# Tank War

Generated by Doxygen 1.8.15

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

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

# Chapter 2

# Class Index

## 2.1 Class List

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

# Chapter 3

# File Index

## 3.1 File List

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

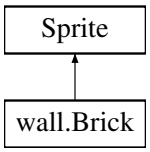
# Chapter 4

# Class Documentation

## 4.1 wall.Brick Class Reference

Brick.

Inheritance diagram for wall.Brick:




### Public Member Functions

- def __init__ (self)

  *Constructor for Brick.*

### Public Attributes

- **image**
- **rect**

### 4.1.1 Detailed Description

Brick.

This class sets up the properties of the brick wall. This class constructs a brick wall and sets the image path so that the image can be loaded on the screen. Also, the coordinate of the brick can be stored in the self.rect

### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 __init__()

```
def wall.Brick.__init__ (
              self )
```

Constructor for Brick.

Constructor initialize the brick wall, set the path for brick Image and stored in the brick objects. Also a self.rect is defined and can be used to store the coordinate of the brick wall.

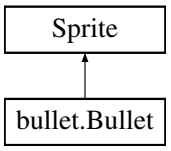
The documentation for this class was generated from the following file:

- src/wall.py

## 4.2 bullet.Bullet Class Reference

Bullet.

Inheritance diagram for bullet.Bullet:




### Public Member Functions

- def __init__ (self)

  *Constructor for Bullet.*
- def setBulletSpeed (self, speed)

  *Mutator for Bullet Speed.*
- def setBulletLife (self, life)

  *Mutator for Bullet Life.*
- def setBulletStrong (self, strong)

  *Mutator for Bullet Strong.*
- def changeImage (self, dir_x, dir_y)

  *Change the image for the bullet depends on its direction.*
- def move (self)

  *Move the bullet in its direction and make it disappear when it hits the boundaries of the map.*

### Public Attributes

- **bullet_up**
- **bullet_down**
- **bullet_left**
- **bullet_right**
- **coordinate**
- **dir_y**
- **speed**
- **life**
- **strong**
- **bullet**
- **rect**
- **dir_x**

### 4.2.1 Detailed Description

Bullet.

This class sets up Bullet's properties include operations for the bullet. This class is able to initialize itself, set the bullet's speed, life, and make it to be strong or not. This class can also load image for the bullet depends on the direction and make the bullet move in that direction.

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 __init__()

```
def bullet.Bullet.__init__ (
              self )
```

Constructor for Bullet.

Constructor initialize the bullet, loads the images for it in 4 different directions, and sets the default value for dir_x, dir_y, speed ,life, strong, image, and its position.

### 4.2.3 Member Function Documentation

#### 4.2.3.1 changeImage()

```
def bullet.Bullet.changeImage (
              self,
              dir_x,
              dir_y )
```

Change the image for the bullet depends on its direction.

**Parameters**

| | |
|---|---|
| *dir_x* | represents the horizontal direction of the bullet. |
| *dir_y* | represents the vertical direction of the bullet. |

#### 4.2.3.2 move()

```
def bullet.Bullet.move (
              self )
```

Move the bullet in its direction and make it disappear when it hits the boundaries of the map.

#### 4.2.3.3 setBulletLife()

```
def bullet.Bullet.setBulletLife (
              self,
              life )
```

Mutator for Bullet Life.

**Parameters**

| | |
|---|---|
| *life* | represents the life of the bullet. |

#### 4.2.3.4 setBulletSpeed()

```
def bullet.Bullet.setBulletSpeed (
              self,
              speed )
```

Mutator for Bullet Speed.

**Parameters**

| | |
|---|---|
| *speed* | represents the speed of the bullet. |

#### 4.2.3.5 setBulletStrong()

```
def bullet.Bullet.setBulletStrong (
              self,
              strong )
```

Mutator for Bullet Strong.

**Parameters**

| | |
|---|---|
| *strong* | represents that the bullet is strong or not. |

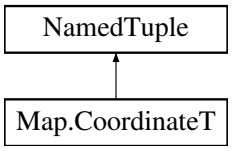
The documentation for this class was generated from the following file:

- src/bullet.py

## 4.3 Map.CoordinateT Class Reference

position coordination format.

Inheritance diagram for Map.CoordinateT:




### 4.3.1 Detailed Description

position coordination format.

This class includes position information and set up its format.

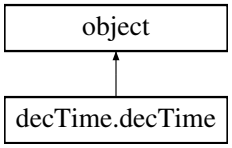
The documentation for this class was generated from the following file:

- src/Map.py

## 4.4 decTime.decTime Class Reference

declining time in the game.

Inheritance diagram for decTime.decTime:




### Public Member Functions

- def __init__ (self, total)
  *Constructor for decTime.*
- def subTime (self)
  *Make the time decline during the game.*

### Public Attributes

- **second**
- **hour**
- **minute**

### 4.4.1 Detailed Description

declining time in the game.

This class sets up the time duration of the game, it initialize the total time of the game, make it decline and return zero when the remaning time is zero.

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 __init__()

```
def decTime.decTime.__init__ (
              self,
              total )
```

Constructor for decTime.

Constructor initialize the decTime and turns the total seconds into the hours:minutes:seconds.

**Parameters**

| | |
|---|---|
| *totalTime* | represents the total time of the game in seconds. |

### 4.4.3 Member Function Documentation

#### 4.4.3.1 subTime()

```
def decTime.decTime.subTime (
              self )
```

Make the time decline during the game.

**Returns**

0 means end the game when time is over.

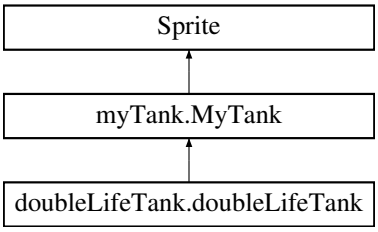
The documentation for this class was generated from the following file:

- src/decTime.py

## 4.5 doubleLifeTank.doubleLifeTank Class Reference

Double Life Tank.

Inheritance diagram for doubleLifeTank.doubleLifeTank:




### Public Member Functions

- def __init__ (self, coordinate)

  *Constructor for doubleLifeTank.*
- def bulletproof_start (self)

  *Change the value of bulletproof to True which means start the skill.*
- def bulletproof_end (self)

  *Change the value of bulletproof to False which means end the skill.*
- def getBulletProof (self)

  *Accessor for bulletProof.*

### Public Attributes

- **life**
- **bulletProof**
- **ID**
- **tank**
- **tank_R0**
- **tank_R1**
- **rect**

### 4.5.1 Detailed Description

Double Life Tank.

This class sets up double life tank's properties include operations for the tank. This class is able to initialize itself, start and end the skill bullet proof and return the value of bulletProof.

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 __init__()

```
def doubleLifeTank.doubleLifeTank.__init__ (
            self,
            coordinate )
```

Constructor for doubleLifeTank.

Constructor initialize the double life tank, load the images for it and set the default value for life, bulletProof, ID, and position.

**Parameters**

| | |
|---|---|
| *coordinate* | represents the position of the tank. |

Reimplemented from myTank.MyTank.

### 4.5.3 Member Function Documentation

#### 4.5.3.1 bulletproof_end()

```
def doubleLifeTank.doubleLifeTank.bulletproof_end (
              self )
```

Change the value of bulletproof to False which means end the skill.

#### 4.5.3.2 bulletproof_start()

```
def doubleLifeTank.doubleLifeTank.bulletproof_start (
              self )
```

Change the value of bulletproof to True which means start the skill.

#### 4.5.3.3 getBulletProof()

```
def doubleLifeTank.doubleLifeTank.getBulletProof (
              self )
```

Accessor for bulletProof.

**Returns**

bulletProof represents the condition of the skill bullet proof.

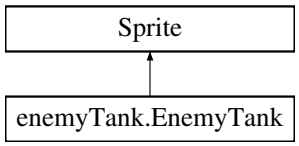
The documentation for this class was generated from the following file:

- src/doubleLifeTank.py

## 4.6 enemyTank.EnemyTank Class Reference

EnemeyTank.

Inheritance diagram for enemyTank.EnemyTank:




### Public Member Functions

- def __init__ (self, x=None, kind=None, isred=None, y=0)

  *Constructor for EnemyTank.*
- def shoot (self)

  *Make the enemy tank shoot, load the bullet image depending on the direction.*
- def move (self, tankGroup, brickGroup, ironGroup)

  *Make the enemy tank move and load the subsurfaces of the tank during moving.*

### Public Attributes

- **flash**
- **times**
- **kind**
- **enemy_x_0**
- **enemy_x_3**
- **enemy_3_0**
- **enemy_3_2**
- **isred**
- **tank**
- **x**
- **tank_R0**
- **tank_R1**
- **rect**
- **speed**
- **dir_y**
- **life**
- **bulletNotCooling**
- **bullet**
- **dirChange**
- **dir_x**

### 4.6.1 Detailed Description

EnemeyTank.

This class sets up enemy tank's properties include operations for the tank. This class is able to initialize itself and make the enemytank shoot and move.

### 4.6.2 Constructor & Destructor Documentation

#### 4.6.2.1 __init__()

```
def enemyTank.EnemyTank.__init__ (
              self,
              x = None,
              kind = None,
              isred = None,
              y = 0 )
```

Constructor for EnemyTank.

Constructor initialize the EnemyTank, sets the default value for falsh, times, kind and loads different images for different kinds of tanks. The constructor also sets the value for speed, direction, and the enemy tank's bullet.

### 4.6.3 Member Function Documentation

#### 4.6.3.1 move()

```
def enemyTank.EnemyTank.move (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

Make the enemy tank move and load the subsurfaces of the tank during moving.

**Parameters**

| | |
|---|---|
| *tankGroup* | The sprite group in pygame for the tanks. |
| *brickGroup* | The sprite group in pygame for the brick walls. |
| *ironGroup* | The sprite group in pygame for the iron walls. |

#### 4.6.3.2 shoot()

```
def enemyTank.EnemyTank.shoot (
              self )
```

Make the enemy tank shoot, load the bullet image depending on the direction.

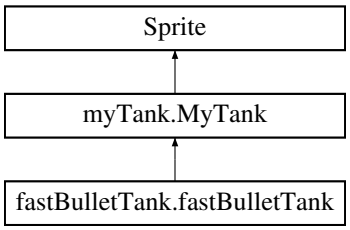
The documentation for this class was generated from the following file:

- src/enemyTank.py

## 4.7 fastBulletTank.fastBulletTank Class Reference

Fast Bullet Tank,.

Inheritance diagram for fastBulletTank.fastBulletTank:




### Public Member Functions

- def __init__ (self, coordinate)

  *Constructor for fastBulletTank.*
- def doubleBullet (self)

  *Activate the second bullte for the tank.*

### Public Attributes

- **bullet2**
- **ID**
- **tank**
- **tank_R0**
- **tank_R1**
- **rect**
- **dir_x**
- **dir_y**
- **level**

### 4.7.1 Detailed Description

Fast Bullet Tank,.

This class sets up fast bullet tank's properties include operations for the tank. This class is able to initialize itself and create the second bullet for the tank

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 __init__()

```
def fastBulletTank.fastBulletTank.__init__ (
            self,
            coordinate )
```

Constructor for fastBulletTank.

Constructor initialize the fast bullet tank, create the second bullet and set default value for the tank's ID, it also load the image for the fast bullet tank.

**Parameters**

| | |
|---|---|
| *coordinate* | represents the position of the tank. |

Reimplemented from myTank.MyTank.

### 4.7.3 Member Function Documentation

#### 4.7.3.1 doubleBullet()

```
def fastBulletTank.fastBulletTank.doubleBullet (
              self )
```

Activate the second bullte for the tank.

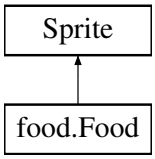
The documentation for this class was generated from the following file:

- src/fastBulletTank.py

## 4.8 food.Food Class Reference

The food which can help the players.

Inheritance diagram for food.Food:




### Public Member Functions

- def __init__ (self)
  *Constructor for food.*
- def change (self)
  *Change the food's life to True and set the position and image for it.*

### Public Attributes

- **food_boom**
- **food_clock**
- **food_gun**
- **food_iron**
- **food_star**
- **kind**
- **image**
- **rect**
- **life**

### 4.8.1 Detailed Description

The food which can help the players.

This class sets up different kinds of food and define the way of making the food appear.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 __init__()

```
def food.Food.__init__ (
              self )
```

Constructor for food.

Constructor initialize the food, load images for different kinds of food and set a random value for kind.

### 4.8.3 Member Function Documentation

#### 4.8.3.1 change()

```
def food.Food.change (
              self )
```

Change the food's life to True and set the position and image for it.

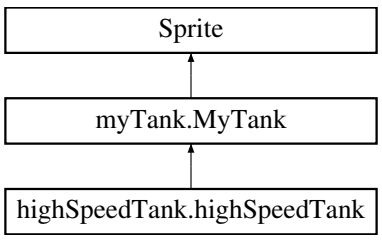
The documentation for this class was generated from the following file:

- src/food.py

## 4.9 highSpeedTank.highSpeedTank Class Reference

High Spped Tank.

Inheritance diagram for highSpeedTank.highSpeedTank:



### Public Member Functions

- def __init__ (self, coordinate)

  *Constructor for highSpeedTank.*
- def leap_start (self)

  *Activate the skill leap.*
- def leap_end (self)

  *End the skill leap.*

### Public Attributes

- **speed**
- **ID**
- **tank**
- **tank_R0**
- **tank_R1**
- **rect**

## 4.9.1 Detailed Description

High Spped Tank.

This class sets up high speed tank's properties include operations for the tank. This class is able to initialize itself, start and end the skill leap.

## 4.9.2 Constructor & Destructor Documentation

### 4.9.2.1 __init__()

```
def highSpeedTank.highSpeedTank.__init__ (
            self,
            coordinate )
```

Constructor for highSpeedTank.

Constructor initialize the high speed tank, load the images for for it and set the default value for speed, ID, and position.

**Parameters**

| | |
|---|---|
| *coordinate* | represents the position of the tank. |

Reimplemented from myTank.MyTank.

### 4.9.3 Member Function Documentation

#### 4.9.3.1 leap_end()

```
def highSpeedTank.highSpeedTank.leap_end (
              self )
```

End the skill leap.

#### 4.9.3.2 leap_start()

```
def highSpeedTank.highSpeedTank.leap_start (
              self )
```

Activate the skill leap.

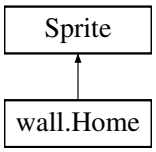
The documentation for this class was generated from the following file:

- src/highSpeedTank.py

## 4.10 wall.Home Class Reference

Home.

Inheritance diagram for wall.Home:




### Public Member Functions

- def __init__ (self, homeID=0)

  *Constructor for Home.*

### Public Attributes

- **homeID**
- **image**
- **rect**

### 4.10.1 Detailed Description

Home.

This class sets up the properties of the Home Base. This class constructs a iron wall and sets the image path so that the image can be loaded on the screen. The home ID is also identified in order to verify different home in PVP mode. The coordinate of the home base can be stored in the self.rect

### 4.10.2 Constructor & Destructor Documentation

#### 4.10.2.1 __init__()

```
def wall.Home.__init__ (
              self,
              homeID = 0 )
```

Constructor for Home.

Constructor initialize the home base, set the path for home Image and stored in the iron objects. Also a home ID is defined and saved in the home object. Moreover, The self.rect is defined and can be used to store the coordinate of the home base.

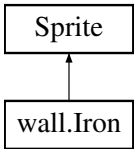
The documentation for this class was generated from the following file:

- src/wall.py

## 4.11 wall.Iron Class Reference

Iron.

Inheritance diagram for wall.Iron:




### Public Member Functions

- def __init__ (self)

  *Constructor for Iron.*

### Public Attributes

- **image**
- **rect**

### 4.11.1 Detailed Description

Iron.

This class sets up the properties of the iron wall. This class constructs a iron wall and sets the image path so that the image can be loaded on the screen. Also, the coordinate of the iron can be stored in the self.rect

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 __init__()

```
def wall.Iron.__init__ (
              self )
```

Constructor for Iron.

Constructor initialize the iron wall, set the path for iron Image and stored in the iron objects. Also a self.rect is defined and can be used to store the coordinate of the iron wall.

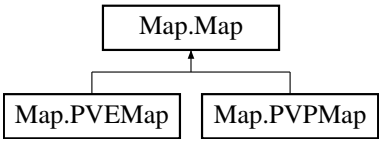
The documentation for this class was generated from the following file:

- src/wall.py

## 4.12 Map.Map Class Reference

Map.

Inheritance diagram for Map.Map:



### Public Member Functions

- def __init__ (self)

  *Constructor for Map.*
- def addBrick (self, coordinate)

  *add brick to the map object.*
- def addIron (self, coordinate)

  *add iron to the map object.*
- def addHome (self, coordinate, homeID=0)

  *add home base to the map object.*
- def loadBrickIron (self, path)

  *load the map.*
- def saveMap (self, path)

  *save the map.*

### Public Attributes

- **brickGroup**
- **ironGroup**
- **homeGroup**

## 4.12.1 Detailed Description

Map.

This class sets up the properties of the map objects. The map object contains brick walls, iron walls, and home bases. They can be saved in the corresponding group. After constructing the map object, new brick, iron and home base can be added into this map object. The properties of map can be loaded from a local text file, and the changed version can also be saved in the same format in a new local file.

## 4.12.2 Constructor & Destructor Documentation

### 4.12.2.1 __init__()

```
def Map.Map.__init__ (
              self )
```

Constructor for Map.

Constructor initialize the map by initializing the brick group, iron group, and home group.

Reimplemented in Map.PVPMap, and Map.PVEMap.

## 4.12.3 Member Function Documentation

#### 4.12.3.1 addBrick()

```
def Map.Map.addBrick (
              self,
              coordinate )
```

add brick to the map object.

Create a brick object using the input coordinate and add it into the brick group of the map object.

#### 4.12.3.2 addHome()

```
def Map.Map.addHome (
              self,
              coordinate,
              homeID = 0 )
```

add home base to the map object.

Create a home object using the input coordinate and home ID and add it into the home group of the map object.

#### 4.12.3.3 addIron()

```
def Map.Map.addIron (
              self,
              coordinate )
```

add iron to the map object.

Create a iron object using the input coordinate and add it into the iron group of the map object.

#### 4.12.3.4 loadBrickIron()

```
def Map.Map.loadBrickIron (
              self,
              path )
```

load the map.

Load the map based on the given file path and add all the brick and iron wall saved in that file into the map object.

#### 4.12.3.5 saveMap()

```
def Map.Map.saveMap (
              self,
              path )
```

save the map.

Save the map to the given file path and add all the brick and iron wall saved in the map object into the corresponding file.

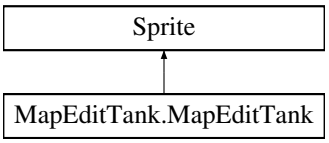
The documentation for this class was generated from the following file:

- src/Map.py

## 4.13 MapEditTank.MapEditTank Class Reference

MapEditTank.

Inheritance diagram for MapEditTank.MapEditTank:




### Public Member Functions

- def __init__ (self, coordinate)

  *Constructor for map editing tank.*
- def shoot (self)

  *shooting.*
- def moveUp (self, tankGroup, brickGroup, ironGroup)

  *move up.*
- def moveDown (self, tankGroup, brickGroup, ironGroup)

  *move down.*
- def moveLeft (self, tankGroup, brickGroup, ironGroup)

  *move left.*
- def moveRight (self, tankGroup, brickGroup, ironGroup)

  *move right.*

### Public Attributes

- **life**
- **level**
- **speed**
- **bullet**
- **bulletNotCooling**
- **tank**
- **tank_R0**
- **tank_R1**
- **rect**
- **dir_y**
- **dir_x**

### 4.13.1 Detailed Description

MapEditTank.

This class sets up the properties of the map edit tank objects. The tank speed, bullet, coordinates, and directions will be defined after creating a map editing tank. Functions like moving the tank and shooting can also be achieved in the map editing mode.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 __init__()

```
def MapEditTank.MapEditTank.__init__ (
              self,
              coordinate )
```

Constructor for map editing tank.

Constructor initialize the map editing tank by initializing the life, level, speed, bullet, coordinate, and direction of the tank.

### 4.13.3 Member Function Documentation

#### 4.13.3.1 moveDown()

```
def MapEditTank.MapEditTank.moveDown (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

move down.

Moveup function allows the tank to move down in the map editing mode without colliding with other sprites.

#### 4.13.3.2 moveLeft()

```
def MapEditTank.MapEditTank.moveLeft (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

move left.

Moveup function allows the tank to move left in the map editing mode without colliding with other sprites.

#### 4.13.3.3 moveRight()

```
def MapEditTank.MapEditTank.moveRight (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

move right.

Moveup function allows the tank to move right in the map editing mode without colliding with other sprites.

#### 4.13.3.4 moveUp()

```
def MapEditTank.MapEditTank.moveUp (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

move up.

Moveup function allows the tank to move up in the map editing mode without colliding with other sprites.

#### 4.13.3.5 shoot()

```
def MapEditTank.MapEditTank.shoot (
              self )
```

shooting.

Shooting function allows the map editing tank to shoot a bullet in the map editing mode and break the brick and iron walls.

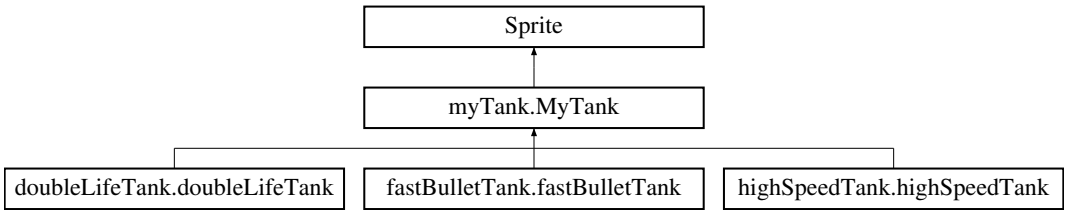
The documentation for this class was generated from the following file:

- src/MapEditTank.py

## 4.14 myTank.MyTank Class Reference

My Tank.

Inheritance diagram for myTank.MyTank:




### Public Member Functions

- def __init__ (self, coordinate)
  *Constructor for myTank.*
- def shoot (self)
  *Make the players' tank shoot, set the bullet life to True and load the bullet image depending on the direction.*
- def levelUp (self)
  *Increase the tank's level by 1 when it was less than 2.*
- def levelDown (self)
  *Decrease the tank's level by 1 when it was higher than 0 and set the property for the corresponding level.*
- def moveUp (self, tankGroup, brickGroup, ironGroup)
  *Move the tank upward.*
- def moveDown (self, tankGroup, brickGroup, ironGroup)
  *Move the tank downward.*
- def moveLeft (self, tankGroup, brickGroup, ironGroup)
  *Move the tank to the left.*
- def moveRight (self, tankGroup, brickGroup, ironGroup)
  *Move the tank to the right.*

### Public Attributes

- **life**
- **level**
- **speed**
- **bullet**
- **bulletNotCooling**
- **tank**
- **tank_R0**
- **tank_R1**
- **rect**
- **dir_y**
- **dir_x**

## 4.14.1 Detailed Description

My Tank.

This class sets up player's tank's properties include operations for the tank. This class is able to initialize itself, make the tank shoot, move, and change the level of the tank.

## 4.14.2 Constructor & Destructor Documentation

### 4.14.2.1 __init__()

```
def myTank.MyTank.__init__ (
            self,
            coordinate )
```

Constructor for myTank.

Constructor initialize the player's tank, load the images for it and set the default value for life, level, speed, position and direction.

**Parameters**

| | |
|---|---|
| *coordinate* | represents the position of the tank. |

Reimplemented in fastBulletTank.fastBulletTank, doubleLifeTank.doubleLifeTank, and highSpeedTank.highSpeedTank.

## 4.14.3 Member Function Documentation

#### 4.14.3.1 levelDown()

```
def myTank.MyTank.levelDown (
              self )
```

Decrease the tank's level by 1 when it was higher than 0 and set the property for the corresponding level.

#### 4.14.3.2 levelUp()

```
def myTank.MyTank.levelUp (
              self )
```

Increase the tank's level by 1 when it was less than 2.

#### 4.14.3.3 moveDown()

```
def myTank.MyTank.moveDown (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

Move the tank downward.

**Parameters**

| | |
|---|---|
| *tankGroup* | The sprite group in pygame for tanks. |
| *brickGroup* | The sprite group in pygame for brick walls. |
| *ironGroup* | The sprite group in pygame for iron walls. |

**Returns**

True means the tank is blocked by the walls or boundaries.
False means the tank can move.

#### 4.14.3.4 moveLeft()

```
def myTank.MyTank.moveLeft (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

Move the tank to the left.

**Parameters**

| | |
|---|---|
| *tankGroup* | The sprite group in pygame for tanks. |
| *brickGroup* | The sprite group in pygame for brick walls. |
| *ironGroup* | The sprite group in pygame for iron walls. |

**Returns**

True means the tank is blocked by the walls or boundaries.
False means the tank can move.

#### 4.14.3.5 moveRight()

```
def myTank.MyTank.moveRight (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

Move the tank to the right.

**Parameters**

| | |
|---|---|
| *tankGroup* | The sprite group in pygame for tanks. |
| *brickGroup* | The sprite group in pygame for brick walls. |
| *ironGroup* | The sprite group in pygame for iron walls. |

**Returns**

True means the tank is blocked by the walls or boundaries.
False means the tank can move.

#### 4.14.3.6 moveUp()

```
def myTank.MyTank.moveUp (
              self,
              tankGroup,
              brickGroup,
              ironGroup )
```

Move the tank upward.

**Parameters**

| | |
|---|---|
| *tankGroup* | The sprite group in pygame for tanks. |
| *brickGroup* | The sprite group in pygame for brick walls. |
| *ironGroup* | The sprite group in pygame for iron walls. |

**Returns**

True means the tank is blocked by the walls or boundaries.
False means the tank can move.

#### 4.14.3.7 shoot()

```
def myTank.MyTank.shoot (
              self )
```

Make the players' tank shoot, set the bullet life to True and load the bullet image depending on the direction.

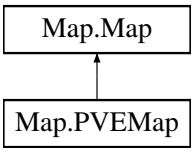
The documentation for this class was generated from the following file:

- src/myTank.py

## 4.15 Map.PVEMap Class Reference

PVE Map.

Inheritance diagram for Map.PVEMap:




### Public Member Functions

- def __init__ (self)

  *Constructor for PVE map.*
- def loadPVEMap (self, path)

  *load PVE map.*

### Additional Inherited Members

### 4.15.1 Detailed Description

PVE Map.

PVEMap is a subclass of Map class. PVEMap can directly use the constructor of Map class and read from the local file to add brick and irons into the corresponding group. The corrdinate for PVE home base can also be defined and add into the PVEMap object.

### 4.15.2 Constructor & Destructor Documentation

#### 4.15.2.1 __init__()

```
def Map.PVEMap.__init__ (
              self )
```

Constructor for PVE map.

Constructor initialize the PVE map by initializing the brick group, iron group, and home group.

Reimplemented from Map.Map.

### 4.15.3 Member Function Documentation

#### 4.15.3.1 loadPVEMap()

```
def Map.PVEMap.loadPVEMap (
              self,
              path )
```

load PVE map.

Load the PVE map from the give path, read the saved brick and iron walls and add into the corresponding group. Initialize a home base and add it into the home group.

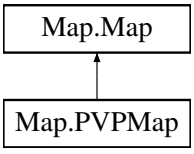
The documentation for this class was generated from the following file:

- src/Map.py

## 4.16 Map.PVPMap Class Reference

PVP Map.

Inheritance diagram for Map.PVPMap:



### Public Member Functions

- def __init__ (self)

  *Constructor for PVP map.*
- def loadPVPMap (self, path)

  *load PVP map.*

### Additional Inherited Members

## 4.16.1 Detailed Description

PVP Map.

PVPMap is a subclass of Map class. PVPMap can directly use the constructor of Map class and read from the local file to add brick and irons into the corresponding group. The corrdinate for PVP home bases can also be defined and add into the PVPMap object.

## 4.16.2 Constructor & Destructor Documentation

#### 4.16.2.1 __init__()

```
def Map.PVPMap.__init__ (
              self )
```

Constructor for PVP map.

Constructor initialize the PVP map by initializing the brick group, iron group, and home group.

Reimplemented from Map.Map.

## 4.16.3 Member Function Documentation

#### 4.16.3.1 loadPVPMap()

```
def Map.PVPMap.loadPVPMap (
              self,
              path )
```

load PVP map.

Load the PVP map from the give path, read the saved brick and iron walls and add into the corresponding group. Initialize two home bases and add them into the home group.

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

- src/Map.py

# Chapter 5

# File Documentation

## 5.1 src/bullet.py File Reference

Bullet Class Module.

### Classes

- class bullet.Bullet
  *Bullet.*

### Variables

- int bullet.EDGE_LENGTH = 3
  *bullet Constant*
- int **bullet.GRID_LENGTH** = 24
- int **bullet.BOUNDARY_SIZE** = 630

### 5.1.1 Detailed Description

Bullet Class Module.

Enemy Tank Module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/24/2020

### 5.1.2 Variable Documentation

#### 5.1.2.1 EDGE_LENGTH

```
int bullet.EDGE_LENGTH = 3
```

bullet Constant

EDGE_LENGTH represents the length of the edge in pixel. GRID_LENGTH represents the length of one grid of the map in pixel. BOUNDARY_SIZE represents the size of the boundary of the map

## 5.2 src/decTime.py File Reference

declining time Module

### Classes

- class decTime.decTime

  *declining time in the game.*

### 5.2.1 Detailed Description

declining time Module

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

## 5.3 src/display.py File Reference

Functions that can show the change during the game.

## Functions

- def display.checkCollideME (bullet, bgMap)

  *checkCollideME.*
- def display.checkCollidePVP (tank_born_coordinate, bullet, bgMap, myTank, homeDead, deadCount2)

  *checkCollidePVP.*
- def display.checkCollidePVE (enemyBulletGroup, bullet, redEnemyGroup, greenEnemyGroup, otherEnemy↩ Group, bgMap, prop, enemyNumber, homeSurvive)

  *checkCollidePVE.*
- def display.drawBG (background_image, screen)

  *drawBG.*
- def display.drawBrick (bgMap, screen)

  *drawBrick.*
- def display.drawIron (bgMap, screen)

  *drawIron.*
- def display.drawHome (bgMap, screen)

  *drawHome.*
- def display.drawTank_1 (deadCount1, switch_R1_R2_image, running_T1, delay, myTank_T1, screen)

  *drawTank_1.*
- def display.drawTank_2 (deadCount2, switch_R1_R2_image, running_T2, myTank_T2, screen)

  *drawTank_2.*
- def display.drawEnemyTank (switch_R1_R2_image, enemyCouldMove, bgMap, allEnemyGroup, screen, all↩ TankGroup, appearance)

  *drawEnemyTank.*
- def display.drawMyBullet (deadCount, homeSurvive, enemyNumber, prop, bgMap, myTank, screen, enemy↩ BulletGroup, redEnemyGroup, greenEnemyGroup, otherEnemyGroup)

  *drawMyBullet.*
- def display.drawEnemyBullet (homeSurvive, mytankGroup, deadCount1, deadCount2, enemyCouldMove, moving, bgMap, allEnemyGroup, enemyBulletGroup, screen, myTank_T1, myTank_T2)

  *drawEnemyBullet.*
- def display.drawFood (enemyNumber, enemyCouldMove, prop, bgMap, screen, allEnemyGroup, myTank)

  *drawFood.*
- def display.drawMyBulletPVP (tank_born_coordinate, deadCount1, deadCount2, myTank_T2, homeDead, bgMap, myTank_T1, screen)

  *drawMyBulletPVP.*
- def display.drawBulletME (bgMap, myTank_T1, screen)

  *drawBulletME.*
- def display.drawPVE (homeSurvive, mytankGroup, deadCount1, deadCount2, switch_R1_R2_image, enemyCouldMove, enemyNumber, prop, moving, running_T1, running_T2, bgMap, background_image, screen, delay, myTank_T1, myTank_T2, allEnemyGroup, allTankGroup, appearance, enemyBulletGroup, redEnemyGroup, greenEnemyGroup, otherEnemyGroup)

  *drawPVE.*
- def display.drawPVP (homeDead, mytankGroup, deadCount1, deadCount2, switch_R1_R2_image, moving, running_T1, running_T2, bgMap, background_image, screen, delay, myTank_T1, myTank_T2, allTankGroup)

  *drawPVP.*
- def display.drawME (bgMap, deadCount1, mytankGroup, switch_R1_R2_image, moving, running_T1, background_image, screen, delay, myTank_T1, allTankGroup)

  *drawME.*

### Variables

- int display.EDGE_LENGTH = 3

  *display Constant*
- int **display.GRID_LENGTH** = 24
- **display.TANK1_BORN** = CoordinateT(8, 24)
- **display.TANK2_BORN** = CoordinateT(16, 24)
- **display.BULLET_COORDINATE** = CoordinateT(12, 24)
- **display.bang_sound** = pygame.mixer.Sound("../music/bang.wav")

## 5.3.1 Detailed Description

Functions that can show the change during the game.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/25/2020

## 5.3.2 Function Documentation

### 5.3.2.1 checkCollideME()

```
def display.checkCollideME (
              bullet,
              bgMap )
```

checkCollideME.

checkCollideME function will check whether the players' bullet collide with brick or iron wall in map editing mode.

### 5.3.2.2 checkCollidePVE()

```
def display.checkCollidePVE (
              enemyBulletGroup,
              bullet,
              redEnemyGroup,
              greenEnemyGroup,
              otherEnemyGroup,
              bgMap,
              prop,
              enemyNumber,
              homeSurvive )
```

checkCollidePVE.

checkCollidePVE function will check whether the players' bullets collide with brick wall, iron wall, enemies' tanks, and home base in the PVP mode.

#### 5.3.2.3 checkCollidePVP()

```
def display.checkCollidePVP (
              tank_born_coordinate,
              bullet,
              bgMap,
              myTank,
              homeDead,
              deadCount2 )
```

checkCollidePVP.

checkCollidePVP function will check whether the players' bullets collide with brick wall, iron wall, the other player's tank, and other player's home base in the PVP mode.

#### 5.3.2.4 drawBG()

```
def display.drawBG (
              background_image,
              screen )
```

drawBG.

drawBG function draw the background on screen.

#### 5.3.2.5 drawBrick()

```
def display.drawBrick (
              bgMap,
              screen )
```

drawBrick.

drawBrick function draw all the brick walls on screen.

#### 5.3.2.6 drawBulletME()

```
def display.drawBulletME (
              bgMap,
              myTank_T1,
              screen )
```

drawBulletME.

drawBulletME function draw the players' bullets on screen in map editing mode. It will also check the collide between the bullet and other sprites by calling the checkCollideME function.

#### 5.3.2.7 drawEnemyBullet()

```
def display.drawEnemyBullet (
              homeSurvive,
              mytankGroup,
              deadCount1,
              deadCount2,
              enemyCouldMove,
              moving,
              bgMap,
              allEnemyGroup,
              enemyBulletGroup,
              screen,
              myTank_T1,
              myTank_T2 )
```

drawEnemyBullet.

drawEnemyBullet function draw the enemies' bullets on screen in PVE mode. It will also check the collide between the bullet and other sprites like players' tanks, brick wall, iron wall, and home base.

#### 5.3.2.8 drawEnemyTank()

```
def display.drawEnemyTank (
              switch_R1_R2_image,
              enemyCouldMove,
              bgMap,
              allEnemyGroup,
              screen,
              allTankGroup,
              appearance )
```

drawEnemyTank.

drawEnemyTank function draw the enemies' tanks on screen. The tank image will change based on switch_R1_↩ R2_image in order to show the dynamic wheel.

#### 5.3.2.9 drawFood()

```
def display.drawFood (
              enemyNumber,
              enemyCouldMove,
              prop,
              bgMap,
              screen,
              allEnemyGroup,
              myTank )
```

drawFood.

drawFood function draws the food on screen in PVE mode. It will also check the collide between the bullet and other sprites. It also defines the result after eating the corresponding food.

#### 5.3.2.10 drawHome()

```
def display.drawHome (
              bgMap,
              screen )
```

drawHome.

drawHome function draw all the home bases on screen.

#### 5.3.2.11 drawIron()

```
def display.drawIron (
              bgMap,
              screen )
```

drawIron.

drawIron function draw all the iron walls on screen.

#### 5.3.2.12 drawME()

```
def display.drawME (
              bgMap,
              deadCount1,
              mytankGroup,
              switch_R1_R2_image,
              moving,
              running_T1,
              background_image,
              screen,
              delay,
              myTank_T1,
              allTankGroup )
```

drawME.

drawME function draw all the necessary elements in map editing mode by calling drawBG, drawBrick, drawIron, drawHome, drawTank_1, drawBulletME functions.

#### 5.3.2.13 drawMyBullet()

```
def display.drawMyBullet (
              deadCount,
              homeSurvive,
              enemyNumber,
              prop,
              bgMap,
              myTank,
              screen,
              enemyBulletGroup,
              redEnemyGroup,
              greenEnemyGroup,
              otherEnemyGroup )
```

drawMyBullet.

drawMyBullet function draw the players' bullets on screen in PVE mode. It will also check the collide between the bullet and other sprites by calling the checkCollidePVE function.

#### 5.3.2.14 drawMyBulletPVP()

```
def display.drawMyBulletPVP (
              tank_born_coordinate,
              deadCount1,
              deadCount2,
              myTank_T2,
              homeDead,
              bgMap,
              myTank_T1,
              screen )
```

drawMyBulletPVP.

drawMyBulletPVP function draw the players' bullets on screen in PVP mode. It will also check the collide between the bullet and other sprites by calling the checkCollidePVP function.

#### 5.3.2.15 drawPVE()

```
def display.drawPVE (
              homeSurvive,
              mytankGroup,
              deadCount1,
              deadCount2,
              switch_R1_R2_image,
              enemyCouldMove,
              enemyNumber,
              prop,
              moving,
              running_T1,
              running_T2,
              bgMap,
              background_image,
              screen,
              delay,
              myTank_T1,
              myTank_T2,
              allEnemyGroup,
              allTankGroup,
              appearance,
              enemyBulletGroup,
              redEnemyGroup,
              greenEnemyGroup,
              otherEnemyGroup )
```

drawPVE.

drawPVE function draw all the necessary elements in PVE mode by calling drawBG, drawBrick, drawIron, draw↩ Home, drawTank_1, drawTank_2, drawEnemyTank, drawMyBullet, drawEnemyBullet, drawFood functions.

#### 5.3.2.16 drawPVP()

```
def display.drawPVP (
              homeDead,
              mytankGroup,
              deadCount1,
              deadCount2,
              switch_R1_R2_image,
              moving,
              running_T1,
              running_T2,
              bgMap,
              background_image,
              screen,
              delay,
              myTank_T1,
              myTank_T2,
              allTankGroup )
```

drawPVP.

drawPVP function draw all the necessary elements in PVP mode by calling drawBG, drawBrick, drawIron, drawHome, drawTank_1, drawTank_2, drawMyBulletPVP functions.

#### 5.3.2.17 drawTank_1()

```
def display.drawTank_1 (
              deadCount1,
              switch_R1_R2_image,
              running_T1,
              delay,
              myTank_T1,
              screen )
```

drawTank_1.

drawTank_1 function draw the player1's tank on screen. The tank image will change based on switch_R1_R2_image in order to show the dynamic wheel.

#### 5.3.2.18 drawTank_2()

```
def display.drawTank_2 (
              deadCount2,
              switch_R1_R2_image,
              running_T2,
              myTank_T2,
              screen )
```

drawTank_2.

drawTank_2 function draw the player2's tank on screen. The tank image will change based on switch_R1_R2_image in order to show the dynamic wheel.

### 5.3.3 Variable Documentation

#### 5.3.3.1 EDGE_LENGTH

```
int display.EDGE_LENGTH = 3
```

display Constant

EDGE_LENGTH represents the length of the edge in pixel. GRID_LENGTH represents the length of one grid of the map in pixel. TANK1_BORN represents the location that player1's tank borns at. TANK2_BORN represents the location that player2's tank borns at. BULLET_COORDINATE represents the coordinate of the bullet.

## 5.4 src/doubleLifeTank.py File Reference

Double Life Tank module.

### Classes

- class doubleLifeTank.doubleLifeTank

  *Double Life Tank.*

### Variables

- string **doubleLifeTank.imagePath** = "../image/DoubleLifeTank.png"

### 5.4.1 Detailed Description

Double Life Tank module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

## 5.5 src/fastBulletTank.py File Reference

Fast Bullet Tank module.

### Classes

- class fastBulletTank.fastBulletTank

  *Fast Bullet Tank,.*

### Variables

- string **fastBulletTank.imagePath** = "../image/FastBulletTank.png"

#### 5.5.1 Detailed Description

Fast Bullet Tank module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/24/2020

## 5.6 src/food.py File Reference

Food module.

### Classes

- class food.Food

  *The food which can help the players.*

#### 5.6.1 Detailed Description

Food module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/24/2020

## 5.7 src/highSpeedTank.py File Reference

High Speed Tank module.

### Classes

- class highSpeedTank.highSpeedTank

  *High Spped Tank.*

### Variables

- string **highSpeedTank.imagePath** = "../image/HighSpeedTank.png"

#### 5.7.1 Detailed Description

High Speed Tank module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/24/2020

## 5.8 src/main.py File Reference

Main Module.

### Variables

- **main.mode** = Menue()

#### 5.8.1 Detailed Description

Main Module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

## 5.9 src/Map.py File Reference

Map class, PVEMap class, PVPMap class.

### Classes

- class Map.Map
  *Map.*
- class Map.PVEMap
  *PVE Map.*
- class Map.PVPMap
  *PVP Map.*
- class Map.CoordinateT
  *position coordination format.*

### Variables

- int Map.EDGE_LENGTH = 3
  *Map Constant.*
- int **Map.GRID_LENGTH** = 24

#### 5.9.1 Detailed Description

Map class, PVEMap class, PVPMap class.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

#### 5.9.2 Variable Documentation

##### 5.9.2.1 EDGE_LENGTH

```
int Map.EDGE_LENGTH = 3
```

Map Constant.

EDGE_LENGTH represents the length of the edge in pixel. GRID_LENGTH represents the length of one grid of the map in pixel.

## 5.10 src/MapEditTank.py File Reference

MapEditTank class module.

### Classes

- class MapEditTank.MapEditTank
  *MapEditTank.*

### Variables

- string **MapEditTank.imagePath** = "../image/HighSpeedTank.png"

#### 5.10.1 Detailed Description

MapEditTank class module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

## 5.11 src/myTank.py File Reference

My Tank module.

### Classes

- class myTank.MyTank
  *My Tank.*

### Variables

- string **myTank.imagePath** = "../image/HighSpeedTank.png"

#### 5.11.1 Detailed Description

My Tank module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/24/2020

## 5.12 src/myTankControl.py File Reference

My Tank Control Module.

### Functions

- def myTankControl.operatePlayer1 (fire_sound, key_pressed, moving, movdir1, myTank, allTankGroup, brickGroup, ironGroup, running_T1)

  *Implement player1's operation.*
- def myTankControl.operatePlayer2 (fire_sound, key_pressed, moving2, movdir2, myTank, allTankGroup, brickGroup, ironGroup, running_T2)

  *Implement player2's operation.*
- def myTankControl.operatePlayerME (key_pressed, moving, movdir1, myTank, allTankGroup, bgMap, running_T1)

  *Implement player operation in Map Editing mode.*

### 5.12.1 Detailed Description

My Tank Control Module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/24/2020

### 5.12.2 Function Documentation

#### 5.12.2.1 operatePlayer1()

```
def myTankControl.operatePlayer1 (
              fire_sound,
              key_pressed,
              moving,
              movdir1,
              myTank,
              allTankGroup,
              brickGroup,
              ironGroup,
              running_T1 )
```

Implement player1's operation.

Read player1's key pressed and do the corresponding moving, @or activation of the ultimate skill.

**Parameters**

| | |
|---|---|
| *fire_sound* | represents a audio file for the sound of shooting. |
| *key_pressed* | represents that if the player is pressing a key. |
| *moving* | represents the moving condition of the tank. |
| *movdir1* | represents the direction of player's tank. |
| *myTank* | represents the player's tank. |
| *allTankGroup* | represents a sprite group in pygame for all the tank. |
| *brickGroup* | represents a sprite group in pygame for brick walls. |
| *ironGroup* | represents a sprite group in pygame for iron walls. |
| *running_T1* | represents that player's tank is moving or not. |

**Returns**

key_pressd represents that if the player is pressing a key.
moving represents the moving condition of the tank.
movdir1 represent the direction of of the tank.
myTank represents the player's tank.
allTankGroup represents a sprite group in pygame for all the tank.
brickGroup represents a sprite group in pygame for brick walls.
ironGroup represents a sprite group in pygame for iron walls.
running_T1 represents that player's tank is moving or not.

#### 5.12.2.2 operatePlayer2()

```
def myTankControl.operatePlayer2 (
              fire_sound,
              key_pressed,
              moving2,
              movdir2,
              myTank,
              allTankGroup,
              brickGroup,
              ironGroup,
              running_T2 )
```

Implement player2's operation.

Read player2's key pressed and do the corresponding moving, @shooting or activatation of the ultimate skill.

**Parameters**

| | |
|---|---|
| *fire_sound* | represents a audio file for the sound of shooting. |
| *key_pressed* | represents that if the player is pressing a key. |
| *moving2* | represents the moving condition of the tank. |
| *movdir2* | represents the direction of player's tank. |
| *myTank* | represents the player's tank. |
| *allTankGroup* | represents a sprite group in pygame for all the tank. |
| *brickGroup* | represents a sprite group in pygame for brick walls. |
| *ironGroup* | represents a sprite group in pygame for iron walls. |
| *running_T2* | represents that player's tank is moving or not. |

**Returns**

key_pressd represents that if the player is pressing a key.
moving2 represents the moving condition of the tank.
movdir2 represent the direction of of the tank.
myTank represents the player's tank.
allTankGroup represents a sprite group in pygame for all the tank.
brickGroup represents a sprite group in pygame for brick walls.
ironGroup represents a sprite group in pygame for iron walls.
running_T2 represents that player's tank is moving or not.

#### 5.12.2.3 operatePlayerME()

```
def myTankControl.operatePlayerME (
              key_pressed,
              moving,
              movdir1,
              myTank,
              allTankGroup,
              bgMap,
              running_T1 )
```

Implement player operation in Map Editing mode.

Read player's key pressed and do the corresponding moving, shooting, adding brick, adding iron.

**Parameters**

| | |
|---|---|
| *key_pressed* | represents that if the player is pressing a key. |
| *moving* | represents the moving condition of the tank. |
| *movdir1* | represents the direction of player's tank. |
| *myTank* | represents the player's tank. |
| *allTankGroup* | represents a sprite group in pygame for all the tank. |
| *brickGroup* | represents a sprite group in pygame for brick walls. |
| *ironGroup* | represents a sprite group in pygame for iron walls. |
| *running_T1* | represents that player's tank is moving or not. |

**Returns**

key_pressd represents that if the player is pressing a key.
moving represents the moving condition of the tank.
movdir1 represent the direction of of the tank.
myTank represents the player's tank.
allTankGroup represents a sprite group in pygame for all the tank.
brickGroup represents a sprite group in pygame for brick walls.
ironGroup represents a sprite group in pygame for iron walls.
running_T1 represents that player's tank is moving or not.

## 5.13 src/PvsE.py File Reference

PvsE Module.

### Functions

- def PvsE.PvsE ()
  *Starts the PvsE mode.*

### 5.13.1 Detailed Description

PvsE Module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

### 5.13.2 Function Documentation

#### 5.13.2.1 PvsE()

```
def PvsE.PvsE ( )
```

Starts the PvsE mode.

Starts the PvsE mode. Two players can control their tanks to have a battle with enermy tnaks.

## 5.14 src/PvsP.py File Reference

PvsP Module.

### Functions

- def PvsP.PvsP ()
  *Starts the PvsP mode.*

### 5.14.1 Detailed Description

PvsP Module.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

### 5.14.2 Function Documentation

#### 5.14.2.1 PvsP()

```
def PvsP.PvsP ( )
```

Starts the PvsP mode.

Starts the PvsP mode. Two players can control their tanks to have a battle against each other.

## 5.15 src/Screen.py File Reference

Functions that can be used to show a particular process.

### Functions

- def Screen.StartGame ()
  *StartGame screen.*
- def Screen.Menue ()
  *Menue screen.*
- def Screen.chooseTankScreen (coordinate_T1, coordinate_T2)
  *chooseTankScreen screen.*
- def Screen.loadingMapScreen (File)
  *loadingMapScreen screen.*
- def Screen.operationInstructPlay ()
  *Operation Instruction Screen for PVP and PVE.*
- def Screen.operationInstructMap ()
  *Operation Instruction Screen for map editing.*
- def Screen.ruleScreen (rule)
  *loadingMapScreen screen.*
- def Screen.endScreen_PVE (result)
  *endScreen_PVE screen.*
- def Screen.endScreen_PVP (result)
  *endScreen_PVP screen.*
- def Screen.saveScreenME (File)
  *saveScreenME screen.*
- def Screen.chooseMapME ()
  *chooseMapME screen.*

### 5.15.1 Detailed Description

Functions that can be used to show a particular process.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/25/2020

### 5.15.2 Function Documentation

#### 5.15.2.1 chooseMapME()

```
def Screen.chooseMapME ( )
```

chooseMapME screen.

chooseMapME function will show the selection of map types that are available for editing, like PVP maps and PVE maps. Users can select one of them by pressing the corresponding key and press return to continue.

#### 5.15.2.2 chooseTankScreen()

```
def Screen.chooseTankScreen (
              coordinate_T1,
              coordinate_T2 )
```

chooseTankScreen screen.

chooseTankScreen function will show the selection of tanks, including double-life tank, fast-bullet tank, and high-speed tank. Users can select one of them by pressing the corresponding key and press return to continue.

#### 5.15.2.3 endScreen_PVE()

```
def Screen.endScreen_PVE (
              result )
```

endScreen_PVE screen.

endScreen_PVE function will show the result screen of PVE game based on the input result. PVE has two possible results: winning and losing. The endScreen_PVE will show the corresponding result screen and show the result directly to the players.

#### 5.15.2.4 endScreen_PVP()

```
def Screen.endScreen_PVP (
              result )
```

endScreen_PVP screen.

endScreen_PVP function will show the result screen of PVP game based on the input result. PVP has three possible results: winning, losing, and draw. The endScreen_PVP will show the corresponding result screen and show the result directly to the players.

#### 5.15.2.5 loadingMapScreen()

```
def Screen.loadingMapScreen (
              File )
```

loadingMapScreen screen.

loadingMapScreen function will show the selection of maps. Users can select one of them by pressing the corresponding key and press return to continue.

#### 5.15.2.6 Menue()

```
def Screen.Menue ( )
```

Menue screen.

Menue function will show the selection of gaming modes, including PVE, PVP, and Map Editing. users can select one of them by pressing the corresponding key and press return to continue.

#### 5.15.2.7 operationInstructMap()

```
def Screen.operationInstructMap ( )
```

Operation Instruction Screen for map editing.

Display operation instruction for users to show how to control their tank for map editing.

#### 5.15.2.8 operationInstructPlay()

```
def Screen.operationInstructPlay ( )
```

Operation Instruction Screen for PVP and PVE.

Display operation instruction for users to show how to control their tank for PVP and PVE.

#### 5.15.2.9 ruleScreen()

```
def Screen.ruleScreen (
              rule )
```

loadingMapScreen screen.

loadingMapScreen function will show the selection of maps. Users can select one of them by pressing the corresponding key and press return to continue.

#### 5.15.2.10 saveScreenME()

```
def Screen.saveScreenME (
              File )
```

saveScreenME screen.

saveScreenME function will show the selection of map names that are available for saving. Users can select one of them by pressing the corresponding key and press return to continue.

#### 5.15.2.11 StartGame()

```
def Screen.StartGame ( )
```

StartGame screen.

StartGame function will show a screen of game starting process. In this gaming procedure, users will be able to know the game starts and return can be pressed to continue the game.

## 5.16 src/wall.py File Reference

Brick class, Iron class, and Home class.

### Classes

- class wall.Brick
  *Brick*.
- class wall.Iron
  *Iron*.
- class wall.Home
  *Home*.

### Variables

- string **wall.brickImage** = "../image/brick.png"
- string **wall.ironImage** = "../image/iron.png"
- string **wall.homeImage** = "../image/home.png"

### 5.16.1 Detailed Description

Brick class, Iron class, and Home class.

**Author**

Xinyu Huang, Di Wu, Jiahao Zhou

**Date**

03/23/2020

# Index