

RussianCheckers

Generated by Doxygen 1.8.6

Fri Jun 5 2015 23:17:00

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	CheckersServer.Checker Namespace Reference	7
4.1.1	Detailed Description	7
5	Class Documentation	9
5.1	CheckersServer.Checker.Checker Class Reference	9
5.1.1	Detailed Description	9
5.2	CheckersServer.Game.Game Class Reference	9
5.2.1	Detailed Description	10
5.3	GameState Class Reference	10
5.3.1	Constructor & Destructor Documentation	10
5.3.1.1	GameState	10
5.3.2	Member Function Documentation	10
5.3.2.1	getPlayerColor	10
5.3.2.2	insertPlayerData	10
5.3.2.3	makeMove	10
5.3.2.4	playerLoss	11
5.3.2.5	playerMove	11
5.3.2.6	playerWin	11
5.4	SimpleChecker Struct Reference	11
5.5	CheckersServer.TopScorers.TopScorers Class Reference	11
5.5.1	Detailed Description	12
5.5.2	Member Function Documentation	12
5.5.2.1	addNewTopScorer	12

5.5.2.2	isGoodEnough	12
5.5.2.3	sendingList	12
5.5.2.4	update	12
5.6	CheckersServer.Server.WSHandler Class Reference	12
5.6.1	Detailed Description	13
5.6.2	Member Function Documentation	13
5.6.2.1	check_origin	13
5.6.2.2	hello	13
5.6.2.3	interpret	13
5.6.2.4	on_message	13
5.6.3	Member Data Documentation	13
5.6.3.1	json_data	13
5.6.3.2	message	13
5.6.3.3	s	13
Index		15

Chapter 1

Namespace Index

1.1 Namespace List

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

CheckersServer.Checker	7
--	---

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

GameState	10
object	
CheckersServer.Checker.Checker	9
CheckersServer.Game.Game	9
CheckersServer.TopScorers.TopScorers	11
SimpleChecker	11
WebSocketHandler	
CheckersServer.Server.WSHandler	12

Chapter 3

Class Index

3.1 Class List

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

CheckersServer.Checker.Checker	9
CheckersServer.Game.Game	9
GameState	10
SimpleChecker	11
CheckersServer.TopScorers.TopScorers	11
CheckersServer.Server.WSHandler	12

Chapter 4

Namespace Documentation

4.1 CheckersServer.Checker Namespace Reference

Classes

- class [Checker](#)

4.1.1 Detailed Description

Klasa przechowująca dane dotyczące pionka

Chapter 5

Class Documentation

5.1 CheckersServer.Checker.Checker Class Reference

Inherits object.

Public Member Functions

- def **__init__**
- def **color**
- def **isKing**
- def **setKing**
- def **setCaptured**
- def **isCapture**
- def **setPosition**
- def **getPosition**

Public Attributes

- **color**
- **king**
- **captured**
- **pos**
- **damka**
- **zbity**
- **position**

5.1.1 Detailed Description

Konstruktor, inicjuje wartosci przechowywane w klasie wartosciami podanymi w parametrach

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

- CheckersServer/Checker.py

5.2 CheckersServer.Game.Game Class Reference

Inherits object.

Public Member Functions

- def `__init__`
- def `startGame`
- def `makeMove`
- def `inform`

5.2.1 Detailed Description

inicjuje kolekcje przechowujaca dane dotyczace pionkow

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

- CheckersServer/Game.py

5.3 GameState Class Reference

Public Member Functions

- void `playerMove` (int, int, int, int)
- void `insertPlayerData` (char const *)
- `GameState` (char cc, char pc)
- char const * `getPlayerColor` ()
- bool `playerWin` ()
- bool `playerLoss` ()
- char const * `makeMove` ()

5.3.1 Constructor & Destructor Documentation

5.3.1.1 `GameState::GameState (char cc, char pc)`

Konstruktor klasy `GameState`.

Ustala kolor pionków gracza i komputera, rozmieszcza pionki na planszy. cc - kolor pionków komputera pc - kolor pionków gracza

5.3.2 Member Function Documentation

5.3.2.1 `char const * GameState::getPlayerColor ()`

Zwraca kolor pionków gracza.

5.3.2.2 `void GameState::insertPlayerData (char const * s)`

Parsuje dane o ruchu przychodzace od klienta oraz wykonujace wskazane ruchy

5.3.2.3 `char const * GameState::makeMove ()`

Realizuje ruch ze strony serwera oraz czy żaden z graczy nie będzie zablokowany.

Szuka najlepszej możliwości ruchu dla gracza komputerowego, jeśli jej nie znajdzie sygnalizuje blokadę. Wykonuje znaleziony ruch i sprawdza klient nie jest zablokowany. Zwraca wykonany ruch w postaci ciągu znaków.

5.3.2.4 bool GameState::playerLoss ()

Informuje czy gracz przegrał czy nie.

5.3.2.5 void GameState::playerMove (int x1, int y1, int x2, int y2)

Alias executeMove

5.3.2.6 bool GameState::playerWin ()

Informuje czy gracz wygrał czy nie.

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

- CheckersServer/GameState.h
- CheckersServer/GameState.cpp

5.4 SimpleChecker Struct Reference

Public Member Functions

- bool **getKing** ()
- char **getColor** ()
- **SimpleChecker** (char c, bool k)

Public Attributes

- char **color**
- bool **king**

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

- CheckersServer/GameState.h

5.5 CheckersServer.TopScorers.TopScorers Class Reference

Inherits object.

Public Member Functions

- def **__init__**
- def **read**
- def **update**
- def **sendingList**
- def **isGoodEnough**
- def **addNewTopScorer**

Static Public Attributes

- tuple **target** = open('TopScorers')
- string **message** = '{"message": "rank", "times": ['
- int **j** = 4

5.5.1 Detailed Description

Klasa przechowująca obecną listę graczy o najlepszych czasach wygranej, lista jest wczytywana z pliku

5.5.2 Member Function Documentation

5.5.2.1 `def CheckersServer.TopScorers.TopScorers.addNewTopScorer (self, time)`

Dodaje wynik oraz dane gracza na odpowiednie miejsce listy

5.5.2.2 `def CheckersServer.TopScorers.TopScorers.isGoodEnough (self, time)`

Sprawdza czy podany czas jest lepszy niż najgorszy wynik na liście

5.5.2.3 `def CheckersServer.TopScorers.TopScorers.sendingList (self)`

Tworzy pakiet gotowy do wysłania na serwer

5.5.2.4 `def CheckersServer.TopScorers.TopScorers.update (self)`

Zapisuje obecny stan listy, nadpisując plik

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

- CheckersServer/TopScorers.py

5.6 CheckersServer.Server.WSHandler Class Reference

Inherits WebSocketHandler.

Public Member Functions

- `def open`
- `def on_message`
- `def on_close`
- `def check_origin`
- `def interpret`
- `def hello`
- `def ready`
- `def move`

Static Public Attributes

- `json_data = message;`
- `tuple useful_data = json.loads(json_data)`
- `string message = '{"message": "hello", "pieces": "white"}'`
- `tuple s = str()`
- `tuple res = str()`
- `dictionary message = {"message": "end", "time": self.gameTime, "clientWin": True}`

5.6.1 Detailed Description

Klasa odpowiadajaca za komunikacje serwera z przegladarka.

Tworzy serwer websocket ktorzy odbiera komunikaty od klienta oraz implementuje gracza komputerowego.

5.6.2 Member Function Documentation

5.6.2.1 `def CheckersServer.Server.WSHandler.check_origin (self, origin)`

wykonuje sie po zamknieciu polaczenia z serwerem

5.6.2.2 `def CheckersServer.Server.WSHandler.hello (self, pieces)`

Funkcja interpretujaca wiadomosc od klienta.

Sprawdza tresc nadeslanej przez klienta wiadomosci

5.6.2.3 `def CheckersServer.Server.WSHandler.interpret (self, dict)`

Pozwala na przyjmowanie zapytan z roznych zrodel

5.6.2.4 `def CheckersServer.Server.WSHandler.on_message (self, message)`

Wykonuje sie po otwarciu polaczenia z serwerem

Wczytywane sa wyniki z pliku tekstowego.

5.6.3 Member Data Documentation

5.6.3.1 `CheckersServer.Server.WSHandler.json_data = message; [static]`

Odbiera wiadomosc tekstowa wyslana do serwera

Funkcja odbiera wiadomosc z serwera, nastepnie konwertuje ja z JSON do dict i interpretuje.

5.6.3.2 `dictionary CheckersServer.Server.WSHandler.message = '{"message": "hello", "pieces": "white"}' [static]`

Funkcja rozpoczynajaca gre.

Przyjmuje powitanie oraz preferowany przez gracza kolor pionkow, odpowiada wysylajac wiadomosc o wybranym kolorze

5.6.3.3 `tuple CheckersServer.Server.WSHandler.s = str() [static]`

Odbiera od klienta wiadomosc o gotowosci.

Odbiera od klienta wiadomosc o gotowosci, rozpoczyna liczenie czasu gry i zaleznie od wybranego koloru czeka na ruch

Wykonuje ruchy, stwierdza koniec gry.

Na podstawie otrzymanych od klienta danych wykonuje ruch, nastepnie sprawdza czy gra nie zostala zakonczona, Nastepnie wykonuje ruch komputera i ponownie sprawdza czy gra nie zostala zakonczona.

Komunikaty o wykonanych ruchach sa przekazywane w postaci JSON do klienta.

Jesli gra zostala zakonczona, sprawdza czas gry i wysyla wiadomosc klientowi.

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

- CheckersServer/Server.py

Index

- addNewTopScorer
 - CheckersServer::TopScorers::TopScorers, [12](#)
- check_origin
 - CheckersServer::Server::WSHandler, [13](#)
- CheckersServer.Checker, [7](#)
- CheckersServer.Checker.Checker, [9](#)
- CheckersServer.Game.Game, [9](#)
- CheckersServer.Server.WSHandler, [12](#)
- CheckersServer.TopScorers.TopScorers, [11](#)
- CheckersServer::Server::WSHandler
 - check_origin, [13](#)
 - hello, [13](#)
 - interpret, [13](#)
 - json_data, [13](#)
 - message, [13](#)
 - on_message, [13](#)
 - s, [13](#)
- CheckersServer::TopScorers::TopScorers
 - addNewTopScorer, [12](#)
 - isGoodEnough, [12](#)
 - sendingList, [12](#)
 - update, [12](#)
- GameState, [10](#)
 - GameState, [10](#)
 - GameState, [10](#)
 - getPlayerColor, [10](#)
 - insertPlayerData, [10](#)
 - makeMove, [10](#)
 - playerLoss, [10](#)
 - playerMove, [11](#)
 - playerWin, [11](#)
- getPlayerColor
 - GameState, [10](#)
- hello
 - CheckersServer::Server::WSHandler, [13](#)
- insertPlayerData
 - GameState, [10](#)
- interpret
 - CheckersServer::Server::WSHandler, [13](#)
- isGoodEnough
 - CheckersServer::TopScorers::TopScorers, [12](#)
- json_data
 - CheckersServer::Server::WSHandler, [13](#)
- makeMove
 - GameState, [10](#)
- message
 - CheckersServer::Server::WSHandler, [13](#)
- on_message
 - CheckersServer::Server::WSHandler, [13](#)
- playerLoss
 - GameState, [10](#)
- playerMove
 - GameState, [11](#)
- playerWin
 - GameState, [11](#)
- s
 - CheckersServer::Server::WSHandler, [13](#)
- sendingList
 - CheckersServer::TopScorers::TopScorers, [12](#)
- SimpleChecker, [11](#)
- update
 - CheckersServer::TopScorers::TopScorers, [12](#)