

FaultInOurPong

1

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

Generated by Doxygen 1.8.12

Contents

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

7	Class Documentation	13
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()	15
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
7.2	startGame.GameController Class Reference	15
7.2.1	Constructor & Destructor Documentation	17
7.2.1.1	GameController()	17
7.2.2	Member Function Documentation	18
7.2.2.1	checkGameOver()	18
7.2.2.2	display()	18
7.2.2.3	getbombVelX()	18
7.2.2.4	getbombVelY()	19
7.2.2.5	getElapsedTime()	19
7.2.2.6	getVelX()	19
7.2.2.7	getVelY()	19
7.2.2.8	resetGame()	19
7.2.2.9	setSpeed()	20
7.2.3	Member Data Documentation	20
7.2.3.1	ADVANCE	20
7.2.3.2	ai	20

7.2.3.3	b	20
7.2.3.4	ballX	20
7.2.3.5	bomb	20
7.2.3.6	bombVelX	20
7.2.3.7	bombX	20
7.2.3.8	bottomPadX	20
7.2.3.9	displayScore	21
7.2.3.10	endTime	21
7.2.3.11	exit	21
7.2.3.12	frameWidth	21
7.2.3.13	gameDisplay	21
7.2.3.14	gameFrame	21
7.2.3.15	gameMode	21
7.2.3.16	inset	21
7.2.3.17	keys	22
7.2.3.18	m	22
7.2.3.19	mode	22
7.2.3.20	paddle_player	22
7.2.3.21	padWidth	22
7.2.3.22	pause	22
7.2.3.23	player	22
7.2.3.24	resume	22
7.2.3.25	save	22
7.2.3.26	scoreTop	22
7.2.3.27	SINGLE	23
7.2.3.28	startTime	23
7.2.3.29	t	23
7.2.3.30	timeElapsed	23
7.2.3.31	tut	23
7.2.3.32	v	23

7.2.3.33	velX	23
7.2.3.34	w	23
7.3	model.GameModel Class Reference	24
7.3.1	Constructor & Destructor Documentation	24
7.3.1.1	GameModel()	24
7.3.2	Member Function Documentation	25
7.3.2.1	getBall()	25
7.3.2.2	getBomb()	25
7.3.2.3	getComputer()	25
7.3.2.4	getComputerPaddle()	25
7.3.2.5	getPlayer()	25
7.3.2.6	getPlayerPaddle()	26
7.3.2.7	setBall()	26
7.3.2.8	setBomb()	26
7.3.3	Member Data Documentation	26
7.3.3.1	b	26
7.3.3.2	p_player	27
7.3.3.3	player	27
7.4	view.GameView Class Reference	27
7.4.1	Constructor & Destructor Documentation	28
7.4.1.1	GameView()	28
7.4.2	Member Function Documentation	28
7.4.2.1	addButton()	28
7.4.2.2	cannotLoadMessage()	28
7.4.2.3	createGame()	29
7.4.2.4	display()	29
7.4.2.5	gameOver()	29
7.4.2.6	getExit()	29
7.4.2.7	getFrameHeight()	29
7.4.2.8	getFrameWidth()	30

7.4.2.9	getGame()	30
7.4.2.10	getGameFrame()	30
7.4.2.11	getGameOptionPanel()	30
7.4.2.12	getmode()	30
7.4.2.13	getPause()	31
7.4.2.14	getResume()	31
7.4.2.15	getSave()	31
7.4.2.16	getTutorial()	31
7.4.2.17	getWelcome()	31
7.4.2.18	noFileAvailMessage()	31
7.4.2.19	tutorialPage()	31
7.4.3	Member Data Documentation	32
7.4.3.1	exit	32
7.4.3.2	FRAMEHEIGHT	32
7.4.3.3	FRAMEWIDTH	32
7.4.3.4	gameFrame	32
7.4.3.5	gameOptions	32
7.4.3.6	mode	32
7.4.3.7	pause	32
7.4.3.8	ponggame	32
7.4.3.9	resume	33
7.4.3.10	save	33
7.4.3.11	tutorial	33
7.4.3.12	welcome	33
7.5	view.HighScore Class Reference	33
7.5.1	Constructor & Destructor Documentation	34
7.5.1.1	HighScore()	34
7.5.2	Member Function Documentation	34
7.5.2.1	checkHighScore()	34
7.5.2.2	findRank()	34

7.5.2.3	highScorePage()	35
7.5.2.4	isHigh()	35
7.5.2.5	readFrom()	35
7.5.2.6	writeTo()	36
7.6	view.Mode Class Reference	36
7.6.1	Constructor & Destructor Documentation	37
7.6.1.1	Mode()	37
7.6.2	Member Function Documentation	37
7.6.2.1	addButton()	37
7.6.2.2	addListener()	37
7.6.2.3	getAdvance()	37
7.6.2.4	getSingle()	38
7.6.3	Member Data Documentation	38
7.6.3.1	buttonPanel	38
7.6.3.2	single	38
7.6.3.3	sObstacle	38
7.7	model.Paddle Class Reference	38
7.7.1	Constructor & Destructor Documentation	39
7.7.1.1	Paddle()	39
7.7.2	Member Function Documentation	39
7.7.2.1	getHeight()	39
7.7.2.2	getInset()	39
7.7.2.3	getPositionX()	39
7.7.2.4	getPositionY()	40
7.7.2.5	getWidth()	40
7.7.2.6	setPositionX()	40
7.7.2.7	setPositionY()	40
7.7.3	Member Data Documentation	41
7.7.3.1	HEIGHT	41
7.7.3.2	INSET	41

7.7.3.3	positionX	41
7.7.3.4	positionY	41
7.7.3.5	speed	41
7.7.3.6	WIDTH	41
7.8	model.Player Class Reference	42
7.8.1	Constructor & Destructor Documentation	42
7.8.1.1	Player()	42
7.8.2	Member Function Documentation	42
7.8.2.1	checkLoss()	42
7.8.2.2	decrementLife()	43
7.8.2.3	getScore()	43
7.8.2.4	resetScore()	43
7.8.2.5	setScore()	43
7.8.3	Member Data Documentation	43
7.8.3.1	LIFE	43
7.8.3.2	NOLIFE	44
7.8.3.3	score	44
7.9	startGame.PongGame Class Reference	44
7.9.1	Member Function Documentation	44
7.9.1.1	main()	44
7.10	view.PongGameDisplay Class Reference	45
7.10.1	Constructor & Destructor Documentation	47
7.10.1.1	PongGameDisplay()	47
7.10.2	Member Function Documentation	47
7.10.2.1	bombTime()	47
7.10.2.2	getBallSize()	47
7.10.2.3	getBallX()	47
7.10.2.4	getBallY()	47
7.10.2.5	getBombX()	48
7.10.2.6	getBombY()	48

7.10.2.7	<code>getBottomScore()</code>	48
7.10.2.8	<code>getBottomX()</code>	48
7.10.2.9	<code>getBottomY()</code>	48
7.10.2.10	<code>getMode()</code>	49
7.10.2.11	<code>getTopScore()</code>	49
7.10.2.12	<code>noBomb()</code>	49
7.10.2.13	<code>paintComponent()</code>	49
7.10.2.14	<code>setAdvance()</code>	49
7.10.2.15	<code>setBall()</code>	49
7.10.2.16	<code>setBallSize()</code>	50
7.10.2.17	<code>setBomb()</code>	50
7.10.2.18	<code>setBottom()</code>	50
7.10.2.19	<code>setBottomPaddle()</code>	51
7.10.2.20	<code>setBottomScore()</code>	51
7.10.2.21	<code>setInset()</code>	51
7.10.2.22	<code>setPaddleHeight()</code>	52
7.10.2.23	<code>setPaddleWidth()</code>	52
7.10.2.24	<code>setTop()</code>	52
7.10.2.25	<code>setTopPaddle()</code>	53
7.10.2.26	<code>setTopScore()</code>	53
7.10.2.27	<code>timeForBomb()</code>	53
7.10.3	Member Data Documentation	54
7.10.3.1	<code>ADVANCE</code>	54
7.10.3.2	<code>ballSize</code>	54
7.10.3.3	<code>ballX</code>	54
7.10.3.4	<code>bombX</code>	54
7.10.3.5	<code>bottomPadX</code>	54
7.10.3.6	<code>first</code>	54
7.10.3.7	<code>frameHeight</code>	54
7.10.3.8	<code>frameWidth</code>	54

7.10.3.9	gameMode	55
7.10.3.10	inset	55
7.10.3.11	padW	55
7.10.3.12	scoreTop	55
7.10.3.13	SINGLE	55
7.10.3.14	startBomb	55
7.10.3.15	topPadX	55
7.11	view.Tutorial Class Reference	55
7.11.1	Constructor & Destructor Documentation	56
7.11.1.1	Tutorial()	56
7.11.2	Member Function Documentation	56
7.11.2.1	addListener()	56
7.11.2.2	getBack()	56
7.11.3	Member Data Documentation	57
7.11.3.1	back	57
7.12	view.Welcome Class Reference	57
7.12.1	Constructor & Destructor Documentation	58
7.12.1.1	Welcome()	58
7.12.2	Member Function Documentation	58
7.12.2.1	addButton()	58
7.12.2.2	addListener()	58
7.12.2.3	exit()	59
7.12.2.4	getStart()	59
7.12.2.5	highScores()	59
7.12.2.6	load()	59
7.12.2.7	tutorial()	59
7.12.3	Member Data Documentation	60
7.12.3.1	buttonPanel	60
7.12.3.2	exit	60
7.12.3.3	highScores	60
7.12.3.4	load	60
7.12.3.5	start	60
7.12.3.6	tutorial	60

8 File Documentation	61
8.1 src/model/Ball.java File Reference	61
8.1.1 Detailed Description	61
8.2 src/model/GameModel.java File Reference	61
8.2.1 Detailed Description	62
8.3 src/model/Paddle.java File Reference	62
8.3.1 Detailed Description	62
8.4 src/model/Player.java File Reference	63
8.4.1 Detailed Description	63
8.5 src/startGame/GameController.java File Reference	63
8.5.1 Detailed Description	64
8.6 src/startGame/PongGame.java File Reference	64
8.6.1 Detailed Description	64
8.7 src/view/MapView.java File Reference	65
8.7.1 Detailed Description	65
8.8 src/view/HighScore.java File Reference	65
8.8.1 Detailed Description	66
8.9 src/view/Mode.java File Reference	66
8.9.1 Detailed Description	66
8.10 src/view/PongGameDisplay.java File Reference	67
8.10.1 Detailed Description	67
8.11 src/view/Tutorial.java File Reference	67
8.11.1 Detailed Description	68
8.12 src/view/Welcome.java File Reference	68
8.12.1 Detailed Description	68
Index	69

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):

model	11
startGame	11
view	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	24
view.GameView	27
view.HighScore	33
model.Paddle	38
model.Player	42
startGame.PongGame	44
JFrame	
view.Mode	36
view.Tutorial	55
view.Welcome	57
JPanel	
view.PongGameDisplay	45

Chapter 4

Class Index

4.1 Class List

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

model.Ball	13
startGame.GameController	15
model.GameModel	24
view.GameView	27
view.HighScore	33
view.Mode	36
model.Paddle	38
model.Player	42
startGame.PongGame	44
view.PongGameDisplay	45
view.Tutorial	55
view.Welcome	57

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

src/model/ Ball.java	
This class represents a ball on the pong game	61
src/model/ GameModel.java	
This class represents a ball on the pong game	61
src/model/ Paddle.java	
This class defines a paddle	62
src/model/ Player.java	
This class represents a player for the game	63
src/startGame/ GameController.java	
This class is the controller for the game	63
src/startGame/ PongGame.java	
This class starts the game	64
src/view/ GameView.java	
This class is the main view model	65
src/view/ HighScore.java	
This class displays score data	65
src/view/ Mode.java	
This class create the game mode window	66
src/view/ PongGameDisplay.java	
This class construct the view of the pong game	67
src/view/ Tutorial.java	
This class create the tutorial window	67
src/view/ Welcome.java	
This class creates the display for welcome page	68

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 [HighScore](#)
- 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) throws ArithmeticException
sets the x position of the ball
- void [setPositionY](#) (int y) throws ArithmeticException
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

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 ) throws ArithmeticException
```

sets the x position of the ball

Parameters

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

Exceptions

<i>ArithmeticException</i>	ball x-position could not be set out of the game frame.
----------------------------	---

7.1.2.5 setPositionY()

```
void model.Ball.setPositionY (
    int y ) throws ArithmeticException
```

sets the y position of the ball

Parameters

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

Exceptions

<i>ArithmeticException</i>	ball y-position could not be set out of the game frame.
----------------------------	---

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

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](#)) throws `ArithmeticException`
this is the constructor for the controller
- void [display](#) ()
sets the display
- int [getVelX](#) ()
returns the velocity of ball in the x direction.
- int [getVelY](#) ()
returns the velocity of ball in the y direction.
- int [getbombVelX](#) ()
returns the velocity of bomb in the x direction.
- int [getbombVelY](#) ()
returns the velocity of bomb in the y direction.

Private Member Functions

- void [checkGameOver](#) ()
checks whether the game ends
- void [getElapsedTime](#) ()
obtains the time the user plays.
- void [resetGame](#) ()
resets to initial when the player exits a game.
- void [setSpeed](#) ()
resets speed to ball and bomb

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`
- int `bombVelX`

- [Player](#) `player`
- [Player](#) `ai`
- `Timer` `t`
- [HighScore](#) `displayScore`
- `long` `startTime`
- `long` `endTime`
- `double` `timeElapsed`
- `JButton` `pause`
- `JButton` `resume`
- `JButton` `save`
- `JButton` `exit`

7.2.1 Constructor & Destructor Documentation

7.2.1.1 GameController()

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

this is the constructor for the controller

contains methods for detecting environment variables and passing variables between model and view.

Parameters

<code>v</code>	is the view framework
<code>m</code>	is the model framework

Exceptions

<i>ArithmeticException</i>	object position could not be out of the game frame
--	--

Set default game mode to be single

Set up velocities

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

Initialize the start time and end time for a player

7.2.2 Member Function Documentation

7.2.2.1 checkGameOver()

```
void startGame.GameController.checkGameOver ( ) [private]
```

checks whether the game ends

if the score of one side is 0, redirect to the ending state (stop the game and display message).

Exceptions

<i>IOException</i>	cannot find and read the highScore.txt
--------------------	--

- 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.
- Calculate the time a player has played, if breaks the record, save the record.

7.2.2.2 display()

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

sets the display

opens a window

7.2.2.3 getbombVelX()

```
int startGame.GameController.getbombVelX ( )
```

returns the velocity of bomb in the x direction.

Returns

bombVelX

7.2.2.4 getbombVelY()

```
int startGame.GameController.getbombVelY ( )
```

returns the velocity of bomb in the y direction.

Returns

bombVelY

7.2.2.5 getElapsedTime()

```
void startGame.GameController.getElapsedTime ( ) [private]
```

obtains the time the user plays.

calculates the time elapsed and save it into a variable.

7.2.2.6 getVelX()

```
int startGame.GameController.getVelX ( )
```

returns the velocity of ball in the x direction.

Returns

velX

7.2.2.7 getVelY()

```
int startGame.GameController.getVelY ( )
```

returns the velocity of ball in the y direction.

Returns

velY

7.2.2.8 resetGame()

```
void startGame.GameController.resetGame ( ) [private]
```

resets to initial when the player exits a game.

re-initializes the variables in the game model and update the variables in view and model. Reset player scores/lives in the model.

Reset the game mode

Reset ball and bomb position

Reset the ball and bomb speed

Re-obtain scores/lives from the model

Reset scores in the view

7.2.2.9 setSpeed()

```
void startGame.GameController.setSpeed ( ) [private]
```

resets speed to ball and bomb

re-initialize by randomized values Randomize the velocity for bomb

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 bombVelX

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

7.2.3.7 bombX

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

7.2.3.8 bottomPadX

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


7.2.3.9 displayScore

`HighScore startGame.GameController.displayScore [private]`

7.2.3.10 endTime

`long startGame.GameController.endTime [private]`

7.2.3.11 exit

`JButton startGame.GameController.exit [private]`

7.2.3.12 frameWidth

`int startGame.GameController.frameWidth [private]`

7.2.3.13 gameDisplay

`PongGameDisplay startGame.GameController.gameDisplay [private]`

7.2.3.14 gameFrame

`JFrame startGame.GameController.gameFrame [private]`

Variable declarations for the game

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

7.2.3.15 gameMode

`int startGame.GameController.gameMode [private]`

7.2.3.16 inset

`int startGame.GameController.inset [private]`

7.2.3.17 keys

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

Declare a variable for storing the key pressed records

7.2.3.18 m

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

7.2.3.19 mode

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

7.2.3.20 paddle_player

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

7.2.3.21 padWidth

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

7.2.3.22 pause

```
JButton startGame.GameController.pause [private]
```

7.2.3.23 player

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

7.2.3.24 resume

```
JButton startGame.GameController.resume [private]
```

7.2.3.25 save

```
JButton startGame.GameController.save [private]
```

7.2.3.26 scoreTop

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

7.2.3.27 SINGLE

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

7.2.3.28 startTime

```
long startGame.GameController.startTime [private]
```

7.2.3.29 t

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

7.2.3.30 timeElapsed

```
double startGame.GameController.timeElapsed [private]
```

7.2.3.31 tut

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

7.2.3.32 v

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

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

7.2.3.33 velX

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

7.2.3.34 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

Exceptions

<i>ArithmeticException</i>	ball position could not be set out of the game frame.
----------------------------	---

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
- JButton `getPause` ()
- JButton `getResume` ()
- JButton `getSave` ()
- JButton `getExit` ()
- JPanel `getGameOptionPanel` ()
- void `addButton` (JPanel panel, JButton button, JButton prefer)
- 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, double time)
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
- [JButton](#) `pause`
- [JButton](#) `resume`
- [JButton](#) `save`
- [JButton](#) `exit`
- [JPanel](#) `gameOptions`

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 `addButton()`

```
void view.GameView.addButton (
    JPanel panel,
    JButton button,
    JButton prefer )
```

7.4.2.2 `cannotLoadMessage()`

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

display message for error loading game

create a frame for display

7.4.2.3 createGame()

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

create the game for display

create a frame under set dimension for the game

7.4.2.4 display()

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

displays the welcome page.

sets the visibility of the window to be true.

7.4.2.5 gameOver()

```
void view.GameView.gameOver (
    int whichplayer,
    double time )
```

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

7.4.2.6 getExit()

```
JButton view.GameView.getExit ( )
```

7.4.2.7 getFrameHeight()

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

gets height of the window

Returns

FRAMEHEIGHT

7.4.2.8 getFrameWidth()

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

gets width of the window

Returns

FRAMEWIDTH

7.4.2.9 getGame()

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

gets game window

Returns

game window object

7.4.2.10 getGameFrame()

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

gets game object

Returns

game object

7.4.2.11 getGameOptionPanel()

```
JPanel view.GameView.getGameOptionPanel ( )
```

7.4.2.12 getmode()

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

gets game mode page window

Returns

game mode page object

7.4.2.13 getPause()

```
JButton view.GameView.getPause ( )
```

7.4.2.14 getResume()

```
JButton view.GameView.getResume ( )
```

7.4.2.15 getSave()

```
JButton view.GameView.getSave ( )
```

7.4.2.16 getTutorial()

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

gets tutorial page window

Returns

tutorial page object

7.4.2.17 getWelcome()

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

gets welcome page window

Returns

welcome page object

7.4.2.18 noFileAvailMessage()

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

display message for error loading game record

create a frame for display

7.4.2.19 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 exit

```
JButton view.GameView.exit [private]
```

7.4.3.2 FRAMEHEIGHT

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

7.4.3.3 FRAMEWIDTH

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

Constant declarations for the view

7.4.3.4 gameFrame

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

7.4.3.5 gameOptions

```
JPanel view.GameView.gameOptions [private]
```

7.4.3.6 mode

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

7.4.3.7 pause

```
JButton view.GameView.pause [private]
```

Set up buttons on the game panel

7.4.3.8 ponggame

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

7.4.3.9 resume

`JButton view.GameView.resume [private]`

7.4.3.10 save

`JButton view.GameView.save [private]`

7.4.3.11 tutorial

`Tutorial view.GameView.tutorial [private]`

7.4.3.12 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.HighScore Class Reference

Public Member Functions

- `HighScore ()` throws `IOException`
Constructor for the [HighScore](#).
- void `checkHighScore` (double nameScore, JFrame main) throws `IOException`
checks the high score and update the score file

Static Public Member Functions

- static void `readFrom ()` throws `IOException`
reads score data.
- static void `writeTo ()`
writes the score onto the file
- static boolean `isHigh` (double nameScored)
checks whether the score can be in top 20 list
- static int `findRank ()` throws `IOException`
determines the rank of a score
- static void `highScorePage` (JFrame main) throws `IOException`
displays the high score

7.5.1 Constructor & Destructor Documentation

7.5.1.1 HighScore()

`view.HighScore.HighScore ()` throws `IOException`

Constructor for the [HighScore](#).

calls the read method to read data

Exceptions

<code>IOException</code>	
--------------------------	--

7.5.2 Member Function Documentation

7.5.2.1 checkHighScore()

```
void view.HighScore.checkHighScore (
    double nameScore,
    JFrame main ) throws IOException
```

checks the high score and update the score file

finds the rank for each score and lists them for display

Exceptions

<code>IOException</code>	cannot find and read the highScore.txt
--------------------------	--

Variable declaration

- the score for each user
- variables for the JPanel

Define ranks for each score/user

- Prompt for the user to enter his/her username
- Pop up a confirmation message after saving

Display the score list

7.5.2.2 findRank()

```
static int view.HighScore.findRank ( ) throws IOException [static]
```

determines the rank of a score

Returns

a rank of the score

Exceptions

<i>IOException</i>	cannot find and read the highScore.txt
--------------------	--

Variable declaration

- the current score list
- an indicator for the check action
- a counter for looping through the list

For each score/record, if the current score is greater than the previous one, set this score to have a higher rank and return the rank.

7.5.2.3 highScorePage()

```
static void view.HighScore.highScorePage (
    JFrame main ) throws IOException [static]
```

displays the high score

creates a frame for the display after reading the score file

Exceptions

<i>IOException</i>	cannot find and read the highScore.txt
--------------------	--

Create a JFrame for the display

Read all the score data and put them into an array

Put all the score data into a JTable for display

7.5.2.4 isHigh()

```
static boolean view.HighScore.isHigh (
    double nameScored ) [static]
```

checks whether the score can be in top 20 list

Returns

a boolean that indicate the score can go into the score list

7.5.2.5 readFrom()

```
static void view.HighScore.readFrom ( ) throws IOException [static]
```

reads score data.

reads from file "highScore.txt" and store them into an array.

Exceptions

<i>IOException</i>	cannot find and read the highScore.txt
--------------------	--

Declare a variable to store the file.

- Read the file using `BufferedReader`
- Organize and put user data into the array

7.5.2.6 writeTo()

```
static void view.HighScore.writeTo ( ) [static]
```

writes the score onto the file

opens the file "highScore.txt" and updates scores

Exceptions

<i>IOException</i>	cannot find and write the highScore.txt
--------------------	---

Declare a variable to store the file name

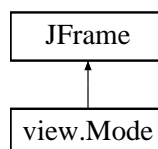
Use `FileWriter` to write scores

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

- `src/view/HighScore.java`

7.6 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
- `JButton getAdvance ()`

Private Attributes

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

7.6.1 Constructor & Destructor Documentation

7.6.1.1 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

7.6.2 Member Function Documentation

7.6.2.1 addButton()

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

adds buttons to a panel

makes buttons align in the panel

7.6.2.2 addListener()

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

adds action listener to the buttons

Parameters

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

7.6.2.3 getAdvance()

```
JButton view.Mode.getAdvance ( )
```

7.6.2.4 `getSingle()`

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

gets the button for single mode

Returns

single

7.6.3 Member Data Documentation

7.6.3.1 `buttonPanel`

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

7.6.3.2 `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

7.6.3.3 `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](#)

7.7 `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](#)

7.7.1 Constructor & Destructor Documentation

7.7.1.1 Paddle()

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

Constructor for a paddle.

Constructor initialize the starting position of a paddle.

7.7.2 Member Function Documentation

7.7.2.1 getHeight()

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

returns the height of the paddle.

Returns

HEIGHT

7.7.2.2 getInset()

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

returns the inset between the paddle and the screen.

Returns

INSET

7.7.2.3 getPositionX()

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

returns the x position of the paddle.

Returns

positionX

7.7.2.4 getPositionY()

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

returns the y position of the paddle.

Returns

positionY

7.7.2.5 getWidth()

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

returns the width of the paddle.

Returns

WIDTH

7.7.2.6 setPositionX()

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

sets the x-position of the paddle.

Parameters

x	is the x position of the paddle.
---	----------------------------------

Exceptions

<i>ArithmeticException</i>	x-position could not be set out of the game frame.
----------------------------	--

7.7.2.7 setPositionY()

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

sets the y-position of the paddle.

Parameters

y	is the y position of the paddle.
---	----------------------------------

Exceptions

<i>ArithmeticException</i>	y-position could not be set out of the game frame.
----------------------------	--

7.7.3 Member Data Documentation

7.7.3.1 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

7.7.3.2 INSET

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

7.7.3.3 positionX

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

The position of a paddle

- horizontal position x
- vertical position y

7.7.3.4 positionY

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

7.7.3.5 speed

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

7.7.3.6 WIDTH

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

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

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

7.8 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.
- void [setScore](#) (int x)
sets the score of the player.
- boolean [checkLoss](#) ()
checks whether the player loses the game or not
- void [resetScore](#) ()

Private Attributes

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

7.8.1 Constructor & Destructor Documentation

7.8.1.1 Player()

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

Constructor for the player.

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

7.8.2 Member Function Documentation

7.8.2.1 checkLoss()

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

checks whether the player loses the game or not

Returns

if a player's score is 0 (NOLIFE), return the true to indicate the player loses.

7.8.2.2 decrementLife()

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

loses score if the ball touches his/her border.

decreases the number of life by 1.

7.8.2.3 getScore()

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

gets the score of a player.

Returns

playerScore returns the score of the player.

7.8.2.4 resetScore()

```
void model.Player.resetScore ( )
```

7.8.2.5 setScore()

```
void model.Player.setScore (
    int x )
```

sets the score of the player.

changes the score to the input value.

Parameters

<i>x</i>	is the input score
----------	--------------------

Exceptions

<i>ArithmeticException</i>	score could not be set less than zero
----------------------------	---------------------------------------

7.8.3 Member Data Documentation

7.8.3.1 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

7.8.3.2 NOLIFE

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

7.8.3.3 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](#)

7.9 startGame.PongGame Class Reference

Static Public Member Functions

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

7.9.1 Member Function Documentation

7.9.1.1 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

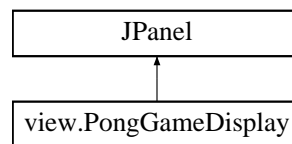
Invoke the game display from the controller

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

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

7.10 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 [setBottomPaddle](#) (int x, int y)
sets the positions of the player paddle
- void [setTopPaddle](#) (int x, int y)
sets the positions of the AI paddle
- void [setBomb](#) (int x, int y)
sets the positions of the bomb
- void [timeForBomb](#) ()
defines the game mode contains a bomb
- void [noBomb](#) ()
defines the game mode does not contain a bomb
- boolean [bombTime](#) ()
checks if it should start displaying the bomb
- void [setBallSize](#) (int s)
sets the size of the ball
- int [getBallSize](#) ()
gets 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*
- int [getTopScore](#) ()
 - gets the score of AI*
- void [setBottomScore](#) (int s)
 - sets the score for player*
- int [getBottomScore](#) ()
 - gets the player score*
- int [getBallX](#) ()
 - gets the x-position of the ball*
- int [getBallY](#) ()
 - gets the y-position of the ball*
- int [getBombX](#) ()
 - gets the x-position of the bomb*
- int [getBombY](#) ()
 - gets the y-position of the bomb*
- 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*
- int [getMode](#) ()
 - gets the mode of the game*

Protected Member Functions

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

Private Attributes

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

7.10.1 Constructor & Destructor Documentation

7.10.1.1 PongGameDisplay()

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

Constructor for [PongGameDisplay](#).

Constructor by default set the game to single mode

7.10.2 Member Function Documentation

7.10.2.1 bombTime()

```
boolean view.PongGameDisplay.bombTime ( )
```

checks if it should start displaying the bomb

Returns

startBomb

7.10.2.2 getBallSize()

```
int view.PongGameDisplay.getBallSize ( )
```

gets the size of the ball

Returns

ballSize

7.10.2.3 getBallX()

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

gets the x-position of the ball

Returns

ballX

7.10.2.4 getBallY()

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

gets the y-position of the ball

Returns

ballY

7.10.2.5 getBombX()

```
int view.PongGameDisplay.getBombX ( )
```

gets the x-position of the bomb

Returns

bombX

7.10.2.6 getBombY()

```
int view.PongGameDisplay.getBombY ( )
```

gets the y-position of the bomb

Returns

bombY

7.10.2.7 getBottomScore()

```
int view.PongGameDisplay.getBottomScore ( )
```

gets the player score

Returns

scoreBottom

7.10.2.8 getBottomX()

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

gets the x-position of the player paddle

Returns

bottomPadX

7.10.2.9 getBottomY()

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

gets the y-position of the player paddle

Returns

bottomPadY

7.10.2.10 `getMode()`

```
int view.PongGameDisplay.getMode ( )
```

gets the mode of the game

Returns

gameMode

7.10.2.11 `getTopScore()`

```
int view.PongGameDisplay.getTopScore ( )
```

gets the score of AI

Returns

scoreTop

7.10.2.12 `noBomb()`

```
void view.PongGameDisplay.noBomb ( )
```

defines the game mode does not contain a bomb

set the flag for the advance to be false

7.10.2.13 `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 the bomb if the mode is the advance mode

Draw scores on the screen by passed in values

7.10.2.14 `setAdvance()`

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

sets the game mode to be advanced

set the flag to advance

7.10.2.15 `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

Exceptions

<i>ArithmeticException</i>	ball position could not be out of the game frame
----------------------------	--

7.10.2.16 setBallSize()

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

sets the size of the ball

Parameters

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

Exceptions

<i>ArithmeticException</i>	ball size should not be less than zero
----------------------------	--

7.10.2.17 setBomb()

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

sets the positions of the bomb

Parameters

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

Exceptions

<i>ArithmeticException</i>	bomb position could not be out of the game frame
----------------------------	--

7.10.2.18 setBottom()

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

sets x-position for the player paddle

Parameters

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

Exceptions

<i>ArithmeticException</i>	paddle x-position could not be out of the game frame
----------------------------	--

7.10.2.19 setBottomPaddle()

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

sets the positions of the player paddle

Parameters

<i>x</i>	is the x-position of the player paddle
<i>y</i>	is the y-position of the player paddle

Exceptions

<i>ArithmeticException</i>	player paddle position could not be out of the game frame
----------------------------	---

7.10.2.20 setBottomScore()

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

sets the score for player

Parameters

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

Exceptions

<i>ArithmeticException</i>	player score could not be less than zero
----------------------------	--

7.10.2.21 setInset()

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

sets the distance between frame and the paddle

Parameters

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

Exceptions

<i>ArithmeticException</i>	distance between paddle and frame could not be set out of the frame
----------------------------	---

7.10.2.22 setPaddleHeight()

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

sets the height of the paddle

Parameters

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

Exceptions

<i>ArithmeticException</i>	paddle height could not be less than zero
----------------------------	---

7.10.2.23 setPaddleWidth()

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

sets the width of the paddle

Parameters

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

Exceptions

<i>ArithmeticException</i>	paddle width could not be less than zero
----------------------------	--

7.10.2.24 setTop()

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

sets x-position for the ai paddle

Parameters

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

Exceptions

<i>ArithmeticException</i>	paddle y-position could not be out of the game frame
----------------------------	--

7.10.2.25 setTopPaddle()

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

sets the positions of the AI paddle

Parameters

<i>x</i>	is the x-position of the AI paddle
<i>y</i>	is the y-position of the AI paddle

Exceptions

<i>ArithmeticException</i>	ai paddle position could not be out of the game frame
----------------------------	---

7.10.2.26 setTopScore()

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

sets the score for ai

Parameters

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

Exceptions

<i>ArithmeticException</i>	ai score could not be less than zero
----------------------------	--------------------------------------

7.10.2.27 timeForBomb()

```
void view.PongGameDisplay.timeForBomb ( )
```

defines the game mode contains a bomb

set the flag for the advance to be true

7.10.3 Member Data Documentation

7.10.3.1 ADVANCE

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

7.10.3.2 ballSize

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

7.10.3.3 ballX

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

7.10.3.4 bombX

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

7.10.3.5 bottomPadX

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

7.10.3.6 first

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

7.10.3.7 frameHeight

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

7.10.3.8 frameWidth

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

Variable declarations for the display

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

7.10.3.9 gameMode

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

7.10.3.10 inset

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

7.10.3.11 padW

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

7.10.3.12 scoreTop

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

7.10.3.13 SINGLE

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

7.10.3.14 startBomb

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

7.10.3.15 topPadX

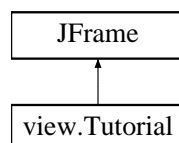
```
int view.PongGameDisplay.topPadX [private]
```

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

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

7.11 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](#)

7.11.1 Constructor & Destructor Documentation

7.11.1.1 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

7.11.2 Member Function Documentation

7.11.2.1 addListener()

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

adds action listener to the button

Parameters

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

7.11.2.2 getBack()

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

gets the button to exit the page

Returns

back is the button for going back to welcome page

7.11.3 Member Data Documentation

7.11.3.1 back

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

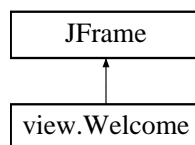
Variable declaration for the back button

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

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

7.12 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`

7.12.1 Constructor & Destructor Documentation

7.12.1.1 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

7.12.2 Member Function Documentation

7.12.2.1 addButton()

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

adds buttons to a panel

makes buttons align in the panel

7.12.2.2 addListener()

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

adds action listener to the buttons

Parameters

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

7.12.2.3 exit()

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

gets the button to exit the program

Returns

exit

7.12.2.4 getStart()

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

gets the start button

Returns

start indicates to start a new game

7.12.2.5 highScores()

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

gets the button to display high score

Returns

highScores

7.12.2.6 load()

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

gets the load button

Returns

load indicates to load a new game

7.12.2.7 tutorial()

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

gets the button to display instructions

Returns

tutorial

7.12.3 Member Data Documentation

7.12.3.1 buttonPanel

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

Define a panel for the arrangement of buttons

7.12.3.2 exit

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

7.12.3.3 highScores

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

7.12.3.4 load

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

7.12.3.5 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

7.12.3.6 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/HighScore.java File Reference

This class displays score data.

Classes

- class [view.HighScore](#)

Packages

- package [view](#)

8.8.1 Detailed Description

This class displays score data.

HighScore

Author

Pongthusiastics

Date

7/12/2016

This class reads, writes, and displays score.

8.9 src/view/Mode.java File Reference

This class create the game mode window.

Classes

- class [view.Mode](#)

Packages

- package [view](#)

8.9.1 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

8.10 src/view/PongGameDisplay.java File Reference

This class construct the view of the pong game.

Classes

- class [view.PongGameDisplay](#)

Packages

- package [view](#)

8.10.1 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

8.11 src/view/Tutorial.java File Reference

This class create the tutorial window.

Classes

- class [view.Tutorial](#)

Packages

- package [view](#)

8.11.1 Detailed Description

This class create the tutorial window.

Tutorial

Author

Pongthusiastics

Date

13/11/2016

This class display instruction for the game

8.12 src/view/Welcome.java File Reference

This class creates the display for welcome page.

Classes

- class [view.Welcome](#)

Packages

- package [view](#)

8.12.1 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, [20](#)
 - view::PongGameDisplay, [54](#)
- addButton
 - view::GameView, [28](#)
 - view::Mode, [37](#)
 - view::Welcome, [58](#)
- addListener
 - view::Mode, [37](#)
 - view::Tutorial, [56](#)
 - view::Welcome, [58](#)
- ai
 - startGame::GameController, [20](#)
- b
 - model::GameModel, [26](#)
 - startGame::GameController, [20](#)
- back
 - view::Tutorial, [57](#)
- Ball
 - model::Ball, [13](#)
- ballSize
 - view::PongGameDisplay, [54](#)
- ballX
 - startGame::GameController, [20](#)
 - view::PongGameDisplay, [54](#)
- bomb
 - startGame::GameController, [20](#)
- bombTime
 - view::PongGameDisplay, [47](#)
- bombVelX
 - startGame::GameController, [20](#)
- bombX
 - startGame::GameController, [20](#)
 - view::PongGameDisplay, [54](#)
- bottomPadX
 - startGame::GameController, [20](#)
 - view::PongGameDisplay, [54](#)
- buttonPanel
 - view::Mode, [38](#)
 - view::Welcome, [60](#)
- cannotLoadMessage
 - view::GameView, [28](#)
- checkGameOver
 - startGame::GameController, [18](#)
- checkHighScore
 - view::HighScore, [34](#)
- checkLoss
 - model::Player, [42](#)

- createGame
 - view::GameView, [28](#)
- decrementLife
 - model::Player, [42](#)
- display
 - startGame::GameController, [18](#)
 - view::GameView, [29](#)
- displayScore
 - startGame::GameController, [20](#)
- endTime
 - startGame::GameController, [21](#)
- exit
 - startGame::GameController, [21](#)
 - view::GameView, [32](#)
 - view::Welcome, [58, 60](#)
- FRAMEHEIGHT
 - view::GameView, [32](#)
- FRAMEWIDTH
 - view::GameView, [32](#)
- findRank
 - view::HighScore, [34](#)
- first
 - view::PongGameDisplay, [54](#)
- frameHeight
 - view::PongGameDisplay, [54](#)
- frameWidth
 - startGame::GameController, [21](#)
 - view::PongGameDisplay, [54](#)
- GameController
 - startGame::GameController, [17](#)
- gameDisplay
 - startGame::GameController, [21](#)
- gameFrame
 - startGame::GameController, [21](#)
 - view::GameView, [32](#)
- gameMode
 - startGame::GameController, [21](#)
 - view::PongGameDisplay, [54](#)
- GameModel
 - model::GameModel, [24](#)
- gameOptions
 - view::GameView, [32](#)
- gameOver
 - view::GameView, [29](#)
- GameView
 - view::GameView, [28](#)

- getAdvance
 - view::Mode, 37
- getBack
 - view::Tutorial, 56
- getBall
 - model::GameModel, 25
- getBallSize
 - view::PongGameDisplay, 47
- getBallX
 - view::PongGameDisplay, 47
- getBallY
 - view::PongGameDisplay, 47
- getBomb
 - model::GameModel, 25
- getBombX
 - view::PongGameDisplay, 47
- getBombY
 - view::PongGameDisplay, 48
- getBottomScore
 - view::PongGameDisplay, 48
- getBottomX
 - view::PongGameDisplay, 48
- getBottomY
 - view::PongGameDisplay, 48
- getComputer
 - model::GameModel, 25
- getComputerPaddle
 - model::GameModel, 25
- getElapsedTime
 - startGame::GameController, 19
- getExit
 - view::GameView, 29
- getFrameHeight
 - view::GameView, 29
- getFrameWidth
 - view::GameView, 29
- getGame
 - view::GameView, 30
- getGameFrame
 - view::GameView, 30
- getGameOptionPanel
 - view::GameView, 30
- getHeight
 - model::Paddle, 39
- getInset
 - model::Paddle, 39
- getMode
 - view::PongGameDisplay, 48
- getPause
 - view::GameView, 30
- getPlayer
 - model::GameModel, 25
- getPlayerPaddle
 - model::GameModel, 25
- getPositionX
 - model::Ball, 14
 - model::Paddle, 39
- getPositionY
 - model::Ball, 14
 - model::Paddle, 39
- getResume
 - view::GameView, 31
- getSave
 - view::GameView, 31
- getScore
 - model::Player, 43
- getSingle
 - view::Mode, 37
- getSize
 - model::Ball, 14
- getStart
 - view::Welcome, 59
- getTopScore
 - view::PongGameDisplay, 49
- getTutorial
 - view::GameView, 31
- getVelX
 - startGame::GameController, 19
- getVelY
 - startGame::GameController, 19
- getWelcome
 - view::GameView, 31
- getWidth
 - model::Paddle, 40
- getbombVelX
 - startGame::GameController, 18
- getbombVelY
 - startGame::GameController, 18
- getmode
 - view::GameView, 30
- HEIGHT
 - model::Paddle, 41
- HighScore
 - view::HighScore, 34
- highScorePage
 - view::HighScore, 35
- highScores
 - view::Welcome, 59, 60
- INSET
 - model::Paddle, 41
- inset
 - startGame::GameController, 21
 - view::PongGameDisplay, 55
- isHigh
 - view::HighScore, 35
- keys
 - startGame::GameController, 21
- LIFE
 - model::Player, 43
- load
 - view::Welcome, 59, 60
- m

- startGame::GameController, 22
- main
 - startGame::PongGame, 44
- Mode
 - view::Mode, 37
- mode
 - startGame::GameController, 22
 - view::GameView, 32
- model, 11
- model.Ball, 13
- model.GameModel, 24
- model.Paddle, 38
- model.Player, 42
- model::Ball
 - Ball, 13
 - getPositionX, 14
 - getPositionY, 14
 - getSize, 14
 - positionX, 15
 - positionY, 15
 - SIZE, 15
 - setPositionX, 14
 - setPositionY, 15
- model::GameModel
 - b, 26
 - GameModel, 24
 - getBall, 25
 - getBomb, 25
 - getComputer, 25
 - getComputerPaddle, 25
 - getPlayer, 25
 - getPlayerPaddle, 25
 - p_player, 26
 - player, 27
 - setBall, 26
 - setBomb, 26
- model::Paddle
 - getHeight, 39
 - getInset, 39
 - getPositionX, 39
 - getPositionY, 39
 - getWidth, 40
 - HEIGHT, 41
 - INSET, 41
 - Paddle, 39
 - positionX, 41
 - positionY, 41
 - setPositionX, 40
 - setPositionY, 40
 - speed, 41
 - WIDTH, 41
- model::Player
 - checkLoss, 42
 - decrementLife, 42
 - getScore, 43
 - LIFE, 43
 - NOLIFE, 44
 - Player, 42
 - resetScore, 43
 - score, 44
 - setScore, 43
- NOLIFE
 - model::Player, 44
- noBomb
 - view::PongGameDisplay, 49
- noFileAvailMessage
 - view::GameView, 31
- p_player
 - model::GameModel, 26
- padWidth
 - startGame::GameController, 22
- Paddle
 - model::Paddle, 39
- paddle_player
 - startGame::GameController, 22
- padW
 - view::PongGameDisplay, 55
- paintComponent
 - view::PongGameDisplay, 49
- pause
 - startGame::GameController, 22
 - view::GameView, 32
- Player
 - model::Player, 42
- player
 - model::GameModel, 27
 - startGame::GameController, 22
- PongGameDisplay
 - view::PongGameDisplay, 47
- ponggame
 - view::GameView, 32
- positionX
 - model::Ball, 15
 - model::Paddle, 41
- positionY
 - model::Ball, 15
 - model::Paddle, 41
- readFrom
 - view::HighScore, 35
- resetGame
 - startGame::GameController, 19
- resetScore
 - model::Player, 43
- resume
 - startGame::GameController, 22
 - view::GameView, 32
- SINGLE
 - startGame::GameController, 22
 - view::PongGameDisplay, 55
- SIZE
 - model::Ball, 15
- sObstacle
 - view::Mode, 38

- save
 - startGame::GameController, 22
 - view::GameView, 33
- score
 - model::Player, 44
- scoreTop
 - startGame::GameController, 22
 - view::PongGameDisplay, 55
- setAdvance
 - view::PongGameDisplay, 49
- setBall
 - model::GameModel, 26
 - view::PongGameDisplay, 49
- setBallSize
 - view::PongGameDisplay, 50
- setBomb
 - model::GameModel, 26
 - view::PongGameDisplay, 50
- setBottom
 - view::PongGameDisplay, 50
- setBottomPaddle
 - view::PongGameDisplay, 51
- setBottomScore
 - view::PongGameDisplay, 51
- setInset
 - view::PongGameDisplay, 51
- setPaddleHeight
 - view::PongGameDisplay, 52
- setPaddleWidth
 - view::PongGameDisplay, 52
- setPositionX
 - model::Ball, 14
 - model::Paddle, 40
- setPositionY
 - model::Ball, 15
 - model::Paddle, 40
- setScore
 - model::Player, 43
- setSpeed
 - startGame::GameController, 19
- setTop
 - view::PongGameDisplay, 52
- setTopPaddle
 - view::PongGameDisplay, 53
- setTopScore
 - view::PongGameDisplay, 53
- single
 - view::Mode, 38
- speed
 - model::Paddle, 41
- src/model/Ball.java, 61
- src/model/GameModel.java, 61
- src/model/Paddle.java, 62
- src/model/Player.java, 63
- src/startGame/GameController.java, 63
- src/startGame/PongGame.java, 64
- src/view/GameView.java, 65
- src/view/HighScore.java, 65
- src/view/Mode.java, 66
- src/view/PongGameDisplay.java, 67
- src/view/Tutorial.java, 67
- src/view/Welcome.java, 68
- start
 - view::Welcome, 60
- startBomb
 - view::PongGameDisplay, 55
- startGame, 11
- startGame.GameController, 15
- startGame.PongGame, 44
- startGame::GameController
 - ADVANCE, 20
 - ai, 20
 - b, 20
 - ballX, 20
 - bomb, 20
 - bombVelX, 20
 - bombX, 20
 - bottomPadX, 20
 - checkGameOver, 18
 - display, 18
 - displayScore, 20
 - endTime, 21
 - exit, 21
 - frameWidth, 21
 - GameController, 17
 - gameDisplay, 21
 - gameFrame, 21
 - gameMode, 21
 - getElapsedTime, 19
 - getVelX, 19
 - getVelY, 19
 - getbombVelX, 18
 - getbombVelY, 18
 - inset, 21
 - keys, 21
 - m, 22
 - mode, 22
 - padWidth, 22
 - paddle_player, 22
 - pause, 22
 - player, 22
 - resetGame, 19
 - resume, 22
 - SINGLE, 22
 - save, 22
 - scoreTop, 22
 - setSpeed, 19
 - startTime, 23
 - t, 23
 - timeElapsed, 23
 - tut, 23
 - v, 23
 - velX, 23
 - w, 23
- startGame::PongGame
 - main, 44

- startTime
 - startGame::GameController, [23](#)
- t
 - startGame::GameController, [23](#)
- timeElapsed
 - startGame::GameController, [23](#)
- timeForBomb
 - view::PongGameDisplay, [53](#)
- topPadX
 - view::PongGameDisplay, [55](#)
- tut
 - startGame::GameController, [23](#)
- Tutorial
 - view::Tutorial, [56](#)
- tutorial
 - view::GameView, [33](#)
 - view::Welcome, [59](#), [60](#)
- tutorialPage
 - view::GameView, [31](#)
- v
 - startGame::GameController, [23](#)
- velX
 - startGame::GameController, [23](#)
- view, [11](#)
- view.GameView, [27](#)
- view.HighScore, [33](#)
- view.Mode, [36](#)
- view.PongGameDisplay, [45](#)
- view.Tutorial, [55](#)
- view.Welcome, [57](#)
- view::GameView
 - addButton, [28](#)
 - cannotLoadMessage, [28](#)
 - createGame, [28](#)
 - display, [29](#)
 - exit, [32](#)
 - FRAMEHEIGHT, [32](#)
 - FRAMEWIDTH, [32](#)
 - gameFrame, [32](#)
 - gameOptions, [32](#)
 - gameOver, [29](#)
 - GameView, [28](#)
 - getExit, [29](#)
 - getFrameHeight, [29](#)
 - getFrameWidth, [29](#)
 - getGame, [30](#)
 - getGameFrame, [30](#)
 - getGameOptionPanel, [30](#)
 - getPause, [30](#)
 - getResume, [31](#)
 - getSave, [31](#)
 - getTutorial, [31](#)
 - getWelcome, [31](#)
 - getmode, [30](#)
 - mode, [32](#)
 - noFileAvailMessage, [31](#)
 - pause, [32](#)
 - ponggame, [32](#)
 - resume, [32](#)
 - save, [33](#)
 - tutorial, [33](#)
 - tutorialPage, [31](#)
 - welcome, [33](#)
- view::HighScore
 - checkHighScore, [34](#)
 - findRank, [34](#)
 - HighScore, [34](#)
 - highScorePage, [35](#)
 - isHigh, [35](#)
 - readFrom, [35](#)
 - writeTo, [36](#)
- view::Mode
 - addButton, [37](#)
 - addListener, [37](#)
 - buttonPanel, [38](#)
 - getAdvance, [37](#)
 - getSingle, [37](#)
 - Mode, [37](#)
 - sObstacle, [38](#)
 - single, [38](#)
- view::PongGameDisplay
 - ADVANCE, [54](#)
 - ballSize, [54](#)
 - ballX, [54](#)
 - bombTime, [47](#)
 - bombX, [54](#)
 - bottomPadX, [54](#)
 - first, [54](#)
 - frameHeight, [54](#)
 - frameWidth, [54](#)
 - gameMode, [54](#)
 - getBallSize, [47](#)
 - getBallX, [47](#)
 - getBallY, [47](#)
 - getBombX, [47](#)
 - getBombY, [48](#)
 - getBottomScore, [48](#)
 - getBottomX, [48](#)
 - getBottomY, [48](#)
 - getMode, [48](#)
 - getTopScore, [49](#)
 - inset, [55](#)
 - noBomb, [49](#)
 - padW, [55](#)
 - paintComponent, [49](#)
 - PongGameDisplay, [47](#)
 - SINGLE, [55](#)
 - scoreTop, [55](#)
 - setAdvance, [49](#)
 - setBall, [49](#)
 - setBallSize, [50](#)
 - setBomb, [50](#)
 - setBottom, [50](#)
 - setBottomPaddle, [51](#)
 - setBottomScore, [51](#)

- setInset, [51](#)
- setPaddleHeight, [52](#)
- setPaddleWidth, [52](#)
- setTop, [52](#)
- setTopPaddle, [53](#)
- setTopScore, [53](#)
- startBomb, [55](#)
- timeForBomb, [53](#)
- topPadX, [55](#)
- view::Tutorial
 - addListener, [56](#)
 - back, [57](#)
 - getBack, [56](#)
 - Tutorial, [56](#)
- view::Welcome
 - addButton, [58](#)
 - addListener, [58](#)
 - buttonPanel, [60](#)
 - exit, [58](#), [60](#)
 - getStart, [59](#)
 - highScores, [59](#), [60](#)
 - load, [59](#), [60](#)
 - start, [60](#)
 - tutorial, [59](#), [60](#)
 - Welcome, [58](#)
- w
 - startGame::GameController, [23](#)
- WIDTH
 - model::Paddle, [41](#)
- Welcome
 - view::Welcome, [58](#)
- welcome
 - view::GameView, [33](#)
- writeTo
 - view::HighScore, [36](#)