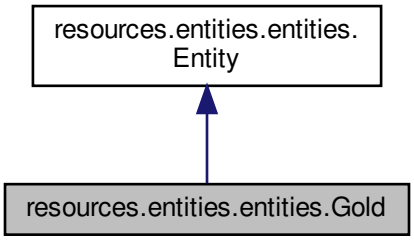
- resources/entities/gold.py

5.23 resources.entities.entities.Gold Class Reference

Inheritance diagram for resources.entities.entities.Gold:



Collaboration diagram for resources.entities.entities.Gold:



Public Member Functions

- def **use** (self, gameinfo=None, player=None)

Additional Inherited Members

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

- Pylunky/pylunky-master/resources/entities/entities.py

5.24 readmap.MapCell Class Reference

Details of map blocks.

Public Member Functions

- def **__init__** (self, posx=None, posy=None, mat=None, trans=None, soli=None, bittype=None, name=None)
- def **__init__** (self, posx=None, posy=None, mat=None, trans=None, soli=None, bittype=None, name=None)

Public Attributes

- **mat**
- **posx**
- **posy**
- **trans**
- **soli**
- **bittype**
- **name**

5.24.1 Detailed Description

Details of map blocks.

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

- Pylunky/pylunky-master/readmap.py

5.25 readmap.MapObj Class Reference

Map of all blocks.

Public Member Functions

- def **__init__** (self, add)
- def **tile** (self, x, y)
- def **map** (self)
- def **height** (self)
- def **width** (self)
- def __init__ (self, add)

  Create the map.
- def filter (self, splitmap)

  Filter the map.
- def changeBlock (self, x, y, type)

  Change a block.
- def tile (self, x, y)

  Get a block.
- def map (self)

  Get block map.
- def height (self)

  Get block map height.
- def width (self)

  Get block map width.

Public Attributes

- **start**
- **mapinfo**
- **end**

5.25.1 Detailed Description

Map of all blocks.

Implements the secrets of M5

5.25.2 Constructor & Destructor Documentation

5.25.2.1 __init__()

```
def readmap.MapObj.__init__ (
              self,
              add )
```

Create the map.

Create the block map from the map data

Parameters

add	File containing map data

5.25.3 Member Function Documentation

5.25.3.1 changeBlock()

```
def readmap.MapObj.changeBlock (
              self,
              x,
              y,
              type )
```

Change a block.

Parameters

x	int representing block x coordinate
y	int representing block y coordinate
type	string representing block type

5.25.3.2 filter()

```
def readmap.MapObj.filter (
              self,
              splitmap )
```

Filter the map.

Remove white space from the map data

Parameters

splitmap	File containing map data

Returns

File containing filtered map data

5.25.3.3 height()

```
def readmap.MapObj.height (
              self )
```

Get block map height.

Returns

int representing block map height

5.25.3.4 map()

```
def readmap.MapObj.map (
              self )
```

Get block map.

Returns

2D array of blocks representing block map

5.25.3.5 tile()

```
def readmap.MapObj.tile (
              self,
              x,
              y )
```

Get a block.

Parameters

x	int representing block x coordinate
y	int representing block y coordinate
MapCell	object representing the block

5.25.3.6 width()

```
def readmap.MapObj.width (
              self )
```

Get block map width.

Returns

int representing block map width

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

- Pylunky/pylunky-master/readmap.py

5.26 mover.Mover Class Reference

Mover is a class that implements the player of the game and implements the secrets of M3 and M8 and M2.

Public Member Functions

- def __init__ (self, x=None, y=None, direction=None, speed=None)
  constructor method for class Mover, intializes a Mover from a given player characteristics
- def position (self)
  get the position of the player Mover object
- def damage (self, d=0)
  decrease the player Mover object health points when it is under attack
- def loop (self, gameinfo=None)
  manages the player Mover object change of movement in the game run
- def gravity (self, gameinfo=None)
  manages the player Mover object change of y direction movement in the game run
- def jump (self, gameinfo=None, maxjump=None)
  manages the player Mover object jump movement in the game run
- def move (self, gameinfo=None, x=None, y=None)
  manages the player Mover object change of x and y direction movement in the game run
- def climb (self, gameinfo=None, y=None)
  manages the player Mover object change of movement when climbing in the game run
- def emptyHand (self)
  manages the player Mover object hand state in the game run
- def changeState (self, state, dir)
  manages the player Mover object change of sprite in the game run
- def controls (self, gameinfo)
  manages the player Mover object input controls and updates the game and player state accordingly
- def **__init__** (self, x=None, y=None, direction=None, speed=None)
- def **loop** (self, gameinfo=None, screen=None)
- def **gravity** (self, gameinfo=None, gravity=None, maxjump=None)
- def **Move** (self, gameinfo=None, x=None, y=None)
- def **controls** (self, key, gameinfo, screen)

Public Attributes

- **state**
- **xs**
- **ys**
- **invincibility**
- **INVINCIBILITYTIME**
- **STOMPDAMAGE**
- **jumpDist**
- **GRAVITY**
- **MAXJUMP**
- **bombs**
- **ropes**
- **adjustCamera**
- **x**
- **y**
- **hp**
- **direction**
- **gold**
- **media**
- **sizes**
- **hand**
- **image**
- **width**
- **height**
- **leave**
- **w**
- **h**
- **speed**
- **jump**
- **MOVE**

5.26.1 Detailed Description

Mover is a class that implements the player of the game and implements the secrets of M3 and M8 and M2.

Implements the secrets of M2, M3, M8. The class contains the different player controls and the movement animation of the player on the screen

5.26.2 Constructor & Destructor Documentation

5.26.2.1 __init__()

```
def mover.Mover.__init__ (
              self,
              x = None,
              y = None,
              direction = None,
              speed = None )
```

constructor method for class Mover, intializes a Mover from a given player characteristics

Parameters

x	is an integer element indicating the x position of the player Mover object
y	is an integer element indicating the y position of the player Mover object
direction	is an integer element indicating the direction of movement of the player Mover object
speed	is an integer element indicating the speed of movement of the player Mover object

5.26.3 Member Function Documentation

5.26.3.1 changeState()

```
def mover.Mover.changeState (
              self,
              state,
              dir )
```

manages the player Mover object change of sprite in the game run

Parameters

state	a State type input indicating the change in the Mover object sprite
dir	an integer indicating the change in direction of the Mover object

5.26.3.2 climb()

```
def mover.Mover.climb (
              self,
              gameinfo = None,
              y = None )
```

manages the player Mover object change of movement when climbing in the game run

Parameters

gameinfo	a ReadMap object indicating the map where the Mover object is on
y	an integer indicating the change in the y direction move of the Mover object

5.26.3.3 controls()

```
def mover.Mover.controls (
```

```
                self,
                gameinfo )
```

manages the player Mover object input controls and updates the game and player state accordingly

Parameters

key	a keyboard input press by the player
gameinfo	a ReadMap object indicating the map where the Mover object is on

5.26.3.4 damage()

```
def mover.Mover.damage (
                self,
                d = 0 )
```

decrease the player Mover object health points when it is under attack

Parameters

d	is an integer element indicating the damage on the player Mover object health points

Returns

nothing if the player is not damaged

5.26.3.5 gravity()

```
def mover.Mover.gravity (
                self,
                gameinfo = None )
```

manages the player Mover object change of y direction movement in the game run

Parameters

gameinfo	a ReadMap object element indicating the map where the Mover object is on

Returns

nothing if the player state is climb

5.26.3.6 jump()

```
def mover.Mover.jump (
              self,
              gameinfo = None,
              maxjump = None )
```

manages the player Mover object jump movement in the game run

Parameters

gameinfo	a ReadMap object indicating the map where the Mover object is on
maxjump	an integer representing the max height of the Mover jump in the y direction

5.26.3.7 loop()

```
def mover.Mover.loop (
              self,
              gameinfo = None )
```

manages the player Mover object change of movement in the game run

Parameters

gameinfo	a ReadMap object element indicating the map where the Mover object is on

5.26.3.8 move()

```
def mover.Mover.move (
              self,
              gameinfo = None,
              x = None,
              y = None )
```

manages the player Mover object change of x and y direction movement in the game run

Parameters

gameinfo	a ReadMap object indicating the map where the Mover object is on
x	an integer indicating the change in the x direction move of the Mover object
y	an integer indicating the change in the y direction move of the Mover object

5.26.3.9 position()

```
def mover.Mover.position (
              self )
```

get the position of the player Mover object

Returns

tuple of integers indicating the x and y position of the player Mover object

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

- mover.py

5.27 readmap.ReadMap Class Reference

Map of all blocks and entities.

Public Member Functions

- def **__init__** (self, file=None)
- def __init__ (self, file=None)
  Create the map.
- def entities (self)
  Get the entities.
- def enemies (self)
  Get the enemies.
- def add (self, ent)
  Add an entity.
- def rem (self, ent)
  Remove an entity.
- def solid (self, x, y)
  Determine if a block is solid.
- def destroyBlock (self, x, y)
  Destroy a block.

Public Attributes

- **gamemap**
- **entlist**

5.27.1 Detailed Description

Map of all blocks and entities.

Implements the secrets of M5

5.27.2 Constructor & Destructor Documentation

5.27.2.1 __init__()

```
def readmap.ReadMap.__init__ (
              self,
              file = None )
```

Create the map.

Create the block map and entities from the map data

Parameters

file	File containing map data

5.27.3 Member Function Documentation

5.27.3.1 add()

```
def readmap.ReadMap.add (
              self,
              ent )
```

Add an entity.

Parameters

ent	Entity object representing the entity to add

5.27.3.2 destroyBlock()

```
def readmap.ReadMap.destroyBlock (
              self,
              x,
              y )
```

Destroy a block.

Parameters

x	int representing block x coordinate
y	int representing block y coordinate

5.27.3.3 enemies()

```
def readmap.ReadMap.enemies (
              self )
```

Get the enemies.

Returns

Array of Enemy objects representing the enemies

5.27.3.4 entities()

```
def readmap.ReadMap.entities (
              self )
```

Get the entities.

Returns

Array of Entity objects representing the entities

5.27.3.5 rem()

```
def readmap.ReadMap.rem (
              self,
              ent )
```

Remove an entity.

Parameters

ent	Entity object representing the entity to remove

5.27.3.6 solid()

```
def readmap.ReadMap.solid (
              self,
              x,
              y )
```

Determine if a block is solid.

Parameters

x	int representing block x coordinate
y	int representing block y coordinate

Returns

Whether or not the block is solid

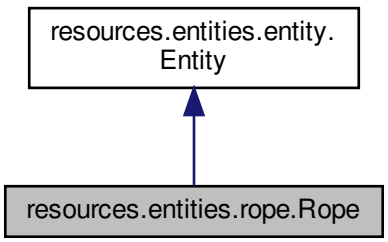
The documentation for this class was generated from the following file:

- Pylunky/pylunky-master/readmap.py

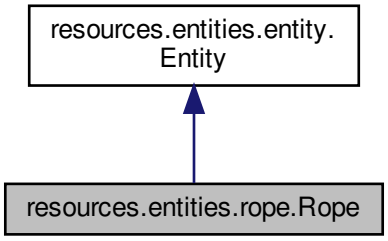
5.28 resources.entities.rope.Rope Class Reference

Rope is an object of type Entity.

Inheritance diagram for resources.entities.rope.Rope:



Collaboration diagram for resources.entities.rope.Rope:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Rope.
- def throw (self)

  Set the rope movement.
- def tick (self, gameinfo=None, player=None)

  Update the rope with 1 unit of time.
- def move (self, gameinfo=None, y=None)

  Move the rope.
- def makeRope (self, gameinfo=None)

  Make the rope.
- def findDist (self, gameinfo=None)

  Find the number of empty blocks downwards until the next solid block.

Public Attributes

- **LENGTH**
- **ys**
- **dist**
- **y**
- **mat**
- **x**
- **height**

5.28.1 Detailed Description

Rope is an object of type Entity.

Extends from entity.py and implements the secrets of M4.13

5.28.2 Constructor & Destructor Documentation

5.28.2.1 __init__()

```
def resources.entities.rope.Rope.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Rope.

Create the rope using super(). Set the dist to LENGTH.

Parameters

x	int representing the x position of the rope
y	int representing the y position of the rope
hp	int representing the health points of the rope
mat	image representing the rope on the game screen
name	string representing the name of the image of the object
width	int representing the width of the rope
height	int representing the height of the rope

5.28.3 Member Function Documentation

5.28.3.1 findDist()

```
def resources.entities.rope.Rope.findDist (
              self,
              gameinfo = None )
```

Find the number of empty blocks downwards until the next solid block.

Parameters

gameinfo	ReadMap object representing the level environment

Returns

int representing the number of empty blocks until the next solid block

5.28.3.2 makeRope()

```
def resources.entities.rope.Rope.makeRope (
              self,
              gameinfo = None )
```

Make the rope.

Create a rope with length dependent on how far it can extend downwards without reaching a block. Add it to the game

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.28.3.3 move()

```
def resources.entities.rope.Rope.move (
              self,
              gameinfo = None,
              y = None )
```

Move the rope.

If there is no block above and the rope can still travel, move it upwards. Else, stop moving and make the rope.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.28.3.4 tick()

```
def resources.entities.rope.Rope.tick (
              self,
              gameinfo = None,
              player = None )
```

Update the rope with 1 unit of time.

If the rope is moving, move it.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

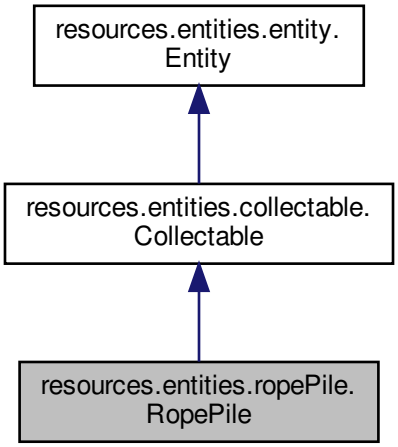
The documentation for this class was generated from the following file:

- resources/entities/rope.py

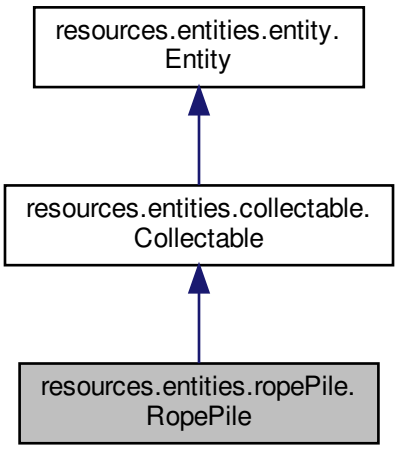
5.29 resources.entities.ropePile.RopePile Class Reference

RopePile is an object of type Collectable.

Inheritance diagram for resources.entities.ropePile.RopePile:



Collaboration diagram for resources.entities.ropePile.RopePile:



Public Member Functions

- def use (self, gameinfo=None, player=None)

  Use the ropePile.

Additional Inherited Members

5.29.1 Detailed Description

RopePile is an object of type Collectable.

Extends from collectable.py and implements the secrets of M4.10

5.29.2 Member Function Documentation

5.29.2.1 use()

```
def resources.entities.ropePile.RopePile.use (
              self,
              gameinfo = None,
              player = None )
```

Use the ropePile.

Give the player ropes equal to the ropePile's value.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

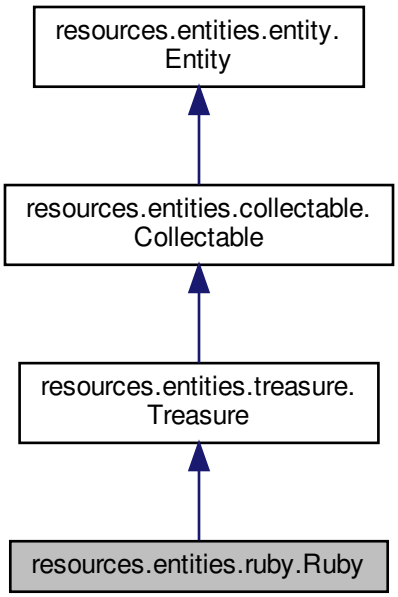
The documentation for this class was generated from the following file:

- resources/entities/ropePile.py

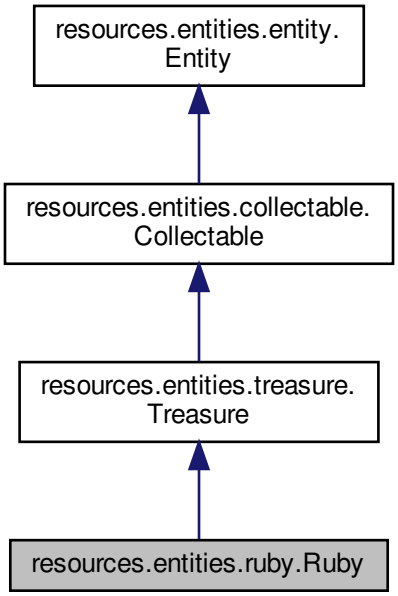
5.30 resources.entities.ruby.Ruby Class Reference

Ruby is an object of type Treasure.

Inheritance diagram for resources.entities.ruby.Ruby:



Collaboration diagram for resources.entities.ruby.Ruby:



Additional Inherited Members

5.30.1 Detailed Description

Ruby is an object of type Treasure.

Extends from treasure.py and implements the secrets of M4.7

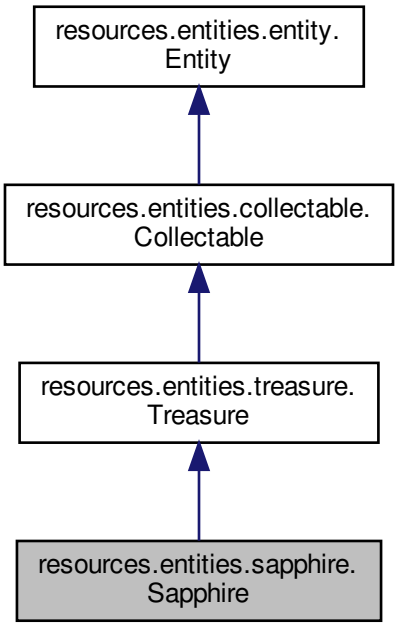
The documentation for this class was generated from the following file:

- resources/entities/ruby.py

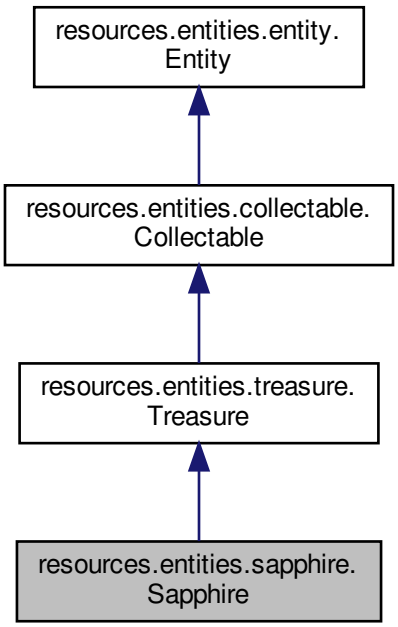
5.31 resources.entities.sapphire.Sapphire Class Reference

Sapphire is an object of type Treasure.

Inheritance diagram for resources.entities.sapphire.Sapphire:



Collaboration diagram for resources.entities.sapphire.Sapphire:



Additional Inherited Members

5.31.1 Detailed Description

Sapphire is an object of type Treasure.

Extends from treasure.py and implements the secrets of M4.5

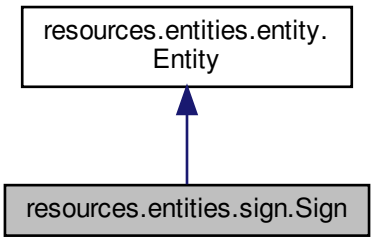
The documentation for this class was generated from the following file:

- resources/entities/sapphire.py

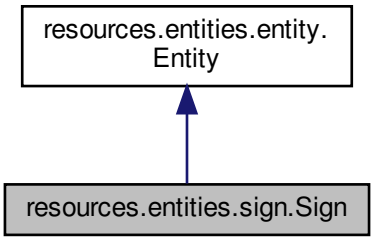
5.32 resources.entities.sign.Sign Class Reference

Sign is an object of type Entity.

Inheritance diagram for resources.entities.sign.Sign:



Collaboration diagram for resources.entities.sign.Sign:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Sign.
- def setText (self, s)

  Set the test.
- def getText (self)

  Get the text.

Public Attributes

- **text**

5.32.1 Detailed Description

Sign is an object of type Entity.

Extends from entity.py and implements the secrets of M4.11

5.32.2 Constructor & Destructor Documentation

5.32.2.1 __init__()

```
def resources.entities.sign.Sign.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Sign.

Create the sign using super(). Set the text.

Parameters

x	int representing the x position of the sign
y	int representing the y position of the sign
hp	int representing the health points of the sign
mat	image representing the sign on the game screen
name	string representing the name of the image of the object
width	int representing the width of the sign
height	int representing the height of the sign

5.32.3 Member Function Documentation

5.32.3.1 getText()

```
def resources.entities.sign.Sign.getText (
              self )
```

Get the text.

Returns

string representing the text

5.32.3.2 setText()

```
def resources.entities.sign.Sign.setText (
              self,
              s )
```

Set the test.

Parameters

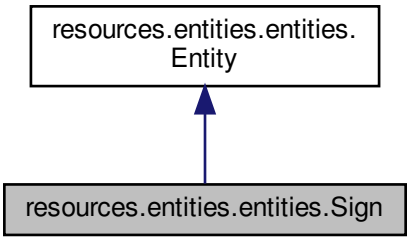
s	string representing the text

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

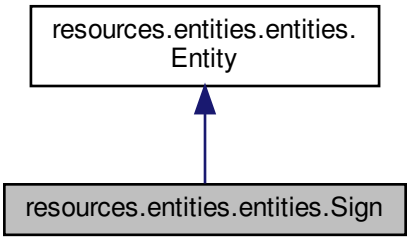
- resources/entities/sign.py

5.33 resources.entities.entities.Sign Class Reference

Inheritance diagram for resources.entities.entities.Sign:



Collaboration diagram for resources.entities.entities.Sign:



Public Member Functions

- def **use** (self, gameinfo=None, player=None)

Additional Inherited Members

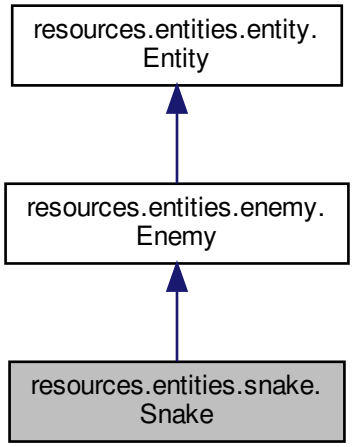
The documentation for this class was generated from the following file:

- Pylunky/pylunky-master/resources/entities/entities.py

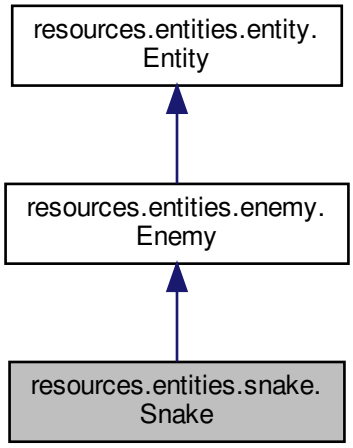
5.34 resources.entities.snake.Snake Class Reference

Snake is an object of type Enemy.

Inheritance diagram for resources.entities.snake.Snake:



Collaboration diagram for resources.entities.snake.Snake:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for snake.
- def move (self, gameinfo=None, x=None, y=None)

  Move the snake.

Public Attributes

- **xs**
- **SNAKEATTACK**
- **attack**
- **x**
- **y**

5.34.1 Detailed Description

Snake is an object of type Enemy.

Extends from enemy.py and implements the secrets of M4.19

5.34.2 Constructor & Destructor Documentation

5.34.2.1 __init__()

```
def resources.entities.snake.Snake.__init__ (
                self,
                x = None,
                y = None,
                name = None,
                height = None,
                width = None,
                hp = None,
                mat = None )
```

Constructor method for snake.

Parameters

x	an integer element indicating the x position of the snake enemy object
y	an integer element indicating the y position of the snake enemy object
hp	an integer element indicating the health points of the snake enemy object
mat	an image representing the snake object on the game screen
name	a string representing the name of the image of the object
text	a string representing additional object information
width	an integer element indicating the width of the snake object
height	an integer element indicating the height of the snake object

5.34.3 Member Function Documentation

5.34.3.1 move()

```
def resources.entities.snake.Snake.move (
              self,
              gameinfo = None,
              x = None,
              y = None )
```

Move the snake.

If there is no solid block the snake continues movement. If solid block exists switch direction of straightline movement.

Parameters

gameinfo	ReadMap object representing the level environment
x	subsequent movement by x amount on x axis
y	subsequent movement by y amount on y axis

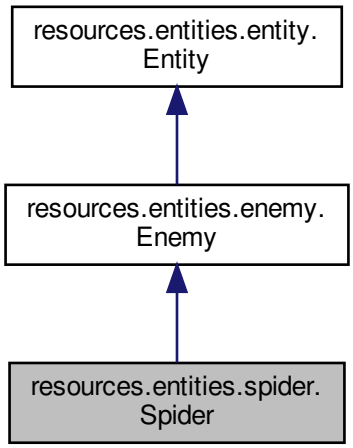
The documentation for this class was generated from the following file:

- resources/entities/snake.py

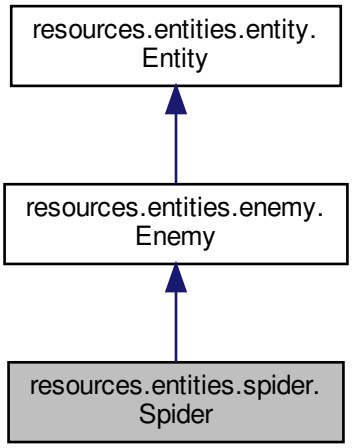
5.35 resources.entities.spider.Spider Class Reference

Spider is an object of type Enemy.

Inheritance diagram for resources.entities.spider.Spider:



Collaboration diagram for resources.entities.spider.Spider:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for spider.
- def gravity (self)

  Controls jump that is gravity restrictions of spider.
- def move (self, gameinfo=None, x=None, y=None)

  Move the spider.
- def tick (self, gameinfo=None, player=None)

  Inflicts the spider object damage on the player's health and checks the range of attack per tick.
- def sense (self, player=None)

  Sense the player based on range from the spider.

Public Attributes

- **MAXJUMP**
- **time**
- **CD**
- **jumpDist**
- **senseRange**
- **SPIDERATTACK**
- **attack**
- **ys**
- **x**
- **y**
- **xs**

5.35.1 Detailed Description

Spider is an object of type Enemy.

Extends from enemy.py and implements the secrets of M4.20

5.35.2 Constructor & Destructor Documentation

5.35.2.1 __init__()

```
def resources.entities.spider.Spider.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for spider.

Parameters

x	an integer element indicating the x position of the spider enemy object
y	an integer element indicating the y position of the spider enemy object
hp	an integer element indicating the health points of the spider enemy object
mat	an image representing the spider object on the game screen
name	a string representing the name of the image of the object
width	an integer element indicating the width of the spider object
height	an integer element indicating the height of the spider object

5.35.3 Member Function Documentation

5.35.3.1 gravity()

```
def resources.entities.spider.Spider.gravity (
              self )
```

Controls jump that is gravity restrictions of spider.

If jumping, set ys to -gravity and increase jumpDist. Else, set ys to gravity and reset jumpDist

5.35.3.2 move()

```
def resources.entities.spider.Spider.move (
              self,
              gameinfo = None,
              x = None,
              y = None )
```

Move the spider.

If there is no solid block the spider can continue to jump based on the spider sense wrt the player's position

Parameters

gameinfo	ReadMap object representing the level environment
x	subsequent movement by x amount on x axis
y	subsequent movement by y amount on y axis

5.35.3.3 sense()

```
def resources.entities.spider.Spider.sense (
              self,
              player = None )
```

Sense the player based on range from the spider.

Parameters

player	Mover object representing the player

5.35.3.4 tick()

```
def resources.entities.spider.Spider.tick (
              self,
              gameinfo = None,
              player = None )
```

Inflicts the spider object damage on the player's health and checks the range of attack per tick.

Parameters

gameinfo	a ReadMap object element indicating the map where the spider object is on
player	a Mover object element representing the player of the game that is on the spider object

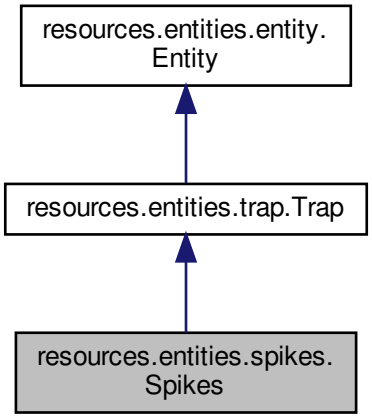
The documentation for this class was generated from the following file:

- resources/entities/spider.py

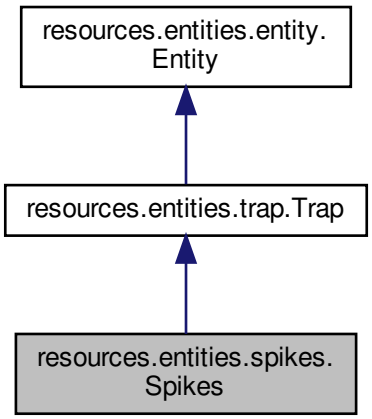
5.36 resources.entities.spikes.Spikes Class Reference

Spike is an object of type Trap.

Inheritance diagram for resources.entities.spikes.Spikes:



Collaboration diagram for resources.entities.spikes.Spikes:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Spike.
- def tick (self, gameinfo=None, player=None)

  Inflicts the spike trap damage on the player's health.

Public Attributes

- **spikeDmg**

5.36.1 Detailed Description

Spike is an object of type Trap.

extends from trap.py and implements the secrets of M4.16

5.36.2 Constructor & Destructor Documentation

5.36.2.1 __init__()

```
def resources.entities.spikes.Spikes.__init__ (
                self,
                x = None,
                y = None,
                name = None,
                height = None,
                width = None,
                hp = None,
                mat = None )
```

Constructor method for Spike.

Parameters

x	an integer element indicating the x position of the Spike trap object
y	an integer element indicating the y position of the Spike trap object
hp	an integer element indicating the health points of the spike trap object
mat	an image representing the spike trap object on the game screen
name	a string representing the name of the image of the object
width	an integer element indicating the width of the spike object
height	an integer element indicating the height of the spike object

5.36.3 Member Function Documentation

5.36.3.1 tick()

```
def resources.entities.spikes.Spikes.tick (
              self,
              gameinfo = None,
              player = None )
```

Inflicts the spike trap damage on the player's health.

Parameters

gameinfo	a ReadMap object element indicating the map where the spike object is on
player	a Mover object element representing the player of the game that is on the spike object

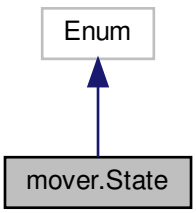
The documentation for this class was generated from the following file:

- resources/entities/spikes.py

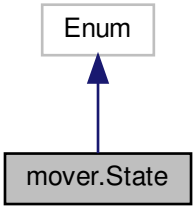
5.37 mover.State Class Reference

State contains an enumeration for the player sprite in the game and implements the secrets of M3.

Inheritance diagram for mover.State:



Collaboration diagram for mover.State:



Static Public Attributes

- int **standing** = 0
- int **walking** = 1
- int **falling** = 2
- int **climbing** = 3
- int **crouching** = 4

5.37.1 Detailed Description

State contains an enumeration for the player sprite in the game and implements the secrets of M3.

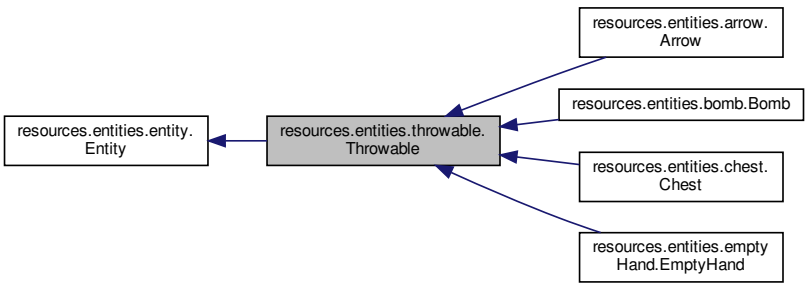
The documentation for this class was generated from the following file:

- mover.py

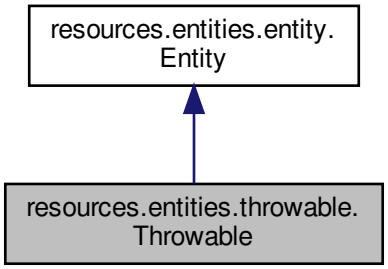
5.38 resources.entities.throwable.Throwable Class Reference

Throwable is an object of type Entity.

Inheritance diagram for resources.entities.throwable.Throwable:



Collaboration diagram for resources.entities.throwable.Throwable:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Throwable.
- def pickup (self)

  Set held to True.
- def putDown (self)

  Set held to False.
- def throw (self, xs=None, ys=None)

  Throw the throeable.
- def flip (self)

  Reverse image and direction.
- def rem (self, gameinfo=None, player=None)

  Remove the throwable.
- def tick (self, gameinfo=None, player=None)

  Update the throwable with 1 unit of time.
- def gravity (self, gameinfo=None)

  Adjust movement of the throwable.
- def move (self, gameinfo=None, x=None, y=None)

  Move the throwable.
- def carry (self, gameinfo=None, x=None, y=None)

  Carry the throwable.

Public Attributes

- **xs**
- **ys**
- **held**
- **thrown**
- **direction**
- **GRAVITY**
- **TOOFASTFORGRAVITY**
- **x**
- **y**

5.38.1 Detailed Description

Throwable is an object of type Entity.

Extends from entity.py and implements the secrets of M4.14

5.38.2 Constructor & Destructor Documentation

5.38.2.1 __init__()

```
def resources.entities.throwable.Throwable.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Throwable.

Create the throwable using super(). Set the speeds to 0, possesions to False.

Parameters

x	int representing the x position of the throwable
y	int representing the y position of the throwable
hp	int representing the health points of the throwable
mat	image representing the throwable on the game screen
name	string representing the name of the image of the object
width	int representing the width of the throwable
height	int representing the height of the throwable

5.38.3 Member Function Documentation

5.38.3.1 carry()

```
def resources.entities.throwable.Throwable.carry (
              self,
              gameinfo = None,
              x = None,
              y = None )
```

Carry the throwable.

Carry the throwable while ignoring solid blocks.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.38.3.2 gravity()

```
def resources.entities.throwable.Throwable.gravity (
              self,
              gameinfo = None )
```

Adjust movement of the throwable.

If xs is not grater than TOOFASTFORGRAVITY, the throwable is affected by gravity.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.38.3.3 move()

```
def resources.entities.throwable.Throwable.move (
              self,
              gameinfo = None,
              x = None,
              y = None )
```

Move the throwable.

Move the throwable, If it collides with a solid block, stop it.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player def move(self, gameinfo=None, x=None, y=None):

5.38.3.4 rem()

```
def resources.entities.throwable.Throwable.rem (
              self,
```

```
                gameinfo = None,
                player = None )
```

Remove the throwable.

Remove the explosion from the game, and empty the player's hand.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.38.3.5 throw()

```
def resources.entities.throwable.Throwable.throw (
                self,
                xs = None,
                ys = None )
```

Throw the throeable.

Set the speeds and possesion.

Parameters

xs	int representing the x speed
ys	int representing the y speed

5.38.3.6 tick()

```
def resources.entities.throwable.Throwable.tick (
                self,
                gameinfo = None,
                player = None )
```

Update the throwable with 1 unit of time.

If not held, the throwable is affected by gravity.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

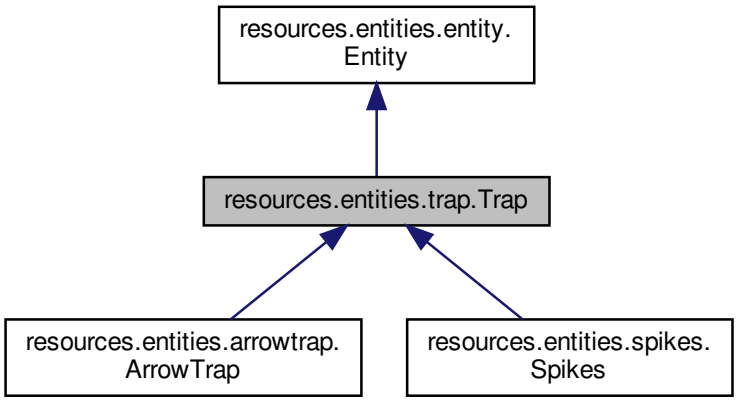
The documentation for this class was generated from the following file:

- resources/entities/throwable.py

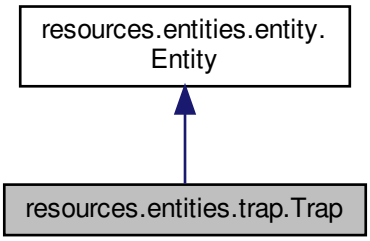
5.39 resources.entities.trap.Trap Class Reference

Trap is an object of type Entity.

Inheritance diagram for resources.entities.trap.Trap:



Collaboration diagram for resources.entities.trap.Trap:



Public Member Functions

- def **damage** (self, d=0, gameinfo=None)

Additional Inherited Members

5.39.1 Detailed Description

Trap is an object of type Entity.

Extends from entities.py and implements the secrets of M4.13

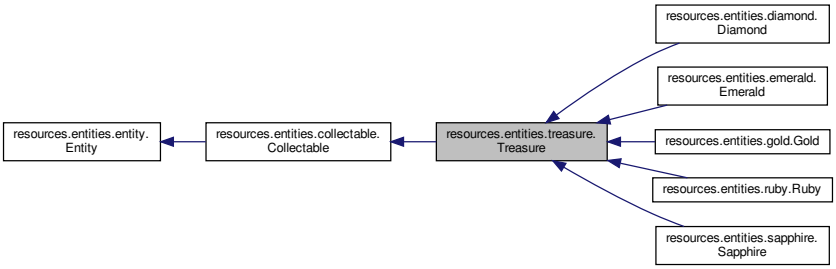
The documentation for this class was generated from the following file:

- resources/entities/trap.py

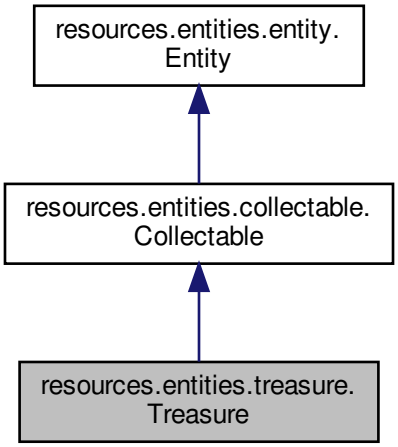
5.40 resources.entities.treasure.Treasure Class Reference

Treasure is an object of type Collectable.

Inheritance diagram for resources.entities.treasure.Treasure:



Collaboration diagram for resources.entities.treasure.Treasure:



Public Member Functions

- def use (self, gameinfo=None, player=None)

  Use the treasure.

Additional Inherited Members

5.40.1 Detailed Description

Treasure is an object of type Collectable.

Extends from collectable.py and implements the secrets of M4.3

5.40.2 Member Function Documentation

5.40.2.1 use()

```
def resources.entities.treasure.Treasure.use (
              self,
              gameinfo = None,
              player = None )
```

Use the treasure.

Give the player gold equal to the treasure's value.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

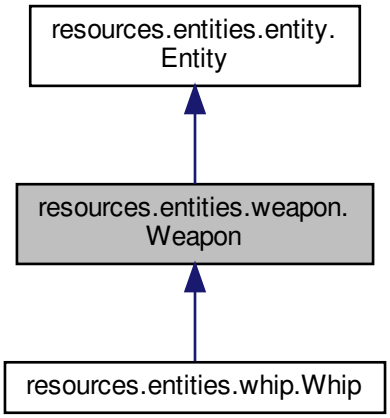
The documentation for this class was generated from the following file:

- resources/entities/treasure.py

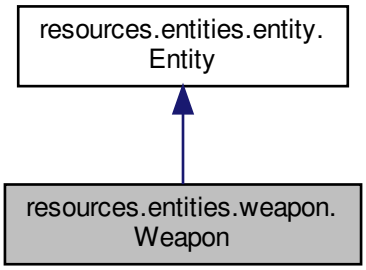
5.41 resources.entities.weapon.Weapon Class Reference

Weapon is an object of type Entity.

Inheritance diagram for resources.entities.weapon.Weapon:



Collaboration diagram for resources.entities.weapon.Weapon:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)

  Constructor method for Weapon.
- def swing (self, gameinfo=None)

  Swing the weapon.
- def tick (self, gameinfo=None, player=None)

  Update the weapon with 1 unit of time.
- def carry (self, gameinfo=None, x=None, y=None)

  Carry the weapon.
- def flip (self)

  Reverse image and direction.

Public Attributes

- **damage**
- **speed**
- **activeFrames**
- **time**
- **active**
- **direction**

5.41.1 Detailed Description

Weapon is an object of type Entity.

Extends from entity.py and implements the secrets of M4.25

5.41.2 Constructor & Destructor Documentation

5.41.2.1 __init__()

```
def resources.entities.weapon.Weapon.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Weapon.

Create the weapon using super(). Set time and speed to 0.

Parameters

x	int representing the x position of the weapon
y	int representing the y position of the weapon
hp	int representing the health points of the weapon
mat	image representing the weapon on the game screen
name	string representing the name of the image of the object
width	int representing the width of the weapon
height	int representing the height of the weapon

5.41.3 Member Function Documentation

5.41.3.1 carry()

```
def resources.entities.weapon.Weapon.carry (
              self,
              gameinfo = None,
              x = None,
              y = None )
```

Carry the weapon.

Carry the weapon while ignoring solid blocks.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

5.41.3.2 swing()

```
def resources.entities.weapon.Weapon.swing (
              self,
              gameinfo = None )
```

Swing the weapon.

If there is no time remaining, set time to SPEED, active to ACTIVEFRAMES, and add the weapon to the game.

Parameters

gameinfo	ReadMap object representing the level environment

5.41.3.3 tick()

```
def resources.entities.weapon.Weapon.tick (
              self,
              gameinfo = None,
              player = None )
```

Update the weapon with 1 unit of time.

If there is time remaining, reduce it. If the weapon is active, damage any overlapping entities. Else, remove it from the player's hand.

Parameters

gameinfo	ReadMap object representing the level environment
player	Mover object representing the player

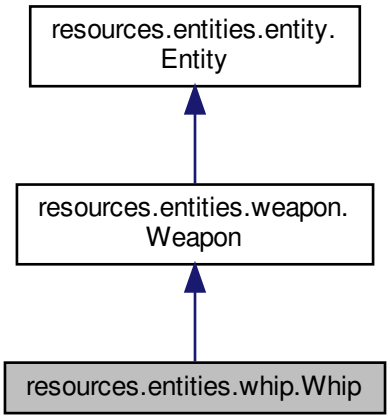
The documentation for this class was generated from the following file:

- resources/entities/weapon.py

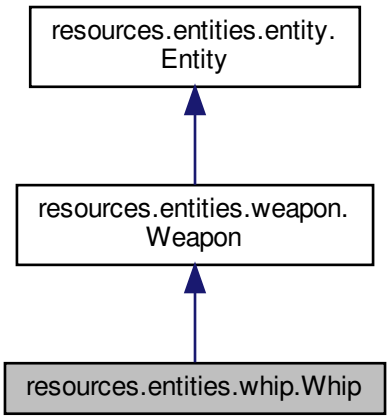
5.42 resources.entities.whip.Whip Class Reference

Whip is an object of type Weapon.

Inheritance diagram for resources.entities.whip.Whip:



Collaboration diagram for resources.entities.whip.Whip:



Public Member Functions

- def __init__ (self, x=None, y=None, name=None, height=None, width=None, hp=None, mat=None)
  Constructor method for Whip.

Public Attributes

- **damage**
- **speed**
- **activeFrames**

5.42.1 Detailed Description

Whip is an object of type Weapon.

Extends from weapon.py and implements the secrets of M4.26

5.42.2 Constructor & Destructor Documentation

5.42.2.1 __init__()

```
def resources.entities.whip.Whip.__init__ (
              self,
              x = None,
              y = None,
              name = None,
              height = None,
              width = None,
              hp = None,
              mat = None )
```

Constructor method for Whip.

Create the whip using super().

Parameters

x	int representing the x position of the whip
y	int representing the y position of the whip
hp	int representing the health points of the whip
mat	image representing the whip on the game screen
name	string representing the name of the image of the object
width	int representing the width of the whip
height	int representing the height of the whip

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

- resources/entities/whip.py

Chapter 6

File Documentation

6.1 resources/entities/arrow.py File Reference

Contain the specific Arrow type represented by a Throwable.

Classes

- class resources.entities.arrow.Arrow
  Arrow is an object of type Throwable.

6.1.1 Detailed Description

Contain the specific Arrow type represented by a Throwable.

Author

Abeer

Date

04/11/2021

6.2 resources/entities/arrowtrap.py File Reference

Contain the specific ArrowTrap type represented by an Trap(Entity)

Classes

- class resources.entities.arrowtrap.ArrowTrap
  ArrowTrap is an object of type Trap.

6.2.1 Detailed Description

Contain the specific ArrowTrap type represented by an Trap(Entity)

Author

Abeer

Date

04/11/2021

6.3 resources/entities/bomb.py File Reference

Contain the specific Bomb type.

Classes

- class resources.entities.bomb.Bomb

  Bomb is an object of type Throwable.

6.3.1 Detailed Description

Contain the specific Bomb type.

Author

Albert

Date

04/11/2021

6.4 resources/entities/bombPile.py File Reference

Contain the specific BombPile type.

Classes

- class resources.entities.bombPile.BombPile

  BombPile is an object of type Collectable.

6.4.1 Detailed Description

Contain the specific BombPile type.

Author

Albert

Date

04/11/2021

6.5 resources/entities/chest.py File Reference

Contain the specific Chest type.

Classes

- class resources.entities.chest.Chest

  Chest is an object of type Throwable.

6.5.1 Detailed Description

Contain the specific Chest type.

Author

Albert

Date

04/11/2021

6.6 resources/entities/collectable.py File Reference

Contain the specific Collectable type.

Classes

- class resources.entities.collectable.Collectable

  Collectable is an object of type Entity.

6.6.1 Detailed Description

Contain the specific Collectable type.

Author

Albert

Date

04/11/2021

6.7 resources/entities/diamond.py File Reference

Contain the specific Diamond type.

Classes

- class resources.entities.diamond.Diamond

  Diamond is an object of type Treasure.

6.7.1 Detailed Description

Contain the specific Diamond type.

Author

Albert

Date

04/11/2021

6.8 resources/entities/emerald.py File Reference

Contain the specific Emerald type.

Classes

- class resources.entities.emerald.Emerald

  Emerald is an object of type Treasure.

6.8.1 Detailed Description

Contain the specific Emerald type.

Author

Albert

Date

04/11/2021

6.9 resources/entities/emptyHand.py File Reference

Contain the specific EmptyHand type.

Classes

- class resources.entities.emptyHand.EmptyHand

  EmptyHand is an object of type Throwable.

6.9.1 Detailed Description

Contain the specific EmptyHand type.

Author

Albert

Date

04/11/2021

6.10 resources/entities/enemy.py File Reference

Contain the specific Enemy type.

Classes

- class resources.entities.enemy.Enemy

  Enemy is an object of type Entity.

6.10.1 Detailed Description

Contain the specific Enemy type.

Author

Niyatha

Date

04/11/2021

6.11 resources/entities/entity.py File Reference

Contain the specific Entity type.

Classes

- class resources.entities.entity.Entity

  Entity is an object.

6.11.1 Detailed Description

Contain the specific Entity type.

Author

Albert

Date

04/11/2021

6.12 resources/entities/explosion.py File Reference

Contain the specific Explosion type.

Classes

- class resources.entities.explosion.Explosion

  Explosion is an object of type Entity.

6.12.1 Detailed Description

Contain the specific Explosion type.

Author

Albert

Date

04/11/2021

6.13 resources/entities/gold.py File Reference

Contain the specific Gold type.

Classes

- class resources.entities.gold.Gold
  Gold is an object of type Treasure.

6.13.1 Detailed Description

Contain the specific Gold type.

Author

Albert

Date

04/11/2021

6.14 resources/entities/rope.py File Reference

Contain the specific Rope type.

Classes

- class resources.entities.rope.Rope
  Rope is an object of type Entity.

6.14.1 Detailed Description

Contain the specific Rope type.

Author

Albert

Date

04/11/2021

6.15 resources/entities/ropePile.py File Reference

Contain the specific RopePile type.

Classes

- class resources.entities.ropePile.RopePile

  RopePile is an object of type Collectable.

6.15.1 Detailed Description

Contain the specific RopePile type.

Author

Albert

Date

04/11/2021

6.16 resources/entities/ruby.py File Reference

Contain the specific Ruby type.

Classes

- class resources.entities.ruby.Ruby

  Ruby is an object of type Treasure.

6.16.1 Detailed Description

Contain the specific Ruby type.

Author

Albert

Date

04/11/2021

6.17 resources/entities/sapphire.py File Reference

Contain the specific Sapphire type.

Classes

- class resources.entities.sapphire.Sapphire

  Sapphire is an object of type Treasure.

6.17.1 Detailed Description

Contain the specific Sapphire type.

Author

Albert

Date

04/11/2021

6.18 resources/entities/sign.py File Reference

Contain the specific Sign type.

Classes

- class resources.entities.sign.Sign

  Sign is an object of type Entity.

6.18.1 Detailed Description

Contain the specific Sign type.

Author

Albert

Date

04/11/2021

6.19 resources/entities/snake.py File Reference

Contain the specific Snake type.

Classes

- class resources.entities.snake.Snake
  Snake is an object of type Enemy.

6.19.1 Detailed Description

Contain the specific Snake type.

Author

Niyatha

Date

04/11/2021

6.20 resources/entities/spider.py File Reference

Contain the specific Spider type.

Classes

- class resources.entities.spider.Spider
  Spider is an object of type Enemy.

6.20.1 Detailed Description

Contain the specific Spider type.

Author

Niyatha

Date

03/29/2021

6.21 resources/entities/throwable.py File Reference

Contain the specific Throwable type.

Classes

- class resources.entities.throwable.Throwable

  Throwable is an object of type Entity.

6.21.1 Detailed Description

Contain the specific Throwable type.

Author

Albert

Date

04/11/2021

6.22 resources/entities/trap.py File Reference

Contain the specific Trap type represented by an entity.

Classes

- class resources.entities.trap.Trap

  Trap is an object of type Entity.

6.22.1 Detailed Description

Contain the specific Trap type represented by an entity.

Author

Abeer

Date

04/11/2021

6.23 resources/entities/treasure.py File Reference

Contain the specific Treasure type.

Classes

- class resources.entities.treasure.Treasure
  Treasure is an object of type Collectable.

6.23.1 Detailed Description

Contain the specific Treasure type.

Author

Albert

Date

04/11/2021

6.24 resources/entities/weapon.py File Reference

Contain the specific Weapon type.

Classes

- class resources.entities.weapon.Weapon
  Weapon is an object of type Entity.

6.24.1 Detailed Description

Contain the specific Weapon type.

Author

Albert

Date

04/11/2021

Index