

# Disease\_Simulator

1.0

Generated by Doxygen 1.8.20



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Actor.Actor Class Reference	5
3.1.1 Constructor & Destructor Documentation	6
3.1.1.1 __init__()	6
3.1.2 Member Function Documentation	6
3.1.2.1 act()	6
3.1.2.2 addedToWorld()	6
3.1.2.3 getID()	7
3.1.2.4 getWorld()	7
3.1.2.5 getX()	7
3.1.2.6 getY()	8
3.1.2.7 Iteration()	8
3.1.2.8 setLocation()	8
3.2 Disease.Disease Class Reference	9
3.2.1 Constructor & Destructor Documentation	9
3.2.1.1 __init__()	9
3.2.2 Member Function Documentation	10
3.2.2.1 act()	10
3.2.2.2 getGrowthCondition()	10
3.2.2.3 getQuadrant()	10
3.2.2.4 getStrength()	11
3.2.2.5 setGrowthCondition()	11
3.3 IDisease.IDisease Class Reference	11
3.3.1 Member Function Documentation	12
3.3.1.1 setGrowthCondition()	12
3.4 IWorld.IWorld Class Reference	12
3.5 mapper.mapper Class Reference	13
3.5.1 Constructor & Destructor Documentation	13
3.5.1.1 __init__()	13
3.5.2 Member Function Documentation	14
3.5.2.1 viewportToWindow()	14
3.5.2.2 windowToViewport()	14
3.5.2.3 windowVecToViewport()	15
3.6 MyWorld.MyWorld Class Reference	15
3.6.1 Constructor & Destructor Documentation	16
3.6.1.1 __init__()	16
3.6.2 Member Function Documentation	16

3.6.2.1 act()	16
3.6.2.2 getTemp()	16
3.6.2.3 initDiseases()	17
3.6.2.4 initGrowthConditions()	17
3.6.2.5 initLocations()	18
3.6.2.6 initTemps()	18
3.6.2.7 prepare()	18
3.6.2.8 setTemp()	19
3.7 SimulationPanel.SimulationPanel Class Reference	19
3.8 Timer.Timer Class Reference	20
3.9 World.World Class Reference	20
<b>Index</b>	<b>21</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

Actor.Actor . . . . .	5
Disease.Disease . . . . .	9
mapper.mapper . . . . .	13
SimulationPanel.SimulationPanel . . . . .	19
Timer.Timer . . . . .	20
World.World . . . . .	20
MyWorld.MyWorld . . . . .	15
ABC	
IDisease.IDisease . . . . .	11
Disease.Disease . . . . .	9
IWorld.IWorld . . . . .	12
MyWorld.MyWorld . . . . .	15



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Actor.Actor</a>	5
<a href="#">Disease.Disease</a>	9
<a href="#">IDisease.IDisease</a>	11
<a href="#">IWorld.IWorld</a>	12
<a href="#">mapper.mapper</a>	13
<a href="#">MyWorld.MyWorld</a>	15
<a href="#">SimulationPanel.SimulationPanel</a>	19
<a href="#">Timer.Timer</a>	20
<a href="#">World.World</a>	20



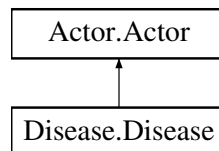


## Chapter 3

# Class Documentation

### 3.1 Actor.Actor Class Reference

Inheritance diagram for Actor.Actor:



#### Public Member Functions

- `def \_\_init\_\_ (self)`  
*Construct a new [Actor](#) object.*
- `def getID (self)`  
*Used for testing.*
- `def Iteration (self)`  
*Used for testing.*
- `def act (self)`  
*Prints on screen in the format "Iteration <ID>: Actor <Actor ID>".*
- `def setLocation (self, x, y)`  
*Sets the cell coordinates of this object.*
- `def addedToWorld (self, world)`  
*Sets the world this actor is into.*
- `def getWorld (self)`  
*Gets the world this object is into.*
- `def getX (self)`  
*Gets the X coordinate of the cell this actor object is into.*
- `def getY (self)`  
*Gets the Y coordinate of the cell this actor object is into.*
- `def \_\_str\_\_ (self)`  
*Return a string with this actor ID and position.*

### 3.1.1 Constructor & Destructor Documentation

#### 3.1.1.1 `__init__()`

```
def Actor.Actor.__init__ (
    self )
```

Construct a new [Actor](#) object.

- Sets the initial values of its member variables.
- Sets the unique ID for the object and initializes the reference to the World object to which this [Actor](#) object belongs to null.
- The ID of the first [Actor](#) object is 0.
- The ID gets incremented by one each time a new [Actor](#) object is created.
- Sets the iteration counter to zero and initialize the location of the object to cell (0,0).

Reimplemented in [Disease.Disease](#).

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `act()`

```
def Actor.Actor.act (
    self )
```

Prints on screen in the format "Iteration <ID>: Actor <Actor ID>".

The < *ID* > is replaced by the current iteration number. < *ActorID* > is replaced by the unique ID of the [Actor](#) object that performs the act(self) method.

For instance, the actor with ID 1 shows the following result on the output screen after its act(self) method has been called twice.

```
Iteration 0: Actor 1
Iteration 1: Actor 1
```

Reimplemented in [Disease.Disease](#).

#### 3.1.2.2 `addedToWorld()`

```
def Actor.Actor.addedToWorld (
    self,
    world )
```

Sets the world this actor is into.

#### Parameters

<i>world</i>	Reference to the World object this <a href="#">Actor</a> object is added.
--------------	---

#### Exceptions

<i>RuntimeError</i>	when world is null.
---------------------	---------------------

#### 3.1.2.3 getID()

```
def Actor.Actor.getID (
    self )
```

Used for testing.

#### Returns

ActorID

#### 3.1.2.4 getWorld()

```
def Actor.Actor.getWorld (
    self )
```

Gets the world this object in into.

#### Returns

the world this object belongs to

#### 3.1.2.5 getX()

```
def Actor.Actor.getX (
    self )
```

Gets the X coordinate of the cell this actor object is into.

#### Returns

the x coordinate of this [Actor](#) object.

### 3.1.2.6 getY()

```
def Actor.Actor.getY (
    self )
```

Gets the Y coordinate of the cell this actor object is into.

#### Returns

the y coordinate of this [Actor](#) object.

### 3.1.2.7 Iteration()

```
def Actor.Actor.Iteration (
    self )
```

Used for testing.

#### Returns

number of iterations

### 3.1.2.8 setLocation()

```
def Actor.Actor.setLocation (
    self,
    x,
    y )
```

Sets the cell coordinates of this object.

#### Parameters

<i>x</i>	the column.
<i>y</i>	the row.

#### Exceptions

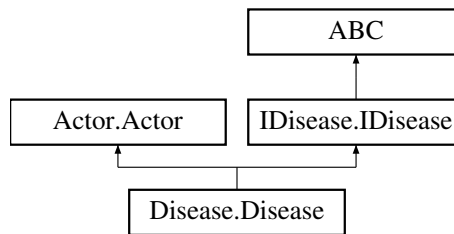
<i>ValueError</i>	when $x < 0$ or $x \geq$ world width,
<i>ValueError</i>	when $y < 0$ or $y \geq$ world height,
<i>RuntimeError</i>	when the world is null.

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

- Actor.py

## 3.2 Disease.Disease Class Reference

Inheritance diagram for Disease.Disease:



### Public Member Functions

- `def __init__ (self)`  
*Constructor.*
- `def setGrowthCondition (self, lTemp, hTemp, gRate)`  
*Sets the disease growth rate, lower temperature and higher temperature.*
- `def getGrowthCondition (self)`  
*Returns the disease growth rate, lower temperature and higher temperature.*
- `def getQuadrant (self)`  
*Returns the quadrant of this disease.*
- `def act (self)`  
*Print on screen in the format "Iteration <ID>: Actor <Actor ID>." The < ID > is replaced by the current iteration number.*
- `def getStrength (self)`  
*Return the disease strength of this object.*
- `def __str__ (self)`  
*Return a string with the strength, growth and quadrant of this disease.*

### 3.2.1 Constructor & Destructor Documentation

#### 3.2.1.1 \_\_init\_\_()

```
def Disease.Disease.__init__ (
    self )
```

Constructor.

- Call its superclass's default constructor.
- Initialize the lower bound and the upper bound temperatures for the growth rate to 0.
- Set the growth rate to 0.
- Set the disease strength to 1.

Reimplemented from [Actor.Actor](#).

## 3.2.2 Member Function Documentation

### 3.2.2.1 act()

```
def Disease.Disease.act (
    self )
```

Print on screen in the format "Iteration <ID>: Actor <Actor ID>." The < *ID* > is replaced by the current iteration number.

< *ActorID* > is replaced by the unique ID of the Actor object that performs the [act\(\)](#) method.

Reimplemented from [Actor.Actor](#).

### 3.2.2.2 getGrowthCondition()

```
def Disease.Disease.getGrowthCondition (
    self )
```

Returns the disease growth rate, lower temperature and higher temperature.

#### Returns

growth rate, lower temp and higher temp

### 3.2.2.3 getQuadrant()

```
def Disease.Disease.getQuadrant (
    self )
```

Returns the quadrant of this disease.

#### Returns

0, 1, 2 or 3.

#### 3.2.2.4 getStrength()

```
def Disease.Disease.getStrength (
    self )
```

Return the disease strength of this object.

##### Returns

disease strength of the object.

Reimplemented from [IDisease.IDisease](#).

#### 3.2.2.5 setGrowthCondition()

```
def Disease.Disease.setGrowthCondition (
    self,
    lTemp,
    hTemp,
    gRate )
```

Sets the disease growth rate, lower temperature and higher temperature.

##### Parameters

<i>lTemp</i>	Lower bound temperature for the disease to grow at this gRate.
<i>hTemp</i>	Upper bound temperature for the disease to grow at this gRate.
<i>gRate</i>	The growth rate.

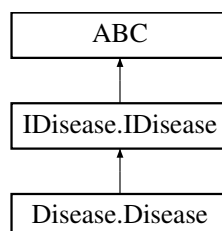
Reimplemented from [IDisease.IDisease](#).

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

- Disease.py

### 3.3 IDisease.IDisease Class Reference

Inheritance diagram for IDisease.IDisease:



## Public Member Functions

- def [setGrowthCondition](#) (self, lTemp, hTemp, gRate)  
*Set the growth condition of a Disease object to gRate.*
- def [getStrength](#) (self)  
*Return the disease strength of the object implements this interface.*

### 3.3.1 Member Function Documentation

#### 3.3.1.1 setGrowthCondition()

```
def IDisease.IDisease.setGrowthCondition (
    self,
    lTemp,
    hTemp,
    gRate )
```

Set the growth condition of a Disease object to gRate.

The value of gRate gets multiplied to the current disease strength only when the disease is located in the world region with the average temperature in between the values of lTemp and hTemp.

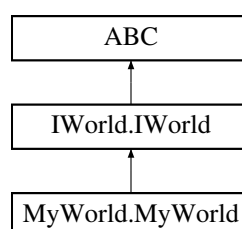
Reimplemented in [Disease.Disease](#).

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

- IDisease.py

## 3.4 IWorld.IWorld Class Reference

Inheritance diagram for IWorld.IWorld:





## Public Member Functions

- def **prepare** (self)
- def **setTemp** (self, quad, temp)
- def **getTemp** (self, quad)
- def **getObjects** (self)
- def **getSumStrength** (self)
- def **initDiseases** (self, numDisStr)
- def **initLocations** (self, locationsStr, diseaseArr)
- def **initGrowthConditions** (self, growthStr, diseaseArr)
- def **initTemps** (self, tempStr)

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

- IWorld.py

## 3.5 mapper.mapper Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, world, viewport, ydown=True, noDistortion=True)  
*Constructor.*
- def **windowVecToViewport** (self, x, y)  
*Maps a single vector from world coordinates to viewport (screen) coordinates.*
- def **viewportToWindow** (self, x, y)  
*Maps a single point from screen coordinates to window (world) coordinates.*
- def **windowToViewport** (self, \*p)  
*Maps points from world coordinates to viewport (screen) coordinates.*

### Public Attributes

- **world**
- **viewport**
- **fx**
- **fy**
- **ys**
- **f**
- **c\_1**
- **c\_2**

### 3.5.1 Constructor & Destructor Documentation

#### 3.5.1.1 \_\_init\_\_()

```
def mapper.mapper.__init__ (
    self,
    world,
    viewport,
    ydown = True,
    noDistortion = True )
```

Constructor.

**Parameters**

<i>world</i>	window rectangle.
<i>viewport</i>	screen rectangle.
<i>ydown</i>	whether Y axis is upside down.
<i>noDistortion</i>	whether to use the same scale for both X and Y.

## 3.5.2 Member Function Documentation

### 3.5.2.1 viewportToWorld()

```
def mapper.mapper.viewportToWorld (
    self,
    x,
    y )
```

Maps a single point from screen coordinates to window (world) coordinates.

**Parameters**

<i>x,y</i>	given point.
------------	--------------

**Returns**

a new point in world coordinates.

### 3.5.2.2 windowToViewport()

```
def mapper.mapper.windowToViewport (
    self,
    * p )
```

Maps points from world coordinates to viewport (screen) coordinates.

**Parameters**

<i>p</i>	a variable number of points.
----------	------------------------------

**Returns**

two new points in screen coordinates.

### 3.5.2.3 windowVecToViewport()

```
def mapper.mapper.windowVecToViewport (
    self,
    x,
    y )
```

Maps a single vector from world coordinates to viewport (screen) coordinates.

#### Parameters

x,y	given vector.
-----	---------------

#### Returns

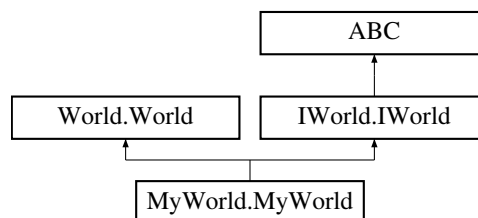
a new vector in screen coordinates.

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

- mapper.py

## 3.6 MyWorld.MyWorld Class Reference

Inheritance diagram for MyWorld.MyWorld:



### Public Member Functions

- def `__init__` (self)  
Call the constructor of the World class with the width and height of 8 and 8 pixels, with 64 cells , respectively .
- def `act` (self)  
This method overrides the act () method in the World class .
- def `prepare` (self)  
Prepare the world .
- def `setTemp` (self, quad, temp)  
Set the temperature of the region of the world to the value of temp .
- def `getTemp` (self, quad)  
Return the temperature of the world region with the ID of quadID .
- def `initDiseases` (self, numDisStr)  
Create Disease objects ; the number of the objects equals to the value passed in numDisStr .
- def `initLocations` (self, locationsStr, diseaseArr)

Add each Disease object into the [MyWorld](#) object implementing this method according to the information in locationStr .

- def [initGrowthConditions](#) (self, growthStr, diseaseArr)  
Set the lower bound and upper bound temperature and the growth rate for each disease according to the input growthStr .
- def [initTemps](#) (self, tempStr)  
Set the temperature for each quadrant of the [MyWorld](#) according to the value of the tempStr .
- def [getSumStrength](#) (self)  
Return the total disease strength of all the diseases in the class implementing this interface .

## Public Attributes

- [avgTemp](#)

## 3.6.1 Constructor & Destructor Documentation

### 3.6.1.1 [\\_\\_init\\_\\_\(\)](#)

```
def MyWorld.MyWorld.__init__ (
    self )
```

Call the constructor of the World class with the width and height of 8 and 8 pixels, with 64 cells , respectively .

Initialize a list to keep the average temperature of each world region ( quadrant ).

Call the prepare () method .

## 3.6.2 Member Function Documentation

### 3.6.2.1 [act\(\)](#)

```
def MyWorld.MyWorld.act (
    self )
```

This method overrides the act () method in the World class .

This method prints :

" Iteration < ITRID >: World disease strength is < WorldDisease >" where < ITRID > is replaced by the current iteration number and < WorldDisease > is replaced by the returned value of [getSumStrength](#) () in 2 decimal places . An example is below .

Iteration 0: World disease strength is 2.00 Iteration 1: World disease strength is 3.00

Reimplemented from [World.World](#).

### 3.6.2.2 [getTemp\(\)](#)

```
def MyWorld.MyWorld.getTemp (
    self,
    quad )
```

Return the temperature of the world region with the ID of quadID .

The valid value is between zero and three inclusive .

**Exceptions**

<i>ValueError</i>	if the quad id is not within 0 and 3
-------------------	--------------------------------------

Reimplemented from [IWorld.IWorld](#).

**3.6.2.3 initDiseases()**

```
def MyWorld.MyWorld.initDiseases (
    self,
    numDisStr )
```

Create Disease objects ; the number of the objects equals to the value passed in numDisStr .

Return a list of object references to the created Disease objects .

An example of a valid numDisStr is below .

Ex : "2"

If numDisStr is None or it cannot be converted to a positive integer , print a message on screen " Check the NumDiseases line in simulation . config ." and return None .

No exceptions are thrown .

Reimplemented from [IWorld.IWorld](#).

**3.6.2.4 initGrowthConditions()**

```
def MyWorld.MyWorld.initGrowthConditions (
    self,
    growthStr,
    diseaseArr )
```

Set the lower bound and upper bound temperature and the growth rate for each disease according to the input growthStr .

An example of a valid string for two Disease objects is :

Ex : "10.0 ,15.0 ,2.0;10.0 ,13.0 ,3.0"

If growthStr is empty or not in the correct format or does not have all the growth for all the Disease objects in the Disease array , print on screen " Check the DiseasesGrowth line in simulation . config ." and return -1.

Return 0 for a successful initialization of the Disease growth conditions . No exceptions are thrown .

Reimplemented from [IWorld.IWorld](#).

### 3.6.2.5 initLocations()

```
def MyWorld.MyWorld.initLocations (
    self,
    locationsStr,
    diseaseArr )
```

Add each Disease object into the [MyWorld](#) object implementing this method according to the information in locationStr .

An example of a locationStr is "200 ,200;400 ,480". This means that the first Disease is planted at cell (200 ,200) and the second Disease is at cell (400 , 480).

If the locationStr is empty or not in the correct format or does not have all the cell coordinates of all the Disease objects , print on screen " Check the Locations line in simulation . config " and return -1.

Return 0 for a successful initialization of the Disease locations . No exceptions are thrown .

Reimplemented from [IWorld.IWorld](#).

### 3.6.2.6 initTemps()

```
def MyWorld.MyWorld.initTemps (
    self,
    tempStr )
```

Set the temperature for each quadrant of the [MyWorld](#) according to the value of the tempStr .

An example of tempStr is below . The region temperatures for regions 0 , 1 , 2 , and 3 are 12 , 20 , 50 , and 100 , respectively .

Return 0 for a successful initialization of the quadrant temperatures . No exceptions are thrown .

Ex : "12;20;50;100"

If tempStr is empty or not in the correct format or does not have all the temperatures of all the regions , print on screen " Check the Temperature line in simulation . config ." and return -1

Reimplemented from [IWorld.IWorld](#).

### 3.6.2.7 prepare()

```
def MyWorld.MyWorld.prepare (
    self )
```

Prepare the world .

Open a text file named " simulation . config " in the current path ( directly under the project directory ). Parse the configuration file for the number of Disease objects , the cell locations of these objects , the growth rates , and the temperature ranges associated with individual growth rates . Read Section 4 on the content of the configuration file before reading the rest .

## Exceptions

<i>IOError</i>	when there are problems with opening a file, reading, or writing to a file.
----------------	---

Reimplemented from [IWorld.IWorld](#).

**3.6.2.8 setTemp()**

```
def MyWorld.MyWorld.setTemp (
    self,
    quad,
    temp )
```

Set the temperature of the region of the world to the value of temp .

The quadID indicates the region . The valid value is between [0 , 3]. Any value of float is accepted for temp .

## Exceptions

<i>ValueError</i>	if the quad id is not within 0 and 3
-------------------	--------------------------------------

Reimplemented from [IWorld.IWorld](#).

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

- MyWorld.py

**3.7 SimulationPanel.SimulationPanel Class Reference****Public Member Functions**

- def **\_\_init\_\_** (self, world, canvas)
- def **distance2circles** (self, ab, bc)
- def **draw** (self)
- def **auxDraw** (self)
- def **resize** (self, event=None)
- def **mousePressed** (self, event)
- def **printData** (self)
- def **rgb** (self)
- def **wvmap** (self)
- def **wvmap** (self, q)

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

- SimulationPanel.py

## 3.8 Timer.Timer Class Reference

### Public Member Functions

- `def __init__ (self, root, callback, delay)`
- `def run (self)`
- `def stop (self)`
- `def restart (self)`

### Public Attributes

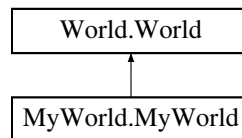
- `root`
- `callback`
- `delay`
- `task`

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

- `Timer.py`

## 3.9 World.World Class Reference

Inheritance diagram for World.World:



### Public Member Functions

- `def __init__ (self, worldWidth, worldHeight)`
- `def createGrid (self, h, w, d)`
- `def __str__ (self)`
- `def __repr__ (self)`
- `def act (self)`
- `def addObject (self, object, x, y)`
- `def getHeight (self)`
- `def getWidth (self)`
- `def getDepth (self)`
- `def numberOfObjects (self)`
- `def getObjects (self)`
- `def setGrid (self, aGrid, numObjs)`

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

- `World.py`



# Index

- `__init__`
    - `Actor.Actor`, 6
    - `Disease.Disease`, 9
    - `mapper.mapper`, 13
    - `MyWorld.MyWorld`, 16
- `act`
  - `Actor.Actor`, 6
  - `Disease.Disease`, 10
  - `MyWorld.MyWorld`, 16
- `Actor.Actor`, 5
  - `__init__`, 6
  - `act`, 6
  - `addedToWorld`, 6
  - `getID`, 7
  - `getWorld`, 7
  - `getX`, 7
  - `getY`, 7
  - `Iteration`, 8
  - `setLocation`, 8
- `addedToWorld`
  - `Actor.Actor`, 6
- `Disease.Disease`, 9
  - `__init__`, 9
  - `act`, 10
  - `getGrowthCondition`, 10
  - `getQuadrant`, 10
  - `getStrength`, 10
  - `setGrowthCondition`, 11
- `getGrowthCondition`
  - `Disease.Disease`, 10
- `getID`
  - `Actor.Actor`, 7
- `getQuadrant`
  - `Disease.Disease`, 10
- `getStrength`
  - `Disease.Disease`, 10
- `getTemp`
  - `MyWorld.MyWorld`, 16
- `getWorld`
  - `Actor.Actor`, 7
- `getX`
  - `Actor.Actor`, 7
- `getY`
  - `Actor.Actor`, 7
- `IDisease.IDisease`, 11
  - `setGrowthCondition`, 12
- `initDiseases`
  - `MyWorld.MyWorld`, 17
- `initGrowthConditions`
  - `MyWorld.MyWorld`, 17
- `initLocations`
  - `MyWorld.MyWorld`, 17
- `initTemps`
  - `MyWorld.MyWorld`, 18
- `Iteration`
  - `Actor.Actor`, 8
- `IWorld.IWorld`, 12
- `mapper.mapper`, 13
  - `__init__`, 13
  - `viewportToWindow`, 14
  - `windowToViewport`, 14
  - `windowVecToViewport`, 14
- `MyWorld.MyWorld`, 15
  - `__init__`, 16
  - `act`, 16
  - `getTemp`, 16
  - `initDiseases`, 17
  - `initGrowthConditions`, 17
  - `initLocations`, 17
  - `initTemps`, 18
  - `prepare`, 18
  - `setTemp`, 19
- `prepare`
  - `MyWorld.MyWorld`, 18
- `setGrowthCondition`
  - `Disease.Disease`, 11
  - `IDisease.IDisease`, 12
- `setLocation`
  - `Actor.Actor`, 8
- `setTemp`
  - `MyWorld.MyWorld`, 19
- `SimulationPanel.SimulationPanel`, 19
- `Timer.Timer`, 20
- `viewportToWindow`
  - `mapper.mapper`, 14
- `windowToViewport`
  - `mapper.mapper`, 14
- `windowVecToViewport`
  - `mapper.mapper`, 14
- `World.World`, 20