

Gra w kości

2.0

Generated by Doxygen 1.9.5



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy . . . . .	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List . . . . .	3
<b>3 File Index</b>	<b>5</b>
3.1 File List . . . . .	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 dice Class Reference . . . . .	7
4.1.1 Detailed Description . . . . .	7
4.1.2 Constructor & Destructor Documentation . . . . .	8
4.1.2.1 dice() . . . . .	8
4.1.2.2 ~dice() . . . . .	8
4.1.3 Member Function Documentation . . . . .	8
4.1.3.1 addValues() . . . . .	8
4.1.3.2 GetDice() . . . . .	8
4.1.3.3 Reroll() . . . . .	9
4.1.3.4 Roll() . . . . .	9
4.1.3.5 RollMenu() . . . . .	9
4.1.3.6 WriteVector() . . . . .	10
4.1.4 Member Data Documentation . . . . .	10
4.1.4.1 heldDice . . . . .	10
4.2 die Class Reference . . . . .	10
4.2.1 Detailed Description . . . . .	11
4.2.2 Constructor & Destructor Documentation . . . . .	11
4.2.2.1 die() . . . . .	11
4.2.2.2 ~die() . . . . .	11
4.2.3 Member Function Documentation . . . . .	11
4.2.3.1 GetNumber() . . . . .	12
4.2.3.2 Roll() . . . . .	12
4.2.3.3 set() . . . . .	12
4.2.4 Member Data Documentation . . . . .	12
4.2.4.1 number . . . . .	12
4.3 MainMenu Class Reference . . . . .	13
4.3.1 Detailed Description . . . . .	13
4.3.2 Constructor & Destructor Documentation . . . . .	13
4.3.2.1 MainMenu() . . . . .	13
4.3.2.2 ~MainMenu() . . . . .	14
4.3.3 Member Function Documentation . . . . .	14
4.3.3.1 DrawMenu() . . . . .	14
4.3.3.2 getChosenFile() . . . . .	14

4.3.3.3 ShowRules()	14
4.3.4 Member Data Documentation	15
4.3.4.1 answer	15
4.4 scorecard Class Reference	15
4.4.1 Detailed Description	16
4.4.2 Constructor & Destructor Documentation	16
4.4.2.1 scorecard() [1/2]	16
4.4.2.2 scorecard() [2/2]	16
4.4.2.3 ~scorecard()	16
4.4.3 Member Function Documentation	16
4.4.3.1 getIsWritten()	17
4.4.3.2 getScore()	17
4.4.3.3 getScorecard()	17
4.4.3.4 getSet()	18
4.4.3.5 isFull()	18
4.4.3.6 setSet()	18
4.4.3.7 writePlayersScorecard()	18
4.4.3.8 writeScore()	19
4.4.4 Friends And Related Function Documentation	19
4.4.4.1 operator<<	19
4.4.5 Member Data Documentation	20
4.4.5.1 isWritten	20
4.4.5.2 scores	20
4.4.5.3 wholeScorecardSize	20
4.4.5.4 yourDice	20
4.5 YahtzeeGame Class Reference	20
4.5.1 Detailed Description	21
4.5.2 Constructor & Destructor Documentation	21
4.5.2.1 YahtzeeGame()	21
4.5.2.2 ~YahtzeeGame()	21
4.5.3 Member Function Documentation	22
4.5.3.1 getPlayerCount()	22
4.5.3.2 getPlayersScorecard()	22
4.5.3.3 getScorecard()	22
4.5.3.4 getScorecards()	22
4.5.3.5 isGameDone()	23
4.5.3.6 loadTheGame()	23
4.5.3.7 writeScorecardToFile() [1/2]	23
4.5.3.8 writeScorecardToFile() [2/2]	24
4.5.3.9 YahtzeeGameEnd()	24
4.5.3.10 YahtzeeGameStart() [1/2]	24
4.5.3.11 YahtzeeGameStart() [2/2]	24

4.5.3.12 YahtzeeMenu()	25
4.5.4 Member Data Documentation	25
4.5.4.1 players	25
4.5.4.2 scorecardHolder	25
<b>5 File Documentation</b>	<b>27</b>
5.1 dice.cpp File Reference	27
5.2 Dice.h File Reference	27
5.3 Dice.h	27
5.4 die.cpp File Reference	28
5.5 Die.h File Reference	28
5.6 Die.h	28
5.7 Functions.cpp File Reference	28
5.7.1 Function Documentation	28
5.7.1.1 bubbleSort()	29
5.7.1.2 comparePairs()	29
5.8 FunctionsH.h File Reference	29
5.8.1 Function Documentation	29
5.8.1.1 bubbleSort()	30
5.8.1.2 comparePairs()	30
5.9 FunctionsH.h	30
5.10 Gra_w_Kosci.cpp File Reference	30
5.10.1 Function Documentation	31
5.10.1.1 main()	31
5.11 MainMenu.cpp File Reference	31
5.12 MainMenu.h File Reference	31
5.13 MainMenu.h	31
5.14 scorecard.cpp File Reference	32
5.14.1 Function Documentation	32
5.14.1.1 operator<<()	32
5.15 Scorecard.h File Reference	32
5.16 Scorecard.h	33
5.17 yahtzee.cpp File Reference	33
5.18 Yahtzee.h File Reference	33
5.19 Yahtzee.h	33
<b>Index</b>	<b>35</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

die . . . . .	10
dice . . . . .	7
scorecard . . . . .	15
YahtzeeGame . . . . .	20
MainMenu . . . . .	13





## Chapter 2

# Class Index

### 2.1 Class List

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

<b>dice</b>	Klasa opisująca zestaw koci gracza . . . . .	7
<b>die</b>	Klasa będąca pojedynczą koscia do gry . . . . .	10
<b>MainMenu</b>	Klasa odpowiadająca za Menu tytułowe gry . . . . .	13
<b>scorecard</b>	Klasa zajmująca się kartą wyników gracza . . . . .	15
<b>YahtzeeGame</b>	Klasa która jest całym kontrolerem gry w koci . . . . .	20



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<b>dice.cpp</b>	27
<b>Dice.h</b>	27
<b>die.cpp</b>	28
<b>Die.h</b>	28
<b>Functions.cpp</b>	28
<b>FunctionsH.h</b>	29
<b>Gra_w_Kosci.cpp</b>	30
<b>MainMenu.cpp</b>	31
<b>MainMenu.h</b>	31
<b>scorecard.cpp</b>	32
<b>Scorecard.h</b>	32
<b>yahtzee.cpp</b>	33
<b>Yahtzee.h</b>	33



## Chapter 4

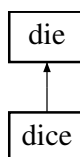
# Class Documentation

### 4.1 dice Class Reference

Klasa opisujaca zestaw kosci gracza.

```
#include <Dice.h>
```

Inheritance diagram for dice:



#### Public Member Functions

- void **Reroll** (std::vector< int > &indexes, bool &canReroll, int &rerollCounter)
- void **RollMenu** (std::vector< int > &indexes, bool &canReroll, int &rerollCounter, std::string &answer, std::shared\_ptr< **dice** > &currentPlayerDice)
- void **Roll** ()
- int **addValues** (int &square, int &rerollCTR)
- void **WriteVector** ()
- std::vector< **die** > & **GetDice** ()
- **dice** ()
- ~**dice** ()

#### Private Attributes

- std::vector< **die** > **heldDice**

#### 4.1.1 Detailed Description

Klasa opisujaca zestaw kosci gracza.

Przerzuca kosci, sumuje wartosci.

## 4.1.2 Constructor & Destructor Documentation

### 4.1.2.1 `dice()`

```
dice::dice ( )
```

Konstruktor. Wrzuca do wektora 5 kosci D6.

### 4.1.2.2 `~dice()`

```
dice::~~dice ( )
```

Destruktor. Nie za wiele robi ale sie stara i za to go cenimy.

## 4.1.3 Member Function Documentation

### 4.1.3.1 `addValues()`

```
int dice::addValues (
    int & square,
    int & rerollCTR )
```

Funkcja sumująca oczka na kociach.

#### Parameters

<i>square</i>	Numer pola do ktorego gracz chce wpisac rzut
<i>rerollCTR</i>	Licznik przerzutow jakie gracz wykonal

#### See also

**bubbleSort** (p. 28)

### 4.1.3.2 `GetDice()`

```
std::vector< die > & dice::GetDice ( )
```

Getter wektora z kociami.

#### Returns

Wektor z kosciami

#### See also

**heldDice** (p. 10)

#### 4.1.3.3 Reroll()

```
void dice::Reroll (
    std::vector< int > & indexes,
    bool & canReroll,
    int & rerollCounter )
```

Funkcja przerzucająca kosci gracza.

#### Parameters

<i>indexes</i>	Indeksy kosci do przerzutu
<i>canReroll</i>	Zmienna boolowska pilnujaca czy gracz moze jeszcze przerzucac
<i>rerollCounter</i>	Licznik przerzutow

#### 4.1.3.4 Roll()

```
void dice::Roll ( ) [virtual]
```

Funkcja rzucająca kosciami poraz pierwszy.

Reimplemented from **die** (p. 12).

#### 4.1.3.5 RollMenu()

```
void dice::RollMenu (
    std::vector< int > & indexes,
    bool & canReroll,
    int & rerollCounter,
    std::string & answer,
    std::shared_ptr< dice > & currentPlayerDice )
```

Funkcja odpowiadająca za uzyskanie od gracza indeksow kosci do przerzutu.

## Parameters

<i>indexes</i>	Indeksy kosci do przerzutu
<i>canReroll</i>	Zmienna boolowska pilnujaca czy gracz moze jeszcze przerzucac
<i>rerollCounter</i>	Licznik przerzutow
<i>answer</i>	Odpowiedz uzytkownika na pytanie co do przerzutu
<i>currentPlayerDice</i>	Wskaźnik na kosci gracza

## See also

**Reroll** (p. 9)

## 4.1.3.6 WriteVector()

```
void dice::WriteVector ( )
```

Funkcja wypisujaca wektor na ekranie

## 4.1.4 Member Data Documentation

## 4.1.4.1 heldDice

```
std::vector< die> dice::heldDice [private]
```

Wektor z obiektami klasy "Die" czyli kosciami.

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

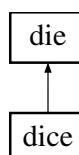
- **Dice.h**
- **dice.cpp**

## 4.2 die Class Reference

Klasa bedaca pojedyncza koscia do gry.

```
#include <Die.h>
```

Inheritance diagram for die:





## Public Member Functions

- virtual void **Roll** ()
- void **set** (int &numberOnDie)
- int **GetNumber** ()
- **die** ()
- **~die** ()

## Private Attributes

- int **number**

### 4.2.1 Detailed Description

Klasa bedaca pojedyncza koscia do gry.

Typ kosci: D6.

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 die()

```
die::die ( )
```

Konstruktor. Ustawia numer na 1.

See also

**number** (p. 12)

#### 4.2.2.2 ~die()

```
die::~~die ( )
```

Destruktor. Nie za wiele robi ale i tak go kochamy.

### 4.2.3 Member Function Documentation

#### 4.2.3.1 GetNumber()

```
int die::GetNumber ( )
```

Funkcja zwracająca numer na kosci.

##### Returns

Liczba oczek na kosci

#### 4.2.3.2 Roll()

```
void die::Roll ( ) [virtual]
```

Wirtualna funkcja do losowania ilosci oczek.

Reimplemented in **dice** (p.9).

#### 4.2.3.3 set()

```
void die::set (
    int & numberOnDie )
```

Funkcja ustawiajaca numer oczek na podany.

##### Parameters

<i>numberOnDie</i>	Podana liczba oczek
--------------------	---------------------

### 4.2.4 Member Data Documentation

#### 4.2.4.1 number

```
int die::number [private]
```

Zmienna przechowujaca ilosc oczek ktore wypadly.

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

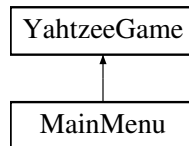
- **Die.h**
- **die.cpp**

## 4.3 MainMenu Class Reference

Klasa odpowiadajaca za Menu tytulowe gry.

```
#include <MainMenu.h>
```

Inheritance diagram for MainMenu:



### Public Member Functions

- **MainMenu** ()
- **~MainMenu** ()
- void **DrawMenu** ()
- void **ShowRules** ()
- std::string **getChosenFile** (std::string index)

### Private Attributes

- std::string **answer**

#### 4.3.1 Detailed Description

Klasa odpowiadajaca za Menu tytulowe gry.

Nie robi zbyt wiele oprócz rysowania Menu Tytulowego na ekranie i odpalania gry.

#### 4.3.2 Constructor & Destructor Documentation

##### 4.3.2.1 MainMenu()

```
MainMenu::MainMenu ( )
```

Konstruktor. Ustawia zmienna answer na "".

See also

**answer** (p. 15)

#### 4.3.2.2 ~MainMenu()

```
MainMenu::~MainMenu ( )
```

Dekonstruktor. Nic ciekawego nie robi.

### 4.3.3 Member Function Documentation

#### 4.3.3.1 DrawMenu()

```
void MainMenu::DrawMenu ( )
```

Funkcja wypisująca Menu Tytułowe gry. W zależności od wyboru gracza wywołuje poszczególne wersje gry.

See also

**YahtzeeGameStart** (p. 24)

**loadTheGame** (p. 23)

#### 4.3.3.2 getChosenFile()

```
std::string MainMenu::getChosenFile (
    std::string index )
```

Funkcja zwracająca poszczególny plik zapisu.

Parameters

<i>index</i>	Numer zapisanego pliku od 1 do 3.
--------------	-----------------------------------

Returns

Nazwa pliku zapisu

#### 4.3.3.3 ShowRules()

```
void MainMenu::ShowRules ( )
```

Funkcja wypisująca na ekranie zasady gry w kosci.

### 4.3.4 Member Data Documentation

#### 4.3.4.1 answer

```
std::string MainMenu::answer [private]
```

Zmienna przechowująca odpowiedz gracza.

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

- **MainMenu.h**
- **MainMenu.cpp**

## 4.4 scorecard Class Reference

Klasa zajmująca się kartą wyników gracza.

```
#include <Scorecard.h>
```

### Public Member Functions

- void **writeScore** (int &index, **dice** &diceSet, int &rerollCTR)
- void **setSet** (std::shared\_ptr< **dice** > otherSet)
- bool **isFull** ()
- void **writePlayersScorecard** (std::string &playerName)
- std::shared\_ptr< **dice** > **getSet** ()
- int \* **getScorecard** ()
- bool \* **getIsWritten** ()
- int **getScore** ()
- **scorecard** ()
- **scorecard** ( **scorecard** &otherScorecard)
- **~scorecard** ()

### Private Attributes

- int **scores** [ **wholeScorecardSize** ] { 0 }
- bool **isWritten** [ **wholeScorecardSize** ] { false }
- std::shared\_ptr< **dice** > **yourDice**

### Static Private Attributes

- static const int **wholeScorecardSize** = 13

## Friends

- `std::ostream & operator<< (std::ostream &os, scorecard &tmp)`

### 4.4.1 Detailed Description

Klasa zajmująca się kartą wyników gracza.

Zapisuje wynik rzutu, sumuje punkty i wypisuje kartę na ekranie.

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 `scorecard()` [1/2]

```
scorecard::scorecard ( )
```

Konstruktor. Ustawia wskaźnik na nowe kości.

#### 4.4.2.2 `scorecard()` [2/2]

```
scorecard::scorecard (
    scorecard & otherScorecard )
```

Konstruktor kopiujący. Ustawia wszystkie dane na takie same jak w podanej karcie.

#### Parameters

<i>otherScorecard</i>	Inna karta wyników
-----------------------	--------------------

#### 4.4.2.3 `~scorecard()`

```
scorecard::~scorecard ( )
```

Destruktor. Nie za wiele robi.

### 4.4.3 Member Function Documentation

#### 4.4.3.1 getIsWritten()

```
bool * scorecard::getIsWritten ( )
```

Funkcja zwracająca tablice z info co do zapisania karty wynikow.

##### Returns

Tablica isWritten

##### See also

**isWritten** (p. 20)

#### 4.4.3.2 getScore()

```
int scorecard::getScore ( )
```

Funkcja zwracająca sume punktow.

##### Returns

Zsumowane punkty z karty wynikow

#### 4.4.3.3 getScorecard()

```
int * scorecard::getScorecard ( )
```

Funkcja zwracająca tablice wynikow.

##### Returns

Tablica scores

##### See also

**scores** (p. 20)

#### 4.4.3.4 getSet()

```
std::shared_ptr< dice > scorecard::getSet ( )
```

Metoda zwracająca wskaźnik na kości gracza.

##### Returns

Wskaźnik na kości

#### 4.4.3.5 isFull()

```
bool scorecard::isFull ( )
```

Funkcja sprawdzająca czy karty są pełne. Sprawdza czy wszystkie pola w tablicy `isWritten` są ustawione na `true`.

##### Returns

True albo false. Zazwyczaj false

##### See also

**isWritten** (p. 20)

#### 4.4.3.6 setSet()

```
void scorecard::setSet (
    std::shared_ptr< dice > otherSet )
```

Funkcja ustawiająca wskaźnik na podany zestaw kości.

##### Parameters

<i>otherSet</i>	Wskaźnik na inne kości
-----------------	------------------------

#### 4.4.3.7 writePlayersScorecard()

```
void scorecard::writePlayersScorecard (
    std::string & playerName )
```

Funkcja wypisująca kartę podanego gracza.



## Parameters

<i>playerName</i>	Imie gracza
-------------------	-------------

## 4.4.3.8 writeScore()

```
void scorecard::writeScore (
    int & index,
    dice & diceSet,
    int & rerollCTR )
```

Funkcja wpisująca do podanej komórki wynik rzutu.

## Parameters

<i>index</i>	Indeks pola do wpisania
<i>diceSet</i>	Zestaw rzuconych kosci
<i>rerollCTR</i>	Ilosc przerzutow jakie gracz wykonął podczas tury

## See also

[addValues](#)  
[scores](#) (p. 20)

## 4.4.4 Friends And Related Function Documentation

## 4.4.4.1 operator&lt;&lt;

```
std::ostream & operator<< (
    std::ostream & os,
    scorecard & tmp ) [friend]
```

Operator wypisywania. Wypisuje ładnie wyglądająca i opisana karte wynikow aktualnego gracza.

## Parameters

<i>os</i>	Zwykly ostream
<i>tmp</i>	Karta wynikow gracza

## Returns

Ostream os

## 4.4.5 Member Data Documentation

### 4.4.5.1 isWritten

```
bool scorecard::isWritten[ wholeScorecardSize] { false } [private]
```

Tablica z informacją czy coś zostało wpisane w daną komórkę w karcie.

### 4.4.5.2 scores

```
int scorecard::scores[ wholeScorecardSize] { 0 } [private]
```

Tablica z wynikami gracza.

### 4.4.5.3 wholeScorecardSize

```
const int scorecard::wholeScorecardSize = 13 [static], [private]
```

Wielkość karty wyników.

### 4.4.5.4 yourDice

```
std::shared_ptr< dice> scorecard::yourDice [private]
```

Wskaźnik na zestaw kości gracza.

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

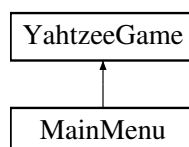
- **Scorecard.h**
- **scorecard.cpp**

## 4.5 YahtzeeGame Class Reference

Klasa która jest całym kontrolerem gry w kości.

```
#include <Yahtzee.h>
```

Inheritance diagram for YahtzeeGame:



## Public Member Functions

- `std::map< std::string, std::shared_ptr< scorecard > > & getScorecards ()`
- `std::shared_ptr< scorecard > getScorecard (std::string &name)`
- `int getPlayerCount ()`
- `void getPlayersScorecard ()`
- `void YahtzeeGameStart ()`
- `void YahtzeeGameStart (int roundFromFile, std::string currentPlayerS)`
- `int YahtzeeMenu (std::shared_ptr< scorecard > &currentPlayerScorecard, std::shared_ptr< dice > currentPlayerDice, int currentPlayer, int crrRound)`
- `bool isGameDone ()`
- `void YahtzeeGameEnd ()`
- `void writeScorecardToFile ()`
- `void writeScorecardToFile (std::string &saveIndex, int currentPlayer, std::shared_ptr< dice > currentPlayerDice, int crrRound)`
- `void loadTheGame (std::string &filename)`
- `YahtzeeGame ()`
- `~YahtzeeGame ()`

## Private Attributes

- `std::map< std::string, std::shared_ptr< scorecard > > scorecardHolder`
- `std::vector< std::string > players`

### 4.5.1 Detailed Description

Klasa która jest całym kontrolerem gry w kosci.

Zajmuje się ona wszystkim: Pilnuje gry, czynów graczy, wyników itp. Zdecydowanie największa i najbardziej rozbudowana klasa w programie

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 YahtzeeGame()

```
YahtzeeGame::YahtzeeGame ( )
```

Konstruktor. Tyle.

#### 4.5.2.2 ~YahtzeeGame()

```
YahtzeeGame::~~YahtzeeGame ( )
```

Dekonstruktor. Mowi ze gra sie zakonczyla.

### 4.5.3 Member Function Documentation

#### 4.5.3.1 getPlayerCount()

```
int YahtzeeGame::getPlayerCount ( )
```

Funkcja zwracajaca ilosc graczy.

##### Returns

Liczba grajacych osob

#### 4.5.3.2 getPlayersScorecard()

```
void YahtzeeGame::getPlayersScorecard ( )
```

Funkcja zajmujaca sie wypisywaniem karty podanego gracza.

#### 4.5.3.3 getScorecard()

```
std::shared_ptr< scorecard > YahtzeeGame::getScorecard (
    std::string & name )
```

Funkcja zwracajaca wskaznik na karte podanego gracza.

##### Parameters

<i>name</i>	Imie gracza
-------------	-------------

##### Returns

Wskaznik na karte tego gracza

##### See also

**scorecardHolder** (p. 25)

#### 4.5.3.4 getScorecards()

```
std::map< std::string, std::shared_ptr< scorecard > > & YahtzeeGame::getScorecards ( )
```

Funkcja zwracajaca mape.

See also

**scorecardHolder** (p. 25)

Returns

Mapa z wskaźnikami

#### 4.5.3.5 isGameDone()

```
bool YahtzeeGame::isGameDone ( )
```

Funkcja sprawdzająca czy gre należy zakonczyc. Sprawdza czy wszystkie pola w kartach graczy zostały zapisane.

Returns

True albo false. W wiekszosci przypadkow False

#### 4.5.3.6 loadTheGame()

```
void YahtzeeGame::loadTheGame (
    std::string & filename )
```

Funkcja wczytująca zapisany stan gry.

Parameters

<i>filename</i>	Nazwa pliku z ktorego wczytujemy stan gry
-----------------	-------------------------------------------

#### 4.5.3.7 writeScorecardToFile() [1/2]

```
void YahtzeeGame::writeScorecardToFile ( )
```

Funkcja wpisująca karty graczy i wyniki do pliku txt.

See also

**YahtzeeGameEnd** (p. 24)

#### 4.5.3.8 writeScorecardToFile() [2/2]

```
void YahtzeeGame::writeScorecardToFile (
    std::string & saveIndex,
    int currentPlayer,
    std::shared_ptr< dice > currentPlayerDice,
    int crrRound )
```

Przeciazona funkcja ktora zapisuje stan gry. Zapisuje runde, aktualnego gracza i dane kart wynikow graczy.

##### Parameters

<i>saveIndex</i>	Indeks do ktorego pliku zapisac stan gry
<i>currentPlayer</i>	Indeks aktualnego gracza
<i>currentPlayerDice</i>	Zestaw kosci aktualnego gracza
<i>crrRound</i>	Aktualna runda

##### See also

**YahtzeeMenu** (p. 25)

#### 4.5.3.9 YahtzeeGameEnd()

```
void YahtzeeGame::YahtzeeGameEnd ( )
```

Funkcja zajmujaca sie zakonczeniem gry. Zpisuje i sortuje wyniki, wypisuje je z fanfarami i zapisuje je do pliku txt.

##### See also

**writeScorecardToFile** (p. 23)

**comparePairs** (p. 29)

#### 4.5.3.10 YahtzeeGameStart() [1/2]

```
void YahtzeeGame::YahtzeeGameStart ( )
```

Funkcja startujaca gre. Startuje kompletnie nowa gra z wybrana iloscia osob.

#### 4.5.3.11 YahtzeeGameStart() [2/2]

```
void YahtzeeGame::YahtzeeGameStart (
    int roundFromFile,
    std::string currentPlayerS )
```

Przeciazona funkcja startujaca gre. Przyjmuje parametry zwiazane z zapisana wczesniej gra.

## Parameters

<i>roundFromFile</i>	Aktualna runda czytana z pliku zapisu
<i>currentPlayerS</i>	Imie aktualnego gracza (tego który zapisal)

## See also

**loadTheGame** (p. 23)

## 4.5.3.12 YahtzeeMenu()

```
int YahtzeeGame::YahtzeeMenu (
    std::shared_ptr< scorecard > & currentPlayerScorecard,
    std::shared_ptr< dice > currentPlayerDice,
    int currentPlayer,
    int crrRound )
```

Metoda przedstawiajaca dozwolone czynnosci podczas rozgrywki. Gracz podczas swojej tury moze przerzucic wybrane kosci 3 razy, podejrzec karte innego gracza, zapisac stan rozgrywki, badz zakonczyc ja wczesniej.

## Parameters

<i>currentPlayerScorecard</i>	Karta aktualnego gracza
<i>currentPlayerDice</i>	Zestaw kosci aktualnego gracza
<i>currentPlayer</i>	Indeks aktualnego gracza
<i>crrRound</i>	Numer rundy

## Returns

Liczba przerzutow jaka gracz wykonal

## 4.5.4 Member Data Documentation

## 4.5.4.1 players

```
std::vector<std::string> YahtzeeGame::players [private]
```

Wektor z imionami graczy.

## 4.5.4.2 scorecardHolder

```
std::map<std::string, std::shared_ptr< scorecard > > YahtzeeGame::scorecardHolder [private]
```

Mapa z wskaznikami na karty graczy.

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

- **Yahtzee.h**
- **yahtzee.cpp**





## Chapter 5

# File Documentation

### 5.1 dice.cpp File Reference

```
#include "Dice.h"
```

### 5.2 Dice.h File Reference

```
#include "Die.h"  
#include <map>  
#include <Windows.h>  
#include "FunctionsH.h"
```

#### Classes

- class **dice**

*Klasa opisujaca zestaw kosci gracza.*

### 5.3 Dice.h

**Go to the documentation of this file.**

```
1 #pragma once  
2 #include "Die.h"  
3 #include <map>  
4 #include <Windows.h>  
5 #include "FunctionsH.h"  
10 class dice : public die  
11 {  
12     std::vector<die> heldDice;  
14 public:  
20     void Reroll(std::vector<int>& indexes, bool& canReroll, int& rerollCounter);  
29     void RollMenu(std::vector<int>& indexes, bool& canReroll, int& rerollCounter, std::string& answer,  
    std::shared_ptr<dice>& currentPlayerDice);  
32     void Roll(); //dorobic cos z funkcja wirtualna moze? Tak zeby byl polimorfizm (To chyba jest to)  
38     int addValues(int& square, int& rerollCTR);  
41     void WriteVector();  
46     std::vector<die>& GetDice();  
49     dice();  
52     ~dice();  
53 };
```

## 5.4 die.cpp File Reference

```
#include "Die.h"
```

## 5.5 Die.h File Reference

```
#include <iostream>
#include <random>
#include <chrono>
#include <memory>
```

### Classes

- class **die**

*Klasa bedaca pojedyncza koscia do gry.*

## 5.6 Die.h

**Go to the documentation of this file.**

```
1 #pragma once
2 #include <iostream>
3 #include <random>
4 #include <chrono>
5 #include <memory>
10 class die
11 {
12     int number;
14 public:
17     virtual void Roll();
21     void set(int& numberOnDie);
25     int GetNumber();
29     die();
32     ~die();
33 };
34
35
```

## 5.7 Functions.cpp File Reference

```
#include "FunctionsH.h"
```

### Functions

- void **bubbleSort** (int arr[], int n)
- bool **comparePairs** (const std::pair< std::string, int > &a, std::pair< std::string, int > &b)

### 5.7.1 Function Documentation

### 5.7.1.1 bubbleSort()

```
void bubbleSort (
    int arr[],
    int n )
```

Bubble Sort wykorzystany na potrzeby sumowania ilosci oczek na kosciach.

#### Parameters

<i>arr</i>	Tablica z oczkami ktore wypadly na kosciach
<i>n</i>	Rozmiar tablicy "arr"

### 5.7.1.2 comparePairs()

```
bool comparePairs (
    const std::pair< std::string, int > & a,
    std::pair< std::string, int > & b )
```

Funkcja porownujaca pary <string, int>.

#### Parameters

<i>a</i>	Para pierwsza
<i>b</i>	Para druga

## 5.8 FunctionsH.h File Reference

```
#include <vector>
#include <string>
#include <istream>
#include <sstream>
```

### Functions

- void **bubbleSort** (int arr[], int n)
- bool **comparePairs** (const std::pair< std::string, int > &a, std::pair< std::string, int > &b)

### 5.8.1 Function Documentation

### 5.8.1.1 bubbleSort()

```
void bubbleSort (
    int arr[],
    int n )
```

Bubble Sort wykorzystany na potrzeby sumowania ilosci oczek na kosciach.

#### Parameters

<i>arr</i>	Tablica z oczkami ktore wypadly na kosciach
<i>n</i>	Rozmiar tablicy "arr"

### 5.8.1.2 comparePairs()

```
bool comparePairs (
    const std::pair< std::string, int > & a,
    std::pair< std::string, int > & b )
```

Funkcja porownujaca pary <string, int>.

#### Parameters

<i>a</i>	Para pierwsza
<i>b</i>	Para druga

## 5.9 FunctionsH.h

**Go to the documentation of this file.**

```
1 #pragma once
2 #include <vector>
3 #include <string>
4 #include <istream>
5 #include <sstream>
10 void bubbleSort(int arr[], int n);
15 bool comparePairs(const std::pair<std::string, int>& a, std::pair<std::string, int>& b);
```

## 5.10 Gra\_w\_Kosci.cpp File Reference

```
#include "MainMenu.h"
```

### Functions

- int **main** ()

## 5.10.1 Function Documentation

### 5.10.1.1 main()

```
int main ( )
```

Program jest komputerowa wersja popularnej gry w kosci (Yahtzee)

Date

2023-06-10

Author

Kacper Sikorski, I rok Informatyki, 3 grupa dziekanska, sekcja 32 PK

## 5.11 MainMenu.cpp File Reference

```
#include "MainMenu.h"
```

## 5.12 MainMenu.h File Reference

```
#include "Yahtzee.h"
```

### Classes

- class **MainMenu**

*Klasa odpowiadajaca za Menu tytulowe gry.*

## 5.13 MainMenu.h

**Go to the documentation of this file.**

```
1 #pragma once
2 #include "Yahtzee.h"
3
4 class MainMenu : public YahtzeeGame
5 {
6     std::string answer;
7
8 public:
9     MainMenu();
10    ~MainMenu();
11    void DrawMenu();
12    void ShowRules();
13    std::string getChosenFile(std::string index);
14 };
15
```

## 5.14 scorecard.cpp File Reference

```
#include "Scorecard.h"
```

### Functions

- `std::ostream & operator<< (std::ostream &os, scorecard &tmp)`

### 5.14.1 Function Documentation

#### 5.14.1.1 `operator<<()`

```
std::ostream & operator<< (  
    std::ostream & os,  
    scorecard & tmp )
```

Operator wypisywania. Wypisuje ładnie wyglądającą i opisaną kartę wyników aktualnego gracza.

#### Parameters

<i>os</i>	Zwykły ostream
<i>tmp</i>	Karta wyników gracza

#### Returns

Ostream os

## 5.15 Scorecard.h File Reference

```
#include "Dice.h"
```

### Classes

- class **scorecard**  
*Klasa zajmująca się kartą wyników gracza.*

## 5.16 Scorecard.h

Go to the documentation of this file.

```
1 #pragma once
2 #include "Dice.h"
3
4 class scorecard
5 {
6     const static int wholeScorecardSize = 13;
7     int scores[wholeScorecardSize]{ 0 };
8     bool isWritten[wholeScorecardSize]{ false };
9     std::shared_ptr<dice> yourDice;
10 public:
11     void writeScore(int& index, dice& diceSet, int& rerollCTR);
12
13     void setSet(std::shared_ptr<dice> otherSet);
14     bool isFull();
15     void writePlayersScorecard(std::string& playerName);
16     std::shared_ptr<dice> getSet();
17     int* getScorecard();
18     bool* getIsWritten();
19     int getScore();
20     scorecard();
21     scorecard(scorecard& otherScorecard);
22     ~scorecard();
23     friend std::ostream& operator<<(std::ostream& os, scorecard& tmp);
24 };
25
```

## 5.17 yahtzee.cpp File Reference

```
#include "Yahtzee.h"
```

## 5.18 Yahtzee.h File Reference

```
#include "Scorecard.h"
#include <stdlib.h>
#include <fstream>
```

### Classes

- class **YahtzeeGame**

*Klasa która jest całym kontrolerem gry w kosci.*

## 5.19 Yahtzee.h

Go to the documentation of this file.

```
1 #pragma once
2 #include "Scorecard.h"
3 #include <stdlib.h>
4 #include <fstream>
5
6 class YahtzeeGame
7 {
8     std::map<std::string, std::shared_ptr<scorecard>> scorecardHolder;
9     std::vector<std::string> players;
10 public:
11     std::map<std::string, std::shared_ptr<scorecard>>& getScorecards();
12     std::shared_ptr<scorecard> getScorecard(std::string& name);
13     int getPlayerCount();
14     void getPlayersScorecard();
15 };
16
```

```
37     void YahtzeeGameStart();
44     void YahtzeeGameStart(int roundFromFile, std::string currentPlayerS);
53     int YahtzeeMenu(std::shared_ptr<scorecard>& currentPlayerScorecard, std::shared_ptr<dice>
currentPlayerDice, int currentPlayer, int crrRound);
58     bool isGameDone();
64     void YahtzeeGameEnd();
68     void writeScorecardToFile();
77     void writeScorecardToFile(std::string& saveIndex, int currentPlayer, std::shared_ptr<dice>
currentPlayerDice, int crrRound);
81     void loadTheGame(std::string& filename);
84     YahtzeeGame();
87     ~YahtzeeGame();
88 };
```



# Index

- ~MainMenu
  - MainMenu, 13
- ~YahtzeeGame
  - YahtzeeGame, 21
- ~dice
  - dice, 8
- ~die
  - die, 11
- ~scorecard
  - scorecard, 16
- addValues
  - dice, 8
- answer
  - MainMenu, 15
- bubbleSort
  - Functions.cpp, 28
  - FunctionsH.h, 29
- comparePairs
  - Functions.cpp, 29
  - FunctionsH.h, 30
- dice, 7
  - ~dice, 8
  - addValues, 8
  - dice, 8
  - GetDice, 8
  - heldDice, 10
  - Reroll, 9
  - Roll, 9
  - RollMenu, 9
  - WriteVector, 10
- dice.cpp, 27
- Dice.h, 27
- die, 10
  - ~die, 11
  - die, 11
  - GetNumber, 11
  - number, 12
  - Roll, 12
  - set, 12
- die.cpp, 28
- Die.h, 28
- DrawMenu
  - MainMenu, 14
- Functions.cpp, 28
  - bubbleSort, 28
  - comparePairs, 29
- FunctionsH.h, 29
  - bubbleSort, 29
  - comparePairs, 30
- getChosenFile
  - MainMenu, 14
- GetDice
  - dice, 8
- getIsWritten
  - scorecard, 16
- GetNumber
  - die, 11
- getPlayerCount
  - YahtzeeGame, 22
- getPlayersScorecard
  - YahtzeeGame, 22
- getScore
  - scorecard, 17
- getScorecard
  - scorecard, 17
  - YahtzeeGame, 22
- getScorecards
  - YahtzeeGame, 22
- getSet
  - scorecard, 17
- Gra\_w\_Kosci.cpp, 30
  - main, 31
- heldDice
  - dice, 10
- isFull
  - scorecard, 18
- isGameDone
  - YahtzeeGame, 23
- isWritten
  - scorecard, 20
- loadTheGame
  - YahtzeeGame, 23
- main
  - Gra\_w\_Kosci.cpp, 31
- MainMenu, 13
  - ~MainMenu, 13
  - answer, 15
  - DrawMenu, 14
  - getChosenFile, 14
  - MainMenu, 13
  - ShowRules, 14
- MainMenu.cpp, 31

- MainMenu.h, 31
- number
  - die, 12
- operator<<
  - scorecard, 19
  - scorecard.cpp, 32
- players
  - YahtzeeGame, 25
- Reroll
  - dice, 9
- Roll
  - dice, 9
  - die, 12
- RollMenu
  - dice, 9
- scorecard, 15
  - ~scorecard, 16
  - getIsWritten, 16
  - getScore, 17
  - getScorecard, 17
  - getSet, 17
  - isFull, 18
  - isWritten, 20
  - operator<<, 19
  - scorecard, 16
  - scores, 20
  - setSet, 18
  - wholeScorecardSize, 20
  - writePlayersScorecard, 18
  - writeScore, 19
  - yourDice, 20
- scorecard.cpp, 32
  - operator<<, 32
- Scorecard.h, 32
- scorecardHolder
  - YahtzeeGame, 25
- scores
  - scorecard, 20
- set
  - die, 12
- setSet
  - scorecard, 18
- ShowRules
  - MainMenu, 14
- wholeScorecardSize
  - scorecard, 20
- writePlayersScorecard
  - scorecard, 18
- writeScore
  - scorecard, 19
- writeScorecardToFile
  - YahtzeeGame, 23
- WriteVector
  - dice, 10
- yahtzee.cpp, 33
- Yahtzee.h, 33
- YahtzeeGame, 20
  - ~YahtzeeGame, 21
  - getPlayerCount, 22
  - getPlayersScorecard, 22
  - getScorecard, 22
  - getScorecards, 22
  - isGameDone, 23
  - loadTheGame, 23
  - players, 25
  - scorecardHolder, 25
  - writeScorecardToFile, 23
  - YahtzeeGame, 21
  - YahtzeeGameEnd, 24
  - YahtzeeGameStart, 24
  - YahtzeeMenu, 25
- YahtzeeGameEnd
  - YahtzeeGame, 24
- YahtzeeGameStart
  - YahtzeeGame, 24
- YahtzeeMenu
  - YahtzeeGame, 25
- yourDice
  - scorecard, 20