

# FaultInOurPong

Version 0

Team 03, Pongthusiastics  
Adwity Sharma - sharma78  
Arfa Butt - buttaa3  
Jie Luo - luoj3

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>The main page for the game FaultInOurPong.</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Packages . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	Package model . . . . .	11
6.2	Package startGame . . . . .	11
6.3	Package view . . . . .	11

<b>7</b>	<b>Class Documentation</b>	<b>13</b>
7.1	model.Ball Class Reference	13
7.1.1	Constructor & Destructor Documentation	13
7.1.1.1	Ball()	13
7.1.2	Member Function Documentation	14
7.1.2.1	getPositionX()	14
7.1.2.2	getPositionY()	14
7.1.2.3	getSize()	14
7.1.2.4	setPositionX()	14
7.1.2.5	setPositionY()	14
7.1.3	Member Data Documentation	15
7.1.3.1	positionX	15
7.1.3.2	positionY	15
7.1.3.3	SIZE	15
7.1.3.4	speed	15
7.2	startGame.GameController Class Reference	15
7.2.1	Constructor & Destructor Documentation	16
7.2.1.1	GameController()	16
7.2.2	Member Function Documentation	17
7.2.2.1	checkGameOver()	17
7.2.2.2	display()	17
7.2.3	Member Data Documentation	17
7.2.3.1	ADVANCE	17
7.2.3.2	ai	17
7.2.3.3	b	18
7.2.3.4	ballX	18
7.2.3.5	bomb	18
7.2.3.6	bombX	18
7.2.3.7	bottomPadX	18
7.2.3.8	frameWidth	18

7.2.3.9	gameDisplay	18
7.2.3.10	gameFrame	18
7.2.3.11	gameMode	19
7.2.3.12	inset	19
7.2.3.13	keys	19
7.2.3.14	m	19
7.2.3.15	mode	19
7.2.3.16	paddle_player	19
7.2.3.17	padWidth	19
7.2.3.18	player	19
7.2.3.19	record	19
7.2.3.20	scoreTop	19
7.2.3.21	SINGLE	20
7.2.3.22	t	20
7.2.3.23	tut	20
7.2.3.24	v	20
7.2.3.25	velX	20
7.2.3.26	w	20
7.3	model.GameModel Class Reference	21
7.3.1	Constructor & Destructor Documentation	21
7.3.1.1	GameModel()	21
7.3.2	Member Function Documentation	22
7.3.2.1	getBall()	22
7.3.2.2	getBomb()	22
7.3.2.3	getComputer()	22
7.3.2.4	getComputerPaddle()	22
7.3.2.5	getPlayer()	22
7.3.2.6	getPlayerPaddle()	23
7.3.2.7	setBall()	23
7.3.2.8	setBomb()	23

7.3.3	Member Data Documentation . . . . .	23
7.3.3.1	b . . . . .	23
7.3.3.2	p_player . . . . .	23
7.3.3.3	player . . . . .	24
7.4	view.GameView Class Reference . . . . .	24
7.4.1	Constructor & Destructor Documentation . . . . .	25
7.4.1.1	GameView() . . . . .	25
7.4.2	Member Function Documentation . . . . .	25
7.4.2.1	cannotLoadMessage() . . . . .	25
7.4.2.2	createGame() . . . . .	25
7.4.2.3	display() . . . . .	25
7.4.2.4	gameOver() . . . . .	25
7.4.2.5	getFrameHeight() . . . . .	26
7.4.2.6	getFrameWidth() . . . . .	26
7.4.2.7	getGame() . . . . .	26
7.4.2.8	getGameFrame() . . . . .	26
7.4.2.9	getmode() . . . . .	27
7.4.2.10	getTutorial() . . . . .	27
7.4.2.11	getWelcome() . . . . .	27
7.4.2.12	noFileAvailMessage() . . . . .	27
7.4.2.13	tutorialPage() . . . . .	27
7.4.3	Member Data Documentation . . . . .	28
7.4.3.1	FRAMEHEIGHT . . . . .	28
7.4.3.2	FRAMEWIDTH . . . . .	28
7.4.3.3	gameFrame . . . . .	28
7.4.3.4	mode . . . . .	28
7.4.3.5	ponggame . . . . .	28
7.4.3.6	tutorial . . . . .	28
7.4.3.7	welcome . . . . .	28
7.5	view.Mode Class Reference . . . . .	29

7.5.1	Constructor & Destructor Documentation . . . . .	29
7.5.1.1	Mode() . . . . .	29
7.5.2	Member Function Documentation . . . . .	29
7.5.2.1	addButton() . . . . .	29
7.5.2.2	addListener() . . . . .	29
7.5.2.3	getSingle() . . . . .	30
7.5.3	Member Data Documentation . . . . .	30
7.5.3.1	buttonPanel . . . . .	30
7.5.3.2	single . . . . .	30
7.5.3.3	sObstacle . . . . .	30
7.6	model.Paddle Class Reference . . . . .	31
7.6.1	Constructor & Destructor Documentation . . . . .	31
7.6.1.1	Paddle() . . . . .	31
7.6.2	Member Function Documentation . . . . .	31
7.6.2.1	getHeight() . . . . .	31
7.6.2.2	getInset() . . . . .	32
7.6.2.3	getPositionX() . . . . .	32
7.6.2.4	getPositionY() . . . . .	32
7.6.2.5	getWidth() . . . . .	32
7.6.2.6	setPositionX() . . . . .	32
7.6.2.7	setPositionY() . . . . .	33
7.6.3	Member Data Documentation . . . . .	33
7.6.3.1	HEIGHT . . . . .	33
7.6.3.2	INSET . . . . .	33
7.6.3.3	positionX . . . . .	33
7.6.3.4	positionY . . . . .	33
7.6.3.5	speed . . . . .	34
7.6.3.6	WIDTH . . . . .	34
7.7	model.Player Class Reference . . . . .	34
7.7.1	Constructor & Destructor Documentation . . . . .	34

7.7.1.1	Player()	34
7.7.2	Member Function Documentation	35
7.7.2.1	checkLoss()	35
7.7.2.2	decrementLife()	35
7.7.2.3	getScore()	35
7.7.3	Member Data Documentation	35
7.7.3.1	LIFE	35
7.7.3.2	NOLIFE	35
7.7.3.3	score	36
7.8	startGame.PongGame Class Reference	36
7.8.1	Member Function Documentation	36
7.8.1.1	main()	36
7.9	view.PongGameDisplay Class Reference	37
7.9.1	Constructor & Destructor Documentation	38
7.9.1.1	PongGameDisplay()	38
7.9.2	Member Function Documentation	38
7.9.2.1	getBallX()	38
7.9.2.2	getBallY()	38
7.9.2.3	getBottomX()	39
7.9.2.4	getBottomY()	39
7.9.2.5	paintComponent()	39
7.9.2.6	setAdvance()	39
7.9.2.7	setBall()	39
7.9.2.8	setBallSize()	40
7.9.2.9	setBottom()	40
7.9.2.10	setBottomScore()	40
7.9.2.11	setInset()	40
7.9.2.12	setPaddleHeight()	41
7.9.2.13	setPaddleWidth()	41
7.9.2.14	setTop()	41



7.9.2.15	setTopScore()	41
7.9.3	Member Data Documentation	42
7.9.3.1	ADVANCE	42
7.9.3.2	ballSize	42
7.9.3.3	ballX	42
7.9.3.4	bottomPadX	42
7.9.3.5	first	42
7.9.3.6	frameHeight	42
7.9.3.7	frameWidth	42
7.9.3.8	gameMode	43
7.9.3.9	inset	43
7.9.3.10	padW	43
7.9.3.11	scoreTop	43
7.9.3.12	SINGLE	43
7.9.3.13	topPadX	43
7.10	view.Tutorial Class Reference	43
7.10.1	Constructor & Destructor Documentation	44
7.10.1.1	Tutorial()	44
7.10.2	Member Function Documentation	44
7.10.2.1	addListener()	44
7.10.2.2	getBack()	44
7.10.3	Member Data Documentation	45
7.10.3.1	back	45
7.11	view.Welcome Class Reference	45
7.11.1	Constructor & Destructor Documentation	46
7.11.1.1	Welcome()	46
7.11.2	Member Function Documentation	46
7.11.2.1	addButton()	46
7.11.2.2	addListener()	46
7.11.2.3	exit()	46
7.11.2.4	getStart()	47
7.11.2.5	highScores()	47
7.11.2.6	load()	47
7.11.2.7	tutorial()	47
7.11.3	Member Data Documentation	47
7.11.3.1	buttonPanel	47
7.11.3.2	exit	48
7.11.3.3	highScores	48
7.11.3.4	load	48
7.11.3.5	start	48
7.11.3.6	tutorial	48

<b>8 File Documentation</b>	<b>49</b>
8.1 src/model/Ball.java File Reference . . . . .	49
8.1.1 Detailed Description . . . . .	49
8.2 src/model/GameModel.java File Reference . . . . .	49
8.2.1 Detailed Description . . . . .	50
8.3 src/model/Paddle.java File Reference . . . . .	50
8.3.1 Detailed Description . . . . .	50
8.4 src/model/Player.java File Reference . . . . .	51
8.4.1 Detailed Description . . . . .	51
8.5 src/startGame/GameController.java File Reference . . . . .	51
8.5.1 Detailed Description . . . . .	52
8.6 src/startGame/PongGame.java File Reference . . . . .	52
8.6.1 Detailed Description . . . . .	52
8.7 src/view/MapView.java File Reference . . . . .	53
8.7.1 Detailed Description . . . . .	53
8.8 src/view/Mode.java File Reference . . . . .	53
8.8.1 Detailed Description . . . . .	54
8.9 src/view/PongGameDisplay.java File Reference . . . . .	54
8.9.1 Detailed Description . . . . .	54
8.10 src/view/Tutorial.java File Reference . . . . .	55
8.10.1 Detailed Description . . . . .	55
8.11 src/view/Welcome.java File Reference . . . . .	55
8.11.1 Detailed Description . . . . .	55
<b>Index</b>	<b>57</b>

## **Chapter 1**

**The main page for the game FaultInOurPong.**



## Chapter 2

# Namespace Index

### 2.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">model</a>	11
<a href="#">startGame</a>	11
<a href="#">view</a>	11



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

model.Ball . . . . .	13
startGame.GameController . . . . .	15
model.GameModel . . . . .	21
view.GameView . . . . .	24
model.Paddle . . . . .	31
model.Player . . . . .	34
startGame.PongGame . . . . .	36
JFrame	
view.Mode . . . . .	29
view.Tutorial . . . . .	43
view.Welcome . . . . .	45
JPanel	
view.PongGameDisplay . . . . .	37





## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">model.Ball</a>	13
<a href="#">startGame.GameController</a>	15
<a href="#">model.GameModel</a>	21
<a href="#">view.GameView</a>	24
<a href="#">view.Mode</a>	29
<a href="#">model.Paddle</a>	31
<a href="#">model.Player</a>	34
<a href="#">startGame.PongGame</a>	36
<a href="#">view.PongGameDisplay</a>	37
<a href="#">view.Tutorial</a>	43
<a href="#">view.Welcome</a>	45



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

src/model/ <a href="#">Ball.java</a>	
This class represents a ball on the pong game . . . . .	49
src/model/ <a href="#">GameModel.java</a>	
This class represents a ball on the pong game . . . . .	49
src/model/ <a href="#">Paddle.java</a>	
This class defines a paddle . . . . .	50
src/model/ <a href="#">Player.java</a>	
This class represents a player for the game . . . . .	51
src/startGame/ <a href="#">GameController.java</a>	
This class is the controller for the game . . . . .	51
src/startGame/ <a href="#">PongGame.java</a>	
This class starts the game . . . . .	52
src/view/ <a href="#">GameView.java</a>	
This class is the main view model . . . . .	53
src/view/ <a href="#">Mode.java</a>	
This class create the game mode window . . . . .	53
src/view/ <a href="#">PongGameDisplay.java</a>	
This class construct the view of the pong game . . . . .	54
src/view/ <a href="#">Tutorial.java</a>	
This class create the tutorial window . . . . .	55
src/view/ <a href="#">Welcome.java</a>	
This class creates the display for welcome page . . . . .	55



## Chapter 6

# Namespace Documentation

### 6.1 Package model

#### Classes

- class [Ball](#)
- class [GameModel](#)
- class [Paddle](#)
- class [Player](#)

### 6.2 Package startGame

#### Classes

- class [GameController](#)
- class [PongGame](#)

### 6.3 Package view

#### Classes

- class [GameView](#)
- class [Mode](#)
- class [PongGameDisplay](#)
- class [Tutorial](#)
- class [Welcome](#)



# Chapter 7

## Class Documentation

### 7.1 model.Ball Class Reference

#### Public Member Functions

- `Ball ()`  
*Constructor for `Ball`.*
- `void setPositionX (int x)`  
*sets the x position of the ball*
- `void setPositionY (int y)`  
*sets the y position of the ball*
- `int getPositionX ()`  
*gets the x-position of the ball*
- `int getPositionY ()`  
*gets the y-position of the ball*
- `int getSize ()`  
*gets the size of the ball*

#### Private Attributes

- `int positionX`
- `int positionY`
- `final int SIZE = 20`
- `int speed`

#### 7.1.1 Constructor & Destructor Documentation

##### 7.1.1.1 Ball()

```
model.Ball.Ball ( )
```

Constructor for `Ball`.

Constructor accepts the x and y position of the ball

## 7.1.2 Member Function Documentation

### 7.1.2.1 getPositionX()

```
int model.Ball.getPositionX ( )
```

gets the x-position of the ball

#### Returns

positionX

### 7.1.2.2 getPositionY()

```
int model.Ball.getPositionY ( )
```

gets the y-position of the ball

#### Returns

positionY

### 7.1.2.3 getSize()

```
int model.Ball.getSize ( )
```

gets the size of the ball

#### Returns

SIZE

### 7.1.2.4 setPositionX()

```
void model.Ball.setPositionX (
    int x )
```

sets the x position of the ball

#### Parameters

<i>x-position</i>	of the ball
-------------------	-------------

### 7.1.2.5 setPositionY()

```
void model.Ball.setPositionY (
```



```
int y )
```

sets the y position of the ball

#### Parameters

<i>y-position</i>	of the ball
-------------------	-------------

### 7.1.3 Member Data Documentation

#### 7.1.3.1 positionX

```
int model.Ball.positionX [private]
```

The X and Y position of a ball on the screen

#### 7.1.3.2 positionY

```
int model.Ball.positionY [private]
```

#### 7.1.3.3 SIZE

```
final int model.Ball.SIZE = 20 [private]
```

The size of a ball

#### 7.1.3.4 speed

```
int model.Ball.speed [private]
```

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

- [src/model/Ball.java](#)

## 7.2 startGame.GameController Class Reference

### Classes

- class **GameListener**  
*action listener for the game page*
- class **ModeListener**  
*action listener for the game mode page*
- class **TutorialListener**  
*action listener for the tutorial page*
- class **WelcomepageListener**  
*action listener for the welcome page*

## Public Member Functions

- [GameController](#) ([GameView](#) v, [GameModel](#) m)
- void [display](#) ()  
*sets the display*
- void [checkGameOver](#) ()  
*checks whether the game ends*

## Private Attributes

- [GameView](#) v
- [GameModel](#) m
- [Welcome](#) w
- [Mode](#) mode
- [Tutorial](#) tut
- [HashSet](#)< [String](#) > [keys](#) = new [HashSet](#)<[String](#)>()
- [JFrame](#) [gameFrame](#)
- int [frameWidth](#)
- [PongGameDisplay](#) [gameDisplay](#)
- int [velX](#) = 1
- int [padWidth](#)
- int [bottomPadX](#)
- [Ball](#) b
- [Paddle](#) [paddle\\_player](#)
- int [ballX](#)
- int [scoreTop](#)
- int [inset](#)
- final int [SINGLE](#) = 0
- final int [ADVANCE](#) = 1
- int [gameMode](#)
- [Ball](#) [bomb](#)
- int [bombX](#)
- [Player](#) [player](#)
- [Player](#) [ai](#)
- [Timer](#) t
- [Timer](#) [record](#)

## 7.2.1 Constructor & Destructor Documentation

### 7.2.1.1 GameController()

```
startGame.GameController.GameController (
    GameView v,
    GameModel m )
```

Obtain the window frame dimentions

Setups for ball in the Model

Setups for the bomb in the Model

Setups for the paddles in the Model

- obtain paddle dimensions
- initialize paddle positions for the player paddle
- initialize paddle positions for the ai paddle

Setups for the players in the Model

- initialize number of life for the player and the ai

Setups for the View

- obtain windows from the view
- add action listener for different windows

## 7.2.2 Member Function Documentation

### 7.2.2.1 checkGameOver()

```
void startGame.GameController.checkGameOver ( )
```

checks whether the game ends

check the number of life for both the player and the ai is 0.

- If the number of life for the ai is 0, the player wins
- If the number of life for the player is 0, the ai wins.

### 7.2.2.2 display()

```
void startGame.GameController.display ( )
```

sets the display

opens a window

## 7.2.3 Member Data Documentation

### 7.2.3.1 ADVANCE

```
final int startGame.GameController.ADVANCE = 1 [private]
```

### 7.2.3.2 ai

```
Player startGame.GameController.ai [private]
```

#### 7.2.3.3 b

```
Ball startGame.GameController.b [private]
```

#### 7.2.3.4 ballX

```
int startGame.GameController.ballX [private]
```

#### 7.2.3.5 bomb

```
Ball startGame.GameController.bomb [private]
```

#### 7.2.3.6 bombX

```
int startGame.GameController.bombX [private]
```

#### 7.2.3.7 bottomPadX

```
int startGame.GameController.bottomPadX [private]
```

#### 7.2.3.8 frameWidth

```
int startGame.GameController.frameWidth [private]
```

#### 7.2.3.9 gameDisplay

```
PongGameDisplay startGame.GameController.gameDisplay [private]
```

#### 7.2.3.10 gameFrame

```
JFrame startGame.GameController.gameFrame [private]
```

Variable declarations for the game

- frame dimension
- paddle information
- ball information
- bomb information
- player information

### 7.2.3.11 gameMode

```
int startGame.GameController.gameMode [private]
```

### 7.2.3.12 inset

```
int startGame.GameController.inset [private]
```

### 7.2.3.13 keys

```
HashSet<String> startGame.GameController.keys = new HashSet<String>() [private]
```

Declare a variable for storing the key pressed records

### 7.2.3.14 m

```
GameModel startGame.GameController.m [private]
```

### 7.2.3.15 mode

```
Mode startGame.GameController.mode [private]
```

### 7.2.3.16 paddle\_player

```
Paddle startGame.GameController.paddle_player [private]
```

### 7.2.3.17 padWidth

```
int startGame.GameController.padWidth [private]
```

### 7.2.3.18 player

```
Player startGame.GameController.player [private]
```

### 7.2.3.19 record

```
Timer startGame.GameController.record [private]
```

### 7.2.3.20 scoreTop

```
int startGame.GameController.scoreTop [private]
```

### 7.2.3.21 SINGLE

```
final int startGame.GameController.SINGLE = 0 [private]
```

### 7.2.3.22 t

```
Timer startGame.GameController.t [private]
```

### 7.2.3.23 tut

```
Tutorial startGame.GameController.tut [private]
```

### 7.2.3.24 v

```
GameView startGame.GameController.v [private]
```

Import model and view to the controller (this interface).

### 7.2.3.25 velX

```
int startGame.GameController.velX =1 [private]
```

### 7.2.3.26 w

```
Welcome startGame.GameController.w [private]
```

Variable declarations for storing the game view windows

- welcome page
- mode page for showing different modes
- tutorial page for giving instructions to the students

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

- src/startGame/[GameController.java](#)

## 7.3 model.GameModel Class Reference

### Public Member Functions

- [GameModel](#) ()  
*Constructor for the game Model.*
- void [setBall](#) (int x, int y)  
*sets the x and y positions of a ball*
- void [setBomb](#) (int x, int y)  
*sets the x and y positions of a bomb*
- [Ball](#) [getBall](#) ()  
*gets the [Ball](#) object*
- [Ball](#) [getBomb](#) ()  
*gets the bomb object*
- [Paddle](#) [getPlayerPaddle](#) ()  
*gets the user paddle object*
- [Paddle](#) [getComputerPaddle](#) ()  
*gets the computer paddle object*
- [Player](#) [getPlayer](#) ()  
*gets the player object*
- [Player](#) [getComputer](#) ()  
*gets the computer object*

### Private Attributes

- [Ball](#) b
- [Paddle](#) p\_player
- [Player](#) player

### 7.3.1 Constructor & Destructor Documentation

#### 7.3.1.1 GameModel()

```
model.GameModel.GameModel ( )
```

Constructor for the game Model.

Contains all the data and models for the game, including the player, paddle, and the ball. Declara variables/instances for the model

- regular ball
- bomb
- paddle for the player
- paddle for the computer
- score and life for the player
- score and life for the ai

### 7.3.2 Member Function Documentation

#### 7.3.2.1 getBall()

`Ball model.GameModel.getBall ( )`

gets the `Ball` object

##### Returns

b is the ball object

#### 7.3.2.2 getBomb()

`Ball model.GameModel.getBomb ( )`

gets the bomb object

##### Returns

bomb is the bomb object

#### 7.3.2.3 getComputer()

`Player model.GameModel.getComputer ( )`

gets the computer object

##### Returns

computer

#### 7.3.2.4 getComputerPaddle()

`Paddle model.GameModel.getComputerPaddle ( )`

gets the computer paddle object

##### Returns

p\_computer

#### 7.3.2.5 getPlayer()

`Player model.GameModel.getPlayer ( )`

gets the player object

##### Returns

player



#### 7.3.2.6 getPlayerPaddle()

```
Paddle model.GameModel.getPlayerPaddle ( )
```

gets the user paddle object

##### Returns

p\_player

#### 7.3.2.7 setBall()

```
void model.GameModel.setBall (
    int x,
    int y )
```

sets the x and y positions of a ball

##### Parameters

x	is the x position of the ball
y	is the y position of the ball

#### 7.3.2.8 setBomb()

```
void model.GameModel.setBomb (
    int x,
    int y )
```

sets the x and y positions of a bomb

##### Parameters

x	is the x position of the bomb
y	is the y position of the bomb

### 7.3.3 Member Data Documentation

#### 7.3.3.1 b

```
Ball model.GameModel.b [private]
```

The ball object for the game

#### 7.3.3.2 p\_player

```
Paddle model.GameModel.p_player [private]
```

The two paddle in the game, one for the player and the other for the computer

### 7.3.3.3 player

`Player model.GameModel.player [private]`

The two players in the game, one for the user and the other for the computer

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

- `src/model/GameModel.java`

## 7.4 view.GameView Class Reference

### Public Member Functions

- `GameView ()`  
*Constructor for the view.*
- `void display ()`  
*displays the welcome page.*
- `Welcome getWelcome ()`  
*gets welcome page window*
- `Mode getmode ()`  
*gets game mode page window*
- `PongGameDisplay getGame ()`  
*gets game window*
- `Tutorial getTutorial ()`  
*gets tutorial page window*
- `void createGame ()`  
*create the game for display*
- `JFrame getGameFrame ()`  
*gets game object*
- `void noFileAvailMessage ()`  
*display message for error loading game record*
- `void cannotLoadMessage ()`  
*display message for error loading game*
- `void gameOver (int whichplayer)`  
*display message for game over*
- `void tutorialPage (ImageIcon img)`  
*create tutorial page*
- `int getFrameWidth ()`  
*gets width of the window*
- `int getFrameHeight ()`  
*gets height of the window*

### Private Attributes

- `Welcome welcome`
- `Mode mode`
- `PongGameDisplay ponggame`
- `Tutorial tutorial`
- `JFrame gameFrame`
- `final int FRAMEWIDTH = 700`
- `final int FRAMEHEIGHT = 500`

## 7.4.1 Constructor & Destructor Documentation

### 7.4.1.1 GameView()

```
view.GameView.GameView ( )
```

Constructor for the view.

declares all other windows

- Pass in different windows to this view interface
- Wait for further invocation

## 7.4.2 Member Function Documentation

### 7.4.2.1 cannotLoadMessage()

```
void view.GameView.cannotLoadMessage ( )
```

display message for error loading game

create a frame for display

### 7.4.2.2 createGame()

```
void view.GameView.createGame ( )
```

create the game for display

create a frame under set dimension for the game

### 7.4.2.3 display()

```
void view.GameView.display ( )
```

displays the welcome page.

sets the visibility of the window to be true.

### 7.4.2.4 gameOver()

```
void view.GameView.gameOver (
    int whichplayer )
```

display message for game over

**Parameters**

<i>whichplayer</i>	is the indicator for the player
--------------------	---------------------------------

- If the computer wins, display winning message for the computer
- If the player wins, display winning message for the player

**7.4.2.5 getFrameHeight()**

```
int view.GameView.getFrameHeight ( )
```

gets height of the window

**Returns**

FRAMEHEIGHT

**7.4.2.6 getFrameWidth()**

```
int view.GameView.getFrameWidth ( )
```

gets width of the window

**Returns**

FRAMEWIDTH

**7.4.2.7 getGame()**

```
PongGameDisplay view.GameView.getGame ( )
```

gets game window

**Returns**

game window object

**7.4.2.8 getGameFrame()**

```
JFrame view.GameView.getGameFrame ( )
```

gets game object

**Returns**

game object

#### 7.4.2.9 getmode()

```
Mode view.GameView.getmode ( )
```

gets game mode page window

##### Returns

game mode page object

#### 7.4.2.10 getTutorial()

```
Tutorial view.GameView.getTutorial ( )
```

gets tutorial page window

##### Returns

tutorial page object

#### 7.4.2.11 getWelcome()

```
Welcome view.GameView.getWelcome ( )
```

gets welcome page window

##### Returns

welcome page object

#### 7.4.2.12 noFileAvailMessage()

```
void view.GameView.noFileAvailMessage ( )
```

display message for error loading game record

create a frame for display

#### 7.4.2.13 tutorialPage()

```
void view.GameView.tutorialPage (
    ImageIcon img )
```

create tutorial page

#### Parameters

<i>img</i>	is the image for display
------------	--------------------------

### 7.4.3 Member Data Documentation

#### 7.4.3.1 FRAMEHEIGHT

```
final int view.GameView.FRAMEHEIGHT = 500 [private]
```

#### 7.4.3.2 FRAMEWIDTH

```
final int view.GameView.FRAMEWIDTH = 700 [private]
```

Constant declarations for the view

#### 7.4.3.3 gameFrame

```
JFrame view.GameView.gameFrame [private]
```

#### 7.4.3.4 mode

```
Mode view.GameView.mode [private]
```

#### 7.4.3.5 ponggame

```
PongGameDisplay view.GameView.ponggame [private]
```

#### 7.4.3.6 tutorial

```
Tutorial view.GameView.tutorial [private]
```

#### 7.4.3.7 welcome

```
Welcome view.GameView.welcome [private]
```

Variable declarations to store different pages

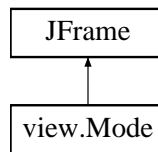
- welcome page
- game mode page
- tutorial page
- game page

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

- src/view/[GameView.java](#)

## 7.5 view.Mode Class Reference

Inheritance diagram for view.Mode:



### Public Member Functions

- `Mode ()`  
*Constructor for the player.*
- `void addButton (JButton x)`  
*adds buttons to a panel*
- `void addListener (ActionListener buttonListener)`  
*adds action listener to the buttons*
- `JButton getSingle ()`  
*gets the button for single mode*

### Private Attributes

- `JButton single = new JButton("Single Player Mode")`
- `JButton sObstacle = new JButton("Advanced Single Player Mode")`
- `JPanel buttonPanel`

### 7.5.1 Constructor & Destructor Documentation

#### 7.5.1.1 Mode()

```
view.Mode.Mode ( )
```

Constructor for the player.

sets the size and header for the window, and adds buttons to the window Setups for the frame

Setups for the buttons on the panel

Add the panel to the frame/window for display

### 7.5.2 Member Function Documentation

#### 7.5.2.1 addButton()

```
void view.Mode.addButton (
    JButton x )
```

adds buttons to a panel

makes buttons align in the panel

#### 7.5.2.2 addListener()

```
void view.Mode.addListener (
    ActionListener buttonListener )
```

adds action listener to the buttons

#### Parameters

<i>buttonListener</i>	is the action listener
-----------------------	------------------------

#### 7.5.2.3 `getSingle()`

```
JButton view.Mode.getSingle ( )
```

gets the button for single mode

#### Returns

single

### 7.5.3 Member Data Documentation

#### 7.5.3.1 `buttonPanel`

```
JPanel view.Mode.buttonPanel [private]
```

#### 7.5.3.2 `single`

```
JButton view.Mode.single = new JButton("Single Player Mode") [private]
```

Variable declarations for the buttons

- easy single mode
- single mode with obstacles
- a panel that contains the buttons

#### 7.5.3.3 `sObstacle`

```
JButton view.Mode.sObstacle = new JButton("Advanced Single Player Mode") [private]
```

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

- [src/view/Mode.java](#)



## 7.6 model.Paddle Class Reference

### Public Member Functions

- [Paddle](#) ()  
*Constructor for a paddle.*
- void [setPositionX](#) (int x)  
*sets the x-position of the paddle.*
- void [setPositionY](#) (int y)  
*sets the y-position of the paddle.*
- int [getPositionX](#) ()  
*returns the x position of the paddle.*
- int [getPositionY](#) ()  
*returns the y position of the paddle.*
- int [getWidth](#) ()  
*returns the width of the paddle.*
- int [getHeight](#) ()  
*returns the height of the paddle.*
- int [getInset](#) ()  
*returns the inset between the paddle and the screen.*

### Private Attributes

- int [positionX](#)
- int [positionY](#)
- final int [HEIGHT](#) = 10
- final int [WIDTH](#) = 80
- final int [INSET](#) = 10
- int [speed](#)

### 7.6.1 Constructor & Destructor Documentation

#### 7.6.1.1 Paddle()

```
model.Paddle.Paddle ( )
```

Constructor for a paddle.

Constructor initialize the starting position of a paddle.

### 7.6.2 Member Function Documentation

#### 7.6.2.1 getHeight()

```
int model.Paddle.getHeight ( )
```

returns the height of the paddle.

#### Returns

HEIGHT

#### 7.6.2.2 `getInset()`

```
int model.Paddle.getInset ( )
```

returns the inset between the paddle and the screen.

##### Returns

INSET

#### 7.6.2.3 `getPositionX()`

```
int model.Paddle.getPositionX ( )
```

returns the x position of the paddle.

##### Returns

positionX

#### 7.6.2.4 `getPositionY()`

```
int model.Paddle.getPositionY ( )
```

returns the y position of the paddle.

##### Returns

positionY

#### 7.6.2.5 `getWidth()`

```
int model.Paddle.getWidth ( )
```

returns the width of the paddle.

##### Returns

WIDTH

#### 7.6.2.6 `setPositionX()`

```
void model.Paddle.setPositionX (
    int x )
```

sets the x-position of the paddle.

#### Parameters

<code>x</code>	is the x position of the paddle.
----------------	----------------------------------

#### 7.6.2.7 setPositionY()

```
void model.Paddle.setPositionY (  
    int y )
```

sets the y-position of the paddle.

#### Parameters

<code>y</code>	is the y position of the paddle.
----------------	----------------------------------

### 7.6.3 Member Data Documentation

#### 7.6.3.1 HEIGHT

```
final int model.Paddle.HEIGHT = 10 [private]
```

The property of a paddle

- the length of a paddle
- the width of a paddle
- the inset between a paddle and the screen frame

#### 7.6.3.2 INSET

```
final int model.Paddle.INSET = 10 [private]
```

#### 7.6.3.3 positionX

```
int model.Paddle.positionX [private]
```

The position of a paddle

- horizontal position x
- vertical position y

#### 7.6.3.4 positionY

```
int model.Paddle.positionY [private]
```

### 7.6.3.5 speed

```
int model.Paddle.speed [private]
```

### 7.6.3.6 WIDTH

```
final int model.Paddle.WIDTH = 80 [private]
```

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

- [src/model/Paddle.java](#)

## 7.7 model.Player Class Reference

### Public Member Functions

- [Player \(\)](#)  
*Constructor for the player.*
- void [decrementLife \(\)](#)  
*loses score if the ball touches his/her border.*
- int [getScore \(\)](#)  
*gets the score of a player*
- boolean [checkLoss \(\)](#)  
*checks whether the player loses the game or not*

### Private Attributes

- final int [LIFE](#) = 3
- final int [NOLIFE](#) = 0
- int [score](#)

### 7.7.1 Constructor & Destructor Documentation

#### 7.7.1.1 Player()

```
model.Player.Player ( )
```

Constructor for the player.

sets the current life is the full life (3).

## 7.7.2 Member Function Documentation

### 7.7.2.1 checkLoss()

```
boolean model.Player.checkLoss ( )
```

checks whether the player loses the game or not

#### Returns

a boolean that is used to indicate whether the player is losing or not

### 7.7.2.2 decrementLife()

```
void model.Player.decrementLife ( )
```

loses score if the ball touches his/her border.

decreases the number of life by 1.

### 7.7.2.3 getScore()

```
int model.Player.getScore ( )
```

gets the score of a player

#### Returns

playerScore returns the score of the player .

## 7.7.3 Member Data Documentation

### 7.7.3.1 LIFE

```
final int model.Player.LIFE = 3 [private]
```

Defines constant number of life of a player

- the player has 3 lives in total
- the player loses if the number of life is 0

### 7.7.3.2 NOLIFE

```
final int model.Player.NOLIFE = 0 [private]
```

### 7.7.3.3 score

```
int model.Player.score [private]
```

Defines the current number of life of the player.

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

- [src/model/Player.java](#)

## 7.8 startGame.PongGame Class Reference

### Static Public Member Functions

- static void [main](#) (String[] args)  
*This is the main function for starting the program.*

### 7.8.1 Member Function Documentation

#### 7.8.1.1 main()

```
static void startGame.PongGame.main (  
    String [] args ) [static]
```

This is the main function for starting the program.

#### Author

Pongthusiastics

#### Parameters

<i>args</i>	is the input for the main function
-------------	------------------------------------

#### Date

13/11/2016

Initialize the model, view, and controller for the game

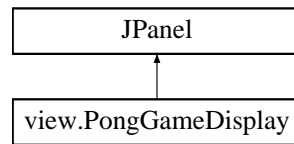
Invoke the game display from the controller

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

- [src/startGame/PongGame.java](#)

## 7.9 view.PongGameDisplay Class Reference

Inheritance diagram for view.PongGameDisplay:



### Public Member Functions

- [PongGameDisplay](#) ()  
*Constructor for [PongGameDisplay](#).*
- void [setBall](#) (int x, int y)  
*sets the positions of the ball*
- void [setBallSize](#) (int s)  
*sets the size of the ball*
- void [setBottom](#) (int x)  
*sets x-position for the player paddle*
- void [setTop](#) (int x)  
*sets x-position for the ai paddle*
- int [getBottomX](#) ()  
*gets the x-position of the player paddle*
- int [getBottomY](#) ()  
*gets the y-position of the player paddle*
- void [setTopScore](#) (int s)  
*sets the score for ai*
- void [setBottomScore](#) (int s)  
*sets the score for player*
- int [getBallX](#) ()  
*gets the x-position of the ball*
- int [getBallY](#) ()  
*gets the y-position of the ball*
- void [setPaddleWidth](#) (int w)  
*sets the width of the paddle*
- void [setPaddleHeight](#) (int h)  
*sets the height of the paddle*
- void [setInset](#) (int i)  
*sets the distance between frame and the paddle*
- void [setAdvance](#) ()  
*sets the game mode to be advanced*

### Protected Member Functions

- void [paintComponent](#) (Graphics g)  
*draws shapes on the screen*

## Private Attributes

- int [frameWidth](#)
- int [frameHeight](#)
- int [scoreTop](#)
- int [ballX](#)
- int [bottomPadX](#)
- int [topPadX](#)
- boolean [first](#)
- int [ballSize](#)
- int [padW](#)
- int [inset](#)
- int [gameMode](#)
- final int [SINGLE](#) =0
- final int [ADVANCE](#) =1

## 7.9.1 Constructor & Destructor Documentation

### 7.9.1.1 PongGameDisplay()

```
view.PongGameDisplay.PongGameDisplay ( )
```

Constructor for [PongGameDisplay](#).

Constructor by default set the game to single mode

## 7.9.2 Member Function Documentation

### 7.9.2.1 getBallX()

```
int view.PongGameDisplay.getBallX ( )
```

gets the x-position of the ball

#### Returns

ballX

### 7.9.2.2 getBallY()

```
int view.PongGameDisplay.getBallY ( )
```

gets the y-position of the ball

#### Returns

ballY



### 7.9.2.3 getBottomX()

```
int view.PongGameDisplay.getBottomX ( )
```

gets the x-position of the player paddle

#### Returns

bottomPadX

### 7.9.2.4 getBottomY()

```
int view.PongGameDisplay.getBottomY ( )
```

gets the y-position of the player paddle

#### Returns

bottomPadY

### 7.9.2.5 paintComponent()

```
void view.PongGameDisplay.paintComponent (
    Graphics g ) [protected]
```

draws shapes on the screen

when the game is started, by default draws the ball and paddles in the middle, otherwise, draws objects by passed in values. Initial positioning

- ball at the center of the screen
- paddle in the middle of the frame width

Draw rectangles by passed in values

Draw the ball by passed in values

Draw scores on the screen by passed in values

### 7.9.2.6 setAdvance()

```
void view.PongGameDisplay.setAdvance ( )
```

sets the game mode to be advanced

set the flag to advance

### 7.9.2.7 setBall()

```
void view.PongGameDisplay.setBall (
    int x,
    int y )
```

sets the positions of the ball

**Parameters**

<i>x</i>	is the x-position of the ball
<i>y</i>	is the y-position of the ball

**7.9.2.8 setBallSize()**

```
void view.PongGameDisplay.setBallSize (
    int s )
```

sets the size of the ball

**Parameters**

<i>s</i>	is the ball size
----------	------------------

**7.9.2.9 setBottom()**

```
void view.PongGameDisplay.setBottom (
    int x )
```

sets x-position for the player paddle

**Parameters**

<i>s</i>	is the x-position
----------	-------------------

**7.9.2.10 setBottomScore()**

```
void view.PongGameDisplay.setBottomScore (
    int s )
```

sets the score for player

**Parameters**

<i>s</i>	is the score
----------	--------------

**7.9.2.11 setInset()**

```
void view.PongGameDisplay.setInset (
    int i )
```

sets the distance between frame and the paddle

**Parameters**

<i>i</i>	is the inset
----------	--------------

**7.9.2.12 setPaddleHeight()**

```
void view.PongGameDisplay.setPaddleHeight (
    int h )
```

sets the height of the paddle

**Parameters**

<i>h</i>	is the height
----------	---------------

**7.9.2.13 setPaddleWidth()**

```
void view.PongGameDisplay.setPaddleWidth (
    int w )
```

sets the width of the paddle

**Parameters**

<i>w</i>	is the width
----------	--------------

**7.9.2.14 setTop()**

```
void view.PongGameDisplay.setTop (
    int x )
```

sets x-position for the ai paddle

**Parameters**

<i>s</i>	is the x-position
----------	-------------------

**7.9.2.15 setTopScore()**

```
void view.PongGameDisplay.setTopScore (
    int s )
```

sets the score for ai

**Parameters**

<b>s</b>	is the score
----------	--------------

### 7.9.3 Member Data Documentation

#### 7.9.3.1 ADVANCE

```
final int view.PongGameDisplay.ADVANCE =1 [private]
```

#### 7.9.3.2 ballSize

```
int view.PongGameDisplay.ballSize [private]
```

#### 7.9.3.3 ballX

```
int view.PongGameDisplay.ballX [private]
```

#### 7.9.3.4 bottomPadX

```
int view.PongGameDisplay.bottomPadX [private]
```

#### 7.9.3.5 first

```
boolean view.PongGameDisplay.first [private]
```

#### 7.9.3.6 frameHeight

```
int view.PongGameDisplay.frameHeight [private]
```

#### 7.9.3.7 frameWidth

```
int view.PongGameDisplay.frameWidth [private]
```

Variable declarations for the display

- frame dimension
- ball information
- bomb information
- player scores
- paddle information

### 7.9.3.8 gameMode

```
int view.PongGameDisplay.gameMode [private]
```

### 7.9.3.9 inset

```
int view.PongGameDisplay.inset [private]
```

### 7.9.3.10 padW

```
int view.PongGameDisplay.padW [private]
```

### 7.9.3.11 scoreTop

```
int view.PongGameDisplay.scoreTop [private]
```

### 7.9.3.12 SINGLE

```
final int view.PongGameDisplay.SINGLE =0 [private]
```

### 7.9.3.13 topPadX

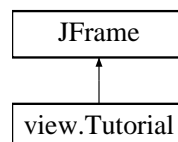
```
int view.PongGameDisplay.topPadX [private]
```

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

- [src/view/PongGameDisplay.java](#)

## 7.10 view.Tutorial Class Reference

Inheritance diagram for view.Tutorial:



### Public Member Functions

- [Tutorial](#) (ImageIcon img)  
*Constructor for the tutorial page.*
- JButton [getBack](#) ()  
*gets the button to exit the page*
- void [addListener](#) (ActionListener listener)  
*adds action listener to the button*

## Private Attributes

- JButton [back](#)

## 7.10.1 Constructor & Destructor Documentation

### 7.10.1.1 Tutorial()

```
view.Tutorial.Tutorial (
    ImageIcon img )
```

Constructor for the tutorial page.

#### Parameters

<i>img</i>	is the image for display
------------	--------------------------

Setups for the window

Add the image to the window

## 7.10.2 Member Function Documentation

### 7.10.2.1 addListener()

```
void view.Tutorial.addListener (
    ActionListener listener )
```

adds action listener to the button

#### Parameters

<i>listener</i>	is the action listener
-----------------	------------------------

### 7.10.2.2 getBack()

```
JButton view.Tutorial.getBack ( )
```

gets the button to exit the page

#### Returns

back is the button for going back to welcome page

### 7.10.3 Member Data Documentation

#### 7.10.3.1 back

```
JButton view.Tutorial.back [private]
```

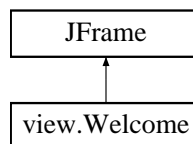
Variable declaration for the back button

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

- [src/view/Tutorial.java](#)

## 7.11 view.Welcome Class Reference

Inheritance diagram for view.Welcome:



### Public Member Functions

- [Welcome](#) ()  
*Constructor for welcome page.*
- JButton [getStart](#) ()  
*gets the start button*
- JButton [load](#) ()  
*gets the load button*
- JButton [highScores](#) ()  
*gets the button to display high score*
- JButton [tutorial](#) ()  
*gets the button to display instructions*
- JButton [exit](#) ()  
*gets the button to exit the program*
- void [addButton](#) (JButton x)  
*adds buttons to a panel*
- void [addListener](#) (ActionListener buttonListener)  
*adds action listener to the buttons*

### Private Attributes

- JButton [start](#) = new JButton("Start New Game")
- JButton [load](#) = new JButton("Load Game")
- JButton [highScores](#) = new JButton("High Scores")
- JButton [tutorial](#) = new JButton("Tutorial")
- JButton [exit](#) = new JButton("Exit")
- JPanel [buttonPanel](#)

## 7.11.1 Constructor & Destructor Documentation

### 7.11.1.1 Welcome()

```
view.Welcome.Welcome ( )
```

Constructor for welcome page.

sets the header and size of window, and add buttons to it.

- Set the header of the window
- Set the size of the window

Add buttons on the window

## 7.11.2 Member Function Documentation

### 7.11.2.1 addButton()

```
void view.Welcome.addButton (
    JButton x )
```

adds buttons to a panel

makes buttons align in the panel

### 7.11.2.2 addListener()

```
void view.Welcome.addListener (
    ActionListener buttonListener )
```

adds action listener to the buttons

#### Parameters

<i>buttonListener</i>	is the action listener
-----------------------	------------------------

### 7.11.2.3 exit()

```
JButton view.Welcome.exit ( )
```

gets the button to exit the program

#### Returns

exit



#### 7.11.2.4 getStart()

```
JButton view.Welcome.getStart ( )
```

gets the start button

##### Returns

start indicates to start a new game

#### 7.11.2.5 highScores()

```
JButton view.Welcome.highScores ( )
```

gets the button to display high score

##### Returns

highScores

#### 7.11.2.6 load()

```
JButton view.Welcome.load ( )
```

gets the load button

##### Returns

load indicates to load a new game

#### 7.11.2.7 tutorial()

```
JButton view.Welcome.tutorial ( )
```

gets the button to display instructions

##### Returns

tutorial

### 7.11.3 Member Data Documentation

#### 7.11.3.1 buttonPanel

```
JPanel view.Welcome.buttonPanel [private]
```

Define a panel for the arrangement of buttons

#### 7.11.3.2 exit

```
JButton view.Welcome.exit = new JButton("Exit")    [private]
```

#### 7.11.3.3 highScores

```
JButton view.Welcome.highScores = new JButton("High Scores")    [private]
```

#### 7.11.3.4 load

```
JButton view.Welcome.load = new JButton("Load Game")    [private]
```

#### 7.11.3.5 start

```
JButton view.Welcome.start = new JButton("Start New Game")    [private]
```

Variable declarations for the page

- start a new game
- load the previous game
- display high score
- tutorial
- exit the game

#### 7.11.3.6 tutorial

```
JButton view.Welcome.tutorial = new JButton("Tutorial")    [private]
```

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

- src/view/[Welcome.java](#)

## Chapter 8

# File Documentation

### 8.1 src/model/Ball.java File Reference

This class represents a ball on the pong game.

#### Classes

- class [model.Ball](#)

#### Packages

- package [model](#)

#### 8.1.1 Detailed Description

This class represents a ball on the pong game.

Ball

#### Author

Pongthusiastics

#### Date

13/11/2016

This class saves the information of a ball, including its position, size and the speed.

### 8.2 src/model/GameModel.java File Reference

This class represents a ball on the pong game.

## Classes

- class [model.GameModel](#)

## Packages

- package [model](#)

### 8.2.1 Detailed Description

This class represents a ball on the pong game.

GameModel

#### Author

Pongthusiastics

#### Date

13/11/2016

This class saves the information of a ball, including its position, size and the speed.

## 8.3 [src/model/Paddle.java](#) File Reference

This class defines a paddle.

## Classes

- class [model.Paddle](#)

## Packages

- package [model](#)

### 8.3.1 Detailed Description

This class defines a paddle.

Paddle

#### Author

Pongthusiastics

#### Date

13/11/2016

This class saves the information of a paddle, including its position, height, width, and inset between the paddle and the screen.

## 8.4 src/model/Player.java File Reference

This class represents a player for the game.

### Classes

- class [model.Player](#)

### Packages

- package [model](#)

### 8.4.1 Detailed Description

This class represents a player for the game.

Player

Author

Pongthusiastics

Date

13/11/2016

This class contains the information for a player, including number of life and his/her current score.

## 8.5 src/startGame/GameController.java File Reference

This class is the controller for the game.

### Classes

- class [startGame.GameController](#)
- class **startGame.GameController.WelcomepageListener**  
*action listener for the welcome page*
- class **startGame.GameController.ModeListener**  
*action listener for the game mode page*
- class **startGame.GameController.TutorialListener**  
*action listener for the tutorial page*
- class **startGame.GameController.GameListener**  
*action listener for the game page*

## Packages

- package [startGame](#)

### 8.5.1 Detailed Description

This class is the controller for the game.

GameController

Author

Pongthusiastics

Date

13/11/2016

This class cooperates with model and view and give direction to the game.

## 8.6 src/startGame/PongGame.java File Reference

This class starts the game.

## Classes

- class [startGame.PongGame](#)

## Packages

- package [startGame](#)

### 8.6.1 Detailed Description

This class starts the game.

PongGame

Author

Pongthusiastics

Date

13/11/2016

This class instantiates a model, view, and controller using the MVC model, and starts the game.

```
GameView view = new GameView();
GameModel model = new GameModel();
GameController controller = new GameController(view, model);
controller.display();
```

## 8.7 src/view/MapView.java File Reference

This class is the main view model.

### Classes

- class [view.MapView](#)

### Packages

- package [view](#)

### 8.7.1 Detailed Description

This class is the main view model.

MapView

#### Author

Pongthusiastics

#### Date

13/11/2016

This class import all different windows for display.

## 8.8 src/view/Mode.java File Reference

This class create the game mode window.

### Classes

- class [view.Mode](#)

### Packages

- package [view](#)

### 8.8.1 Detailed Description

This class create the game mode window.

Mode

Author

Pongthusiastics

Date

13/11/2016

This class create a frame and buttons for different game level

## 8.9 src/view/PongGameDisplay.java File Reference

This class construct the view of the pong game.

### Classes

- class [view.PongGameDisplay](#)

### Packages

- package [view](#)

### 8.9.1 Detailed Description

This class construct the view of the pong game.

PongGameDisplay

Author

Pongthusiastics

Date

13/11/2016

This class gets data from controller and display them on the screen



## 8.10 src/view/Tutorial.java File Reference

This class create the tutorial window.

### Classes

- class [view.Tutorial](#)

### Packages

- package [view](#)

#### 8.10.1 Detailed Description

This class create the tutorial window.

Tutorial

Author

Pongthusiastics

Date

13/11/2016

This class display instruction for the game

## 8.11 src/view/Welcome.java File Reference

This class creates the display for welcome page.

### Classes

- class [view.Welcome](#)

### Packages

- package [view](#)

#### 8.11.1 Detailed Description

This class creates the display for welcome page.

Welcome

Author

Pongthusiastics

Date

13/11/2016

This class defines buttons for options in the welcome page.



# Index

## ADVANCE

- startGame::GameController, [17](#)
- view::PongGameDisplay, [42](#)

## addButton

- view::Mode, [29](#)
- view::Welcome, [46](#)

## addListener

- view::Mode, [29](#)
- view::Tutorial, [44](#)
- view::Welcome, [46](#)

## ai

- startGame::GameController, [17](#)

## b

- model::GameModel, [23](#)
- startGame::GameController, [17](#)

## back

- view::Tutorial, [45](#)

## Ball

- model::Ball, [13](#)

## ballSize

- view::PongGameDisplay, [42](#)

## ballX

- startGame::GameController, [18](#)
- view::PongGameDisplay, [42](#)

## bomb

- startGame::GameController, [18](#)

## bombX

- startGame::GameController, [18](#)

## bottomPadX

- startGame::GameController, [18](#)
- view::PongGameDisplay, [42](#)

## buttonPanel

- view::Mode, [30](#)
- view::Welcome, [47](#)

## cannotLoadMessage

- view::GameView, [25](#)

## checkGameOver

- startGame::GameController, [17](#)

## checkLoss

- model::Player, [35](#)

## createGame

- view::GameView, [25](#)

## decrementLife

- model::Player, [35](#)

## display

- startGame::GameController, [17](#)
- view::GameView, [25](#)

## exit

- view::Welcome, [46, 47](#)

## FRAMEHEIGHT

- view::GameView, [28](#)

## FRAMEWIDTH

- view::GameView, [28](#)

## first

- view::PongGameDisplay, [42](#)

## frameHeight

- view::PongGameDisplay, [42](#)

## frameWidth

- startGame::GameController, [18](#)
- view::PongGameDisplay, [42](#)

## GameController

- startGame::GameController, [16](#)

## gameDisplay

- startGame::GameController, [18](#)

## gameFrame

- startGame::GameController, [18](#)
- view::GameView, [28](#)

## gameMode

- startGame::GameController, [18](#)
- view::PongGameDisplay, [42](#)

## GameModel

- model::GameModel, [21](#)

## gameOver

- view::GameView, [25](#)

## GameView

- view::GameView, [25](#)

## getBack

- view::Tutorial, [44](#)

## getBall

- model::GameModel, [22](#)

## getBallX

- view::PongGameDisplay, [38](#)

## getBallY

- view::PongGameDisplay, [38](#)

## getBomb

- model::GameModel, [22](#)

## getBottomX

- view::PongGameDisplay, [38](#)

## getBottomY

- view::PongGameDisplay, [39](#)

## getComputer

- model::GameModel, [22](#)

## getComputerPaddle

- model::GameModel, [22](#)

## getFrameHeight

- view::GameView, 26
- getFrameWidth
  - view::GameView, 26
- getGame
  - view::GameView, 26
- getGameFrame
  - view::GameView, 26
- getHeight
  - model::Paddle, 31
- getInset
  - model::Paddle, 31
- getPlayer
  - model::GameModel, 22
- getPlayerPaddle
  - model::GameModel, 22
- getPositionX
  - model::Ball, 14
  - model::Paddle, 32
- getPositionY
  - model::Ball, 14
  - model::Paddle, 32
- getScore
  - model::Player, 35
- getSingle
  - view::Mode, 30
- getSize
  - model::Ball, 14
- getStart
  - view::Welcome, 46
- getTutorial
  - view::GameView, 27
- getWelcome
  - view::GameView, 27
- getWidth
  - model::Paddle, 32
- getmode
  - view::GameView, 26
- HEIGHT
  - model::Paddle, 33
- highScores
  - view::Welcome, 47, 48
- INSET
  - model::Paddle, 33
- inset
  - startGame::GameController, 19
  - view::PongGameDisplay, 43
- keys
  - startGame::GameController, 19
- LIFE
  - model::Player, 35
- load
  - view::Welcome, 47, 48
- m
  - startGame::GameController, 19
- main
  - startGame::PongGame, 36
- Mode
  - view::Mode, 29
- mode
  - startGame::GameController, 19
  - view::GameView, 28
- model, 11
- model.Ball, 13
- model.GameModel, 21
- model.Paddle, 31
- model.Player, 34
- model::Ball
  - Ball, 13
  - getPositionX, 14
  - getPositionY, 14
  - getSize, 14
  - positionX, 15
  - positionY, 15
  - SIZE, 15
  - setPositionX, 14
  - setPositionY, 14
  - speed, 15
- model::GameModel
  - b, 23
  - GameModel, 21
  - getBall, 22
  - getBomb, 22
  - getComputer, 22
  - getComputerPaddle, 22
  - getPlayer, 22
  - getPlayerPaddle, 22
  - p\_player, 23
  - player, 23
  - setBall, 23
  - setBomb, 23
- model::Paddle
  - getHeight, 31
  - getInset, 31
  - getPositionX, 32
  - getPositionY, 32
  - getWidth, 32
  - HEIGHT, 33
  - INSET, 33
  - Paddle, 31
  - positionX, 33
  - positionY, 33
  - setPositionX, 32
  - setPositionY, 33
  - speed, 33
  - WIDTH, 34
- model::Player
  - checkLoss, 35
  - decrementLife, 35
  - getScore, 35
  - LIFE, 35
  - NOLIFE, 35
  - Player, 34

- score, [35](#)
- NOLIFE
  - model::Player, [35](#)
- noFileAvailMessage
  - view::GameView, [27](#)
- p\_player
  - model::GameModel, [23](#)
- padWidth
  - startGame::GameController, [19](#)
- Paddle
  - model::Paddle, [31](#)
- paddle\_player
  - startGame::GameController, [19](#)
- padW
  - view::PongGameDisplay, [43](#)
- paintComponent
  - view::PongGameDisplay, [39](#)
- Player
  - model::Player, [34](#)
- player
  - model::GameModel, [23](#)
  - startGame::GameController, [19](#)
- PongGameDisplay
  - view::PongGameDisplay, [38](#)
- ponggame
  - view::GameView, [28](#)
- positionX
  - model::Ball, [15](#)
  - model::Paddle, [33](#)
- positionY
  - model::Ball, [15](#)
  - model::Paddle, [33](#)
- record
  - startGame::GameController, [19](#)
- SINGLE
  - startGame::GameController, [19](#)
  - view::PongGameDisplay, [43](#)
- SIZE
  - model::Ball, [15](#)
- sObstacle
  - view::Mode, [30](#)
- score
  - model::Player, [35](#)
- scoreTop
  - startGame::GameController, [19](#)
  - view::PongGameDisplay, [43](#)
- setAdvance
  - view::PongGameDisplay, [39](#)
- setBall
  - model::GameModel, [23](#)
  - view::PongGameDisplay, [39](#)
- setBallSize
  - view::PongGameDisplay, [40](#)
- setBomb
  - model::GameModel, [23](#)
- setBottom
  - view::PongGameDisplay, [40](#)
- setBottomScore
  - view::PongGameDisplay, [40](#)
- setInset
  - view::PongGameDisplay, [40](#)
- setPaddleHeight
  - view::PongGameDisplay, [41](#)
- setPaddleWidth
  - view::PongGameDisplay, [41](#)
- setPositionX
  - model::Ball, [14](#)
  - model::Paddle, [32](#)
- setPositionY
  - model::Ball, [14](#)
  - model::Paddle, [33](#)
- setTop
  - view::PongGameDisplay, [41](#)
- setTopScore
  - view::PongGameDisplay, [41](#)
- single
  - view::Mode, [30](#)
- speed
  - model::Ball, [15](#)
  - model::Paddle, [33](#)
- src/model/Ball.java, [49](#)
- src/model/GameModel.java, [49](#)
- src/model/Paddle.java, [50](#)
- src/model/Player.java, [51](#)
- src/startGame/GameController.java, [51](#)
- src/startGame/PongGame.java, [52](#)
- src/view/GameView.java, [53](#)
- src/view/Mode.java, [53](#)
- src/view/PongGameDisplay.java, [54](#)
- src/view/Tutorial.java, [55](#)
- src/view/Welcome.java, [55](#)
- start
  - view::Welcome, [48](#)
- startGame, [11](#)
- startGame.GameController, [15](#)
- startGame.PongGame, [36](#)
- startGame::GameController
  - ADVANCE, [17](#)
  - ai, [17](#)
  - b, [17](#)
  - ballX, [18](#)
  - bomb, [18](#)
  - bombX, [18](#)
  - bottomPadX, [18](#)
  - checkGameOver, [17](#)
  - display, [17](#)
  - frameWidth, [18](#)
  - GameController, [16](#)
  - gameDisplay, [18](#)
  - gameFrame, [18](#)
  - gameMode, [18](#)
  - inset, [19](#)
  - keys, [19](#)

- m, [19](#)
  - mode, [19](#)
  - padWidth, [19](#)
  - paddle\_player, [19](#)
  - player, [19](#)
  - record, [19](#)
  - SINGLE, [19](#)
  - scoreTop, [19](#)
  - t, [20](#)
  - tut, [20](#)
  - v, [20](#)
  - velX, [20](#)
  - w, [20](#)
- startGame::PongGame
  - main, [36](#)
- t
  - startGame::GameController, [20](#)
- topPadX
  - view::PongGameDisplay, [43](#)
- tut
  - startGame::GameController, [20](#)
- Tutorial
  - view::Tutorial, [44](#)
- tutorial
  - view::GameView, [28](#)
  - view::Welcome, [47, 48](#)
- tutorialPage
  - view::GameView, [27](#)
- v
  - startGame::GameController, [20](#)
- velX
  - startGame::GameController, [20](#)
- view, [11](#)
- view.GameView, [24](#)
- view.Mode, [29](#)
- view.PongGameDisplay, [37](#)
- view.Tutorial, [43](#)
- view.Welcome, [45](#)
- view::GameView
  - cannotLoadMessage, [25](#)
  - createGame, [25](#)
  - display, [25](#)
  - FRAMEHEIGHT, [28](#)
  - FRAMEWIDTH, [28](#)
  - gameFrame, [28](#)
  - gameOver, [25](#)
  - GameView, [25](#)
  - getFrameHeight, [26](#)
  - getFrameWidth, [26](#)
  - getGame, [26](#)
  - getGameFrame, [26](#)
  - getTutorial, [27](#)
  - getWelcome, [27](#)
  - getmode, [26](#)
  - mode, [28](#)
  - noFileAvailMessage, [27](#)
  - ponggame, [28](#)
  - tutorial, [28](#)
  - tutorialPage, [27](#)
  - welcome, [28](#)
- view::Mode
  - addButton, [29](#)
  - addListener, [29](#)
  - buttonPanel, [30](#)
  - getSingle, [30](#)
  - Mode, [29](#)
  - sObstacle, [30](#)
  - single, [30](#)
- view::PongGameDisplay
  - ADVANCE, [42](#)
  - ballSize, [42](#)
  - ballX, [42](#)
  - bottomPadX, [42](#)
  - first, [42](#)
  - frameHeight, [42](#)
  - frameWidth, [42](#)
  - gameMode, [42](#)
  - getBallX, [38](#)
  - getBallY, [38](#)
  - getBottomX, [38](#)
  - getBottomY, [39](#)
  - inset, [43](#)
  - padW, [43](#)
  - paintComponent, [39](#)
  - PongGameDisplay, [38](#)
  - SINGLE, [43](#)
  - scoreTop, [43](#)
  - setAdvance, [39](#)
  - setBall, [39](#)
  - setBallSize, [40](#)
  - setBottom, [40](#)
  - setBottomScore, [40](#)
  - setInset, [40](#)
  - setPaddleHeight, [41](#)
  - setPaddleWidth, [41](#)
  - setTop, [41](#)
  - setTopScore, [41](#)
  - topPadX, [43](#)
- view::Tutorial
  - addListener, [44](#)
  - back, [45](#)
  - getBack, [44](#)
  - Tutorial, [44](#)
- view::Welcome
  - addButton, [46](#)
  - addListener, [46](#)
  - buttonPanel, [47](#)
  - exit, [46, 47](#)
  - getStart, [46](#)
  - highScores, [47, 48](#)
  - load, [47, 48](#)
  - start, [48](#)
  - tutorial, [47, 48](#)
  - Welcome, [46](#)
- w

---

- startGame::GameController, [20](#)
- WIDTH
  - model::Paddle, [34](#)
- Welcome
  - view::Welcome, [46](#)
- welcome
  - view::GameView, [28](#)