

Regex Games

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	checkForClick Namespace Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Variable Documentation . . . . .	7
4.1.2.1	dispEnd . . . . .	7
4.1.2.2	ms . . . . .	7
4.1.2.3	resp . . . . .	8
4.1.2.4	rt . . . . .	8
4.2	FinalImplementation Namespace Reference . . . . .	8
4.2.1	Detailed Description . . . . .	8
4.2.2	Function Documentation . . . . .	8
4.2.2.1	fillAnswerListBasic() . . . . .	8
4.2.2.2	fillAnswerListDifficulty(difficultyString) . . . . .	9
4.2.2.3	fillQuestionListBasic() . . . . .	9
4.2.2.4	fillQuestionListDifficulty(difficultyString) . . . . .	9
4.2.2.5	gameDeclaration() . . . . .	9

4.2.3	Variable Documentation . . . . .	9
4.2.3.1	answerList . . . . .	9
4.2.3.2	cur . . . . .	9
4.2.3.3	db . . . . .	9
4.2.3.4	questionList . . . . .	9
4.3	mouseClick Namespace Reference . . . . .	10
4.3.1	Detailed Description . . . . .	10
4.3.2	Function Documentation . . . . .	10
4.3.2.1	do_on_click(clicked_position) . . . . .	10
4.3.2.2	draw(canvas) . . . . .	10
4.3.3	Variable Documentation . . . . .	10
4.3.3.1	click . . . . .	10
4.3.3.2	click_pos . . . . .	10
4.3.3.3	frame . . . . .	11
4.4	playerClass Namespace Reference . . . . .	11
4.4.1	Detailed Description . . . . .	11
<b>5</b>	<b>Class Documentation</b>	<b>13</b>
5.1	playerClass.player Class Reference . . . . .	13
5.1.1	Constructor & Destructor Documentation . . . . .	13
5.1.1.1	__init__(self) . . . . .	13
5.1.2	Member Function Documentation . . . . .	13
5.1.2.1	addScore(self) . . . . .	13
5.1.2.2	printScore(self) . . . . .	14
5.1.3	Member Data Documentation . . . . .	14
5.1.3.1	score . . . . .	14
5.1.3.2	score . . . . .	14
<b>6</b>	<b>File Documentation</b>	<b>15</b>
6.1	checkForClick.py File Reference . . . . .	15
6.2	FinalImplementation.py File Reference . . . . .	15
6.3	mouseClick.py File Reference . . . . .	16
6.4	playerClass.py File Reference . . . . .	16
<b>Index</b>		<b>17</b>

# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">checkForClick</a>	We check for a click on the gameboard Each time we get a click we update the screen We make sure that the mouse is visible as well . . . . .	7
<a href="#">FinalImplementation</a>	This is the final implementation for the terminal version of our game . . . . .	8
<a href="#">mouseClick</a>	Here is our functionality for mouse clicking within the graphical version . . . . .	10
<a href="#">playerClass</a>	This is the definition for our player class . . . . .	11



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">playerClass.player</a> . . . . .	13
--	----





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">checkForClick.py</a>	15
<a href="#">FinalImplementation.py</a>	15
<a href="#">mouseClick.py</a>	16
<a href="#">playerClass.py</a>	16



## Chapter 4

# Namespace Documentation

### 4.1 checkForClick Namespace Reference

We check for a click on the gameboard Each time we get a click we update the screen We make sure that the mouse is visible as well.

#### Variables

- `resp` = None
- `dispEnd` = `time.time()`  
*we display the time*
- `ms` = `pygame.mouse.get_pressed()`  
*This is the variable that holds the mouse click itself.*
- `rt` = `dispEnd-dispSt`

#### 4.1.1 Detailed Description

We check for a click on the gameboard Each time we get a click we update the screen We make sure that the mouse is visible as well.

#### 4.1.2 Variable Documentation

##### 4.1.2.1 `checkForClick.dispEnd = time.time()`

we display the time

##### 4.1.2.2 `checkForClick.ms = pygame.mouse.get_pressed()`

This is the variable that holds the mouse click itself.

4.1.2.3 `string checkForClick.resp = None`

4.1.2.4 `checkForClick.rt = dispEnd-dispSt`

## 4.2 FinalImplementation Namespace Reference

This is the final implementation for the terminal version of our game.

### Functions

- `def fillQuestionListBasic ()`  
*These are the functions for our "easy" mode, using basic questions.*
- `def fillAnswerListBasic ()`  
*These are the functions for our "easy" mode, using basic questions.*
- `def fillQuestionListDifficulty (difficultyString)`  
*Here is the function for our difficulty selector.*
- `def fillAnswerListDifficulty (difficultyString)`  
*Here is the function for our difficulty selector.*
- `def gameDeclaration ()`  
*We declare an instance of the "player" class that we have declared.*

### Variables

- `db = MySQLdb.connect("127.0.0.1", "root", "root", "regexQuestions")`  
*This is the DB that we connect to, it holds all of our questions.*
- `cur = db.cursor()`
- `list questionList = []`  
*This is the list that we append all of our questions to.*
- `list answerList = []`  
*This is the list we append all of our answers to.*

### 4.2.1 Detailed Description

This is the final implementation for the terminal version of our game.

We have implemented our 'hardness-selection' functionality, we pull in from our database of questions and we are able to create a player and track their score.

### 4.2.2 Function Documentation

4.2.2.1 `def FinalImplementation.fillAnswerListBasic ( )`

These are the functions for our "easy" mode, using basic questions.

We pull a single question/answer from the database at a time using a continuous for loop. We then append the lists with questions/answers

#### 4.2.2.2 `def FinalImplementation.fillAnswerListDifficulty ( difficultyString )`

Here is the function for our difficulty selector.

It follows the same principle as above, except it has a input string check for Hard/Medium settings. This fills the Answer List that we have

#### 4.2.2.3 `def FinalImplementation.fillQuestionListBasic ( )`

These are the functions for our "easy" mode, using basic questions.

We pull a single question/answer from the database at a time using a continuous for loop. We then append the lists with questions/answers""

#### 4.2.2.4 `def FinalImplementation.fillQuestionListDifficulty ( difficultyString )`

Here is the function for our difficulty selector.

It follows the same principle as above, except it has a input string check for Hard/Medium settings

#### 4.2.2.5 `def FinalImplementation.gameDeclaration ( )`

We declare an instance of the "player" class that we have declared.

### 4.2.3 Variable Documentation

#### 4.2.3.1 `FinalImplementation.answerList = [ ]`

This is the list we append all of our answers to.

It's also a simple python list.

#### 4.2.3.2 `FinalImplementation.cur = db.cursor()`

#### 4.2.3.3 `FinalImplementation.db = MySQLdb.connect("127.0.0.1", "root", "root", "regexQuestions")`

This is the DB that we connect to, it holds all of our questions.

#### 4.2.3.4 `FinalImplementation.questionList = [ ]`

This is the list that we append all of our questions to.

It's a simple python list

## 4.3 mouseClicked Namespace Reference

Here is our functionality for mouse clicking within the graphical version.

### Functions

- def `draw` (canvas)  
*We create the screen necessary for the click here.*
- def `do_on_click` (clicked\_position)

### Variables

- list `click_pos` = []  
*This is to keep track of the position of the mouse click.*
- list `click` = []  
*We store our 'click' in the list here.*
- `frame` = `simplegui.create_frame("Mouse", 400,400)`

### 4.3.1 Detailed Description

Here is our functionality for mouse clicking within the graphical version.

### 4.3.2 Function Documentation

4.3.2.1 `def mouseClicked.do_on_click ( clicked_position )`

4.3.2.2 `def mouseClicked.draw ( canvas )`

We create the screen necessary for the click here.

### 4.3.3 Variable Documentation

4.3.3.1 `mouseClick.click = []`

We store our 'click' in the list here.

4.3.3.2 `mouseClick.click_pos = []`

This is to keep track of the position of the mouse click.

4.3.3.3 `mouseClick.frame = simplegui.create_frame("Mouse", 400,400)`

## 4.4 playerClass Namespace Reference

This is the definition for our player class.

### Classes

- class `player`

#### 4.4.1 Detailed Description

This is the definition for our player class.

Each player object is initialized with a score of 0 and can add to that score and print from that score.





## Chapter 5

# Class Documentation

### 5.1 playerClass.player Class Reference

#### Public Member Functions

- def `__init__` (self)  
*This is the constructor for the player class.*
- def `addScore` (self)  
*We define the addScore function here, to add to the score of the player.*
- def `printScore` (self)  
*We define the printScore function here, so as to print the current score of the player.*

#### Public Attributes

- `score`

#### Static Public Attributes

- int `score` = 0

#### 5.1.1 Constructor & Destructor Documentation

##### 5.1.1.1 def playerClass.player.\_\_init\_\_ ( self )

This is the constructor for the player class.

We initialize the score to 0.

#### 5.1.2 Member Function Documentation

##### 5.1.2.1 def playerClass.player.addScore ( self )

We define the addScore function here, to add to the score of the player.

5.1.2.2 `def playerClass.player.printScore ( self )`

We define the `printScore` function here, so as to print the current score of the player.

### 5.1.3 Member Data Documentation

5.1.3.1 `int playerClass.player.score = 0` `[static]`

5.1.3.2 `playerClass.player.score`

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

- [playerClass.py](#)

## Chapter 6

# File Documentation

### 6.1 checkForClick.py File Reference

#### Namespaces

- [checkForClick](#)

*We check for a click on the gameboard Each time we get a click we update the screen We make sure that the mouse is visible as well.*

#### Variables

- [checkForClick.resp](#) = None
- [checkForClick.dispEnd](#) = time.time()  
*we display the time*
- [checkForClick.ms](#) = pygame.mouse.get\_pressed()  
*This is the variable that holds the mouse click itself.*
- [checkForClick.rt](#) = dispEnd-dispSt

### 6.2 FinalImplementation.py File Reference

#### Namespaces

- [FinalImplementation](#)

*This is the final implementation for the terminal version of our game.*

#### Functions

- def [FinalImplementation.fillQuestionListBasic](#) ()  
*These are the functions for our "easy" mode, using basic questions.*
- def [FinalImplementation.fillAnswerListBasic](#) ()  
*These are the functions for our "easy" mode, using basic questions.*
- def [FinalImplementation.fillQuestionListDifficulty](#) (difficultyString)  
*Here is the function for our difficulty selector.*
- def [FinalImplementation.fillAnswerListDifficulty](#) (difficultyString)  
*Here is the function for our difficulty selector.*
- def [FinalImplementation.gameDeclaration](#) ()  
*We declare an instance of the "player" class that we have declared.*

## Variables

- `FinalImplementation.db = MySQLdb.connect("127.0.0.1", "root", "root", "regexQuestions")`  
*This is the DB that we connect to, it holds all of our questions.*
- `FinalImplementation.cur = db.cursor()`
- list `FinalImplementation.questionList = []`  
*This is the list that we append all of our questions to.*
- list `FinalImplementation.answerList = []`  
*This is the list we append all of our answers to.*

## 6.3 mouseClicked.py File Reference

### Namespaces

- `mouseClick`  
*Here is our functionality for mouse clicking within the graphical version.*

### Functions

- def `mouseClick.draw` (canvas)  
*We create the screen necessary for the click here.*
- def `mouseClick.do_on_click` (clicked\_position)

### Variables

- list `mouseClick.click_pos = []`  
*This is to keep track of the position of the mouse click.*
- list `mouseClick.click = []`  
*We store our 'click' in the list here.*
- `mouseClick.frame = simplegui.create_frame("Mouse", 400,400)`

## 6.4 playerClass.py File Reference

### Classes

- class `playerClass.player`

### Namespaces

- `playerClass`  
*This is the definition for our player class.*

# Index

- [\\_\\_init\\_\\_](#)
    - [playerClass::player](#), [13](#)
- [addScore](#)
  - [playerClass::player](#), [13](#)
- [answerList](#)
  - [FinalImplementation](#), [9](#)
- [checkForClick](#), [7](#)
  - [dispEnd](#), [7](#)
  - [ms](#), [7](#)
  - [resp](#), [7](#)
  - [rt](#), [8](#)
- [checkForClick.py](#), [15](#)
- [click](#)
  - [mouseClick](#), [10](#)
- [click\\_pos](#)
  - [mouseClick](#), [10](#)
- [cur](#)
  - [FinalImplementation](#), [9](#)
- [db](#)
  - [FinalImplementation](#), [9](#)
- [dispEnd](#)
  - [checkForClick](#), [7](#)
- [do\\_on\\_click](#)
  - [mouseClick](#), [10](#)
- [draw](#)
  - [mouseClick](#), [10](#)
- [fillAnswerListBasic](#)
  - [FinalImplementation](#), [8](#)
- [fillAnswerListDifficulty](#)
  - [FinalImplementation](#), [8](#)
- [fillQuestionListBasic](#)
  - [FinalImplementation](#), [9](#)
- [fillQuestionListDifficulty](#)
  - [FinalImplementation](#), [9](#)
- [FinalImplementation](#), [8](#)
  - [answerList](#), [9](#)
  - [cur](#), [9](#)
  - [db](#), [9](#)
  - [fillAnswerListBasic](#), [8](#)
  - [fillAnswerListDifficulty](#), [8](#)
  - [fillQuestionListBasic](#), [9](#)
  - [fillQuestionListDifficulty](#), [9](#)
  - [gameDeclaration](#), [9](#)
  - [questionList](#), [9](#)
- [FinalImplementation.py](#), [15](#)
- [frame](#)

- [mouseClick](#), [10](#)
- [gameDeclaration](#)
  - [FinalImplementation](#), [9](#)
- [mouseClick](#), [10](#)
  - [click](#), [10](#)
  - [click\\_pos](#), [10](#)
  - [do\\_on\\_click](#), [10](#)
  - [draw](#), [10](#)
  - [frame](#), [10](#)
- [mouseClick.py](#), [16](#)
- [ms](#)
  - [checkForClick](#), [7](#)
- [playerClass](#), [11](#)
- [playerClass.player](#), [13](#)
- [playerClass.py](#), [16](#)
- [playerClass::player](#)
  - [\\_\\_init\\_\\_](#), [13](#)
  - [addScore](#), [13](#)
  - [printScore](#), [13](#)
  - [score](#), [14](#)
- [printScore](#)
  - [playerClass::player](#), [13](#)
- [questionList](#)
  - [FinalImplementation](#), [9](#)
- [resp](#)
  - [checkForClick](#), [7](#)
- [rt](#)
  - [checkForClick](#), [8](#)
- [score](#)
  - [playerClass::player](#), [14](#)