

Marszałek&Matuszewski&Pitala

# Przygotowanie dokumentacji

## Założenia wstępne przyjęte w realizacji projektu

Naszym założeniem było stworzenie bitmapy przedstawiającej teren w postaci mapy 2D. Kolejnym założeniem był wybór języka angielskiego w celu dotarcia do większego grona odbiorców.

## Specyfikacja danych wejściowych

Użytkownik ma możliwość wprowadzenia danych wejściowych poprzez wybór rodzaju terenu, który prezentuje mapa. Kolejną możliwością jest zmiana terenu w czasie poprzez slider jak i ilość kontur (również poprzez slider). Użytkownik może również wybrać czy chce mieć zaprezentowane wektory pola przyśpieszeń oraz czy chce wygenerować poszczególne klatki do animacji lub chce zapisać bitmapę do wydruku. Ostatnią dostępną funkcją jest wybór prezentacji animacji piłki spadającej z kierunku największego spadku spadającej z największego punktu na mapie.

## Specyfikacja danych wejściowych

Nasz program nie bazuje na danych wejściowych bezpośrednio wprowadzonych przez użytkownika (zewnętrzne pliki, napisany przez użytkownika ciąg znaków). Po wybraniu powierzchni pojawia się ona na bitmapie. Przycisk SAVE to BMP generuje plik plik do wydruku (bez kolorów) w formacie .bmp. Slider pod słowem *Timeline* zmienia ukształtowanie terenu modyfikując wzór funkcji. Slider pod słowem *Number of contours* pozwala ustalić liczbę kontur widocznych na mapie. Poniżej znajdą się przyciski służące kolejno do prezentacji wektorów pola przyśpieszeń na mapie, generowania animacji poprzez ciąg plików .bmp, prezentacji kulki która spada z najwyższego punktu na mapie. Na samym dole znajduje się przycisk *Help and about* pozwalający rozwijać wszelkie wątpliwości napotkane przez użytkownika podczas obsługi programu.

## Opis oczekiwanych danych wyjściowych

Po wybraniu powierzchni pojawia się ona na bitmapie. Przycisk SAVE to BMP generuje plik plik do wydruku (bez kolorów) w formacie .bmp. Slider pod słowem *Timeline* zmienia ukształtowanie terenu modyfikując wzór funkcji. Slider pod słowem *Number of contours* pozwala ustalić liczbę kontur widocznych na mapie. Poniżej znajdą się przyciski służące kolejno do prezentacji wektorów pola przyśpieszeń na mapie, generowania animacji poprzez ciąg plików .bmp, prezentacji kulki która spada z najwyższego punktu na mapie. Na samym dole znajduje się przycisk *Help and about* pozwalający rozwijać wszelkie wątpliwości napotkane przez użytkownika podczas obsługi programu.

## Zdefiniowanie struktur danych

W programie zaimplementowano strukturę Point przechowującą koordynaty oraz wartość w danym punkcie. Dodatkowo wykorzystaliśmy dynamiczną tablicę o wielkości wxPanela. W tablicy tej przechowywaliśmy z-tową współrzędną każdego punktu na mapie.

## Specyfikacja interfejsu użytkownika

Interfejs użytkownika został opisany w 2 i 3 punkcie.

## **Wyodrębnienie i zdefiniowanie zadań**

- implementacja ekranu powitalnego
- wyświetlanie mapy terenu
- dostarczenie możliwości wyboru ukształtowania terenu (możliwość modyfikacja ilości kontur oraz wartości czasu modyfikującej teren)
- implementacja wektorów pola przyśpieszeń na mapie
- implementacja generowania klatek do dalszej animacji
- implementacja animacji piłki na mapie
- stworzenie panelu pomocy dla użytkownika
- dostarczenie możliwości zapisu do pliku odpowiednio spreparowanej bitmapy w celu jej wydrukowania

## **Decyzja o wyborze narzędzi programistycznych**

Zdecydowano się wybrać bibliotekę WxWidgets oraz język C++. Powodem takiej decyzji była analogia zadanego projektu do ćwiczeń laboratoryjnych z przedmiotu Grafika komputerowa, jak i najlepsza znajomość tej biblioteki. Pomimo wielu ograniczeń która stawia to środowisko WxWidgets (brak plastyczności interfejsu użytkownika i brak wsparcia w dzisiejszych czasach) aproksymowany czas wykonania zadania wydawał się najkrótszy.

## **Podział pracy i analiza czasowa**

Przewidywany czas realizacji projektu: 20h.

Wspólnie:

- implementacja mapy terenu na bitmapie 7%
- implementacja wyboru ukształtowania terenu 6%
- implementacja możliwości wyboru ilości kontur na mapie 5%

Marszałek:

- implementacja podwójnego buforowania 6%
- implementacja metody największego spadku odpowiadającej za obliczenia kierunków i wartości wektorów 6%
- zapis do pliku BMP 3%
- włączanie i wyłączanie wektorów (wstępna implementacja w postaci strzałek bez grotów) 6%
- całkowita implementacja animacji kulki 6%
- generowanie dokumentacji wraz z wymaganiami postawionymi przez prowadzącego w formacie PDF 4%

Matuszewski:

- implementacja możliwości generowania klatek 7%
- implementacja możliwości zmiany czasu poprzez odpowiedni slider 6%
- implementacja ekranu powitalnego oraz zarysu panelu pomocy dla użytkownika 7%
- optymalizacja interfejsu użytkownika 4%
- generowanie dokumentacji wraz z wymaganiami postawionymi przez prowadzącego w formacie PDF 4%

Pitala:

- dokończenie implementacji wyświetlania wektorów (dodanie grotów) 5%
- sformalizowanie wzorów odpowiadających odpowiedniemu ukształtowaniu terenu (Choose a surface) 5%
- zrealizowanie panelu pomocy dla użytkownika 5%
- stworzenie angielskiej części dokumentacji (Doxygen) 7%

## Opracowanie i opis niezbędnych algorytmów

**Metoda najszybszego spadku-** algorytm numeryczny mający za zadanie znalezienie minimum zadanej funkcji celu.

Algorytm znajduje minima lokalne.

### OPIS DZIAŁANIA

Pierwszym krokiem jest wybranie punktu startowego  $\vec{r}_0 \in D$ . W punkcie tym obliczany jest antygradient  $-\nabla f(\vec{r})$  funkcji celu, który będzie stanowił kierunek poszukiwań w procedurze.

Następny punkt jest obliczany według poniższego wzoru:

$$\vec{r}_{i+1} = \vec{r}_i - h * \nabla f(\vec{r}_i)$$

Wzór na gradient:

$$\nabla f(\vec{r}) = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

Pochodne cząstkowe liczone numerycznie ze wzorów:

$$\frac{\partial f(\vec{r})}{\partial x} = \frac{f(\vec{r} + \Delta * \vec{e}_x) - f(\vec{r} - \Delta * \vec{e}_x)}{2\Delta}$$

$$\frac{\partial f(\vec{r})}{\partial y} = \frac{f(\vec{r} + \Delta * \vec{e}_y) - f(\vec{r} - \Delta * \vec{e}_y)}{2\Delta}$$

gdzie:

$\vec{e}_x, \vec{e}_y \rightarrow$  wersory układu kartezjańskiego.

Wylaczanie następnego punktu odbywa się do momentu spełnienia warunku stopu. Naszym ostatnim wyliczonym  $\vec{r}_i$  jest położenie minimum lokalnego.

## Kodowanie

Opis klas, metod oraz zmiennych znajduje się w poniższej, angielskiej części dokumentacji.

## **Testowanie**

W projekcie nie wykorzystano specjalnych klas służących do testowania. Jednak podczas implementacji nowych funkcjonalności sprawdzano nie wpływały one negatywnie na poprawną już część programu. Wykonywano serię testów sprawdzając skrajne oraz docelowe przypadki wykorzystania nowo powstałej funkcjonalności.

## **Wdrożenie, raport i wnioski**

- W przyszłości będą zaimplementowane sizery w programie (na chwilę obecną nie występują w programie)
- Program działa niezależnie od urządzenia na którym jest uruchamiany
- Powtarzają się małe fragmenty kodu, co będzie łatwo zniwelować w przyszłości
- Projekt powinien być realizowany systematycznie (dłuższe przerwy powodują brak odpowiedniej znajomości kodu)
- Program zrezygnowano w założonym na początku czasie
- Kod programu został utrzymany w jednym stylu klamrowym przy wykorzystaniu ogólnie przyjętych norm
- Odpowiedni podział obowiązków znacznie skrócił czas tworzenia projektu

# Contents

## Chapter 1

### Hierarchical Index

#### 1.1 Class Hierarchy

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

```
wxApp
    Pole_przyspieszen_czastki_GFKFrmApp ..... ???
wxDialog
    HelpDlg ..... ???
    InitialScreenDlg ..... ???
wxFrame
    Pole_przyspieszen_czastki_GFKFrm ..... ???
Generated by Doxygen
```

# Chapter 2

## Class Index

### 2.1 Class List

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

<b>HelpDlg</b> .....	??
<b>InitialScreenDlg</b> .....	??
<b>Pole_przypieszen_czastki_GFKFrm</b> .....	??
<b>Pole_przypieszen_czastki_GFKFrmApp</b> .....	?

---

# Chapter 3

## File Index

### 3.1 File List

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

<b>HelpDlg.cpp</b> .....	??
<b>HelpDlg.h</b> .....	??
<b>InitialScreenDlg.cpp</b> .....	??
<b>InitialScreenDlg.h</b> .....	??
<b>Pole przyspieszen czastki GFKApp.h</b> .....	??
<b>Pole przyspieszen czastki GFKFrm.h</b> .....	??

# Chapter 4

## Class Documentation

### 4.1 HelpDlg Class Reference

Inherits wxDialog.

#### Public Member Functions

- **HelpDlg** (wxWindow parent, wxWindowID id=1, const wxString &title=wxT("Help"), const wxPoint &pos=wxDefaultPosition, const wxSize &size=wxDefaultSize, long style=HelpDlg\_STYLE)
- void **WxButton1Click** (wxCommandEvent &event)
- void **ReturnButtonClick** (wxCommandEvent &event)

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

- **HelpDlg.h**
- **HelpDlg.cpp**

### 4.2 InitialScreenDlg Class Reference

Inherits wxDialog.

#### Public Member Functions

- **InitialScreenDlg** (wxWindow parent, wxWindowID id=1, const wxString &title=wxT("Initial Screen"), const wxPoint &pos=wxDefaultPosition, const wxSize &size=wxDefaultSize, long style=InitialScreenDlg\_STYLE)
- void **ReturnButtonClick** (wxCommandEvent &event)
- void **InitialScreenDlgActivate** (wxActivateEvent &event)
- void **ReturnButton\_1Click** (wxCommandEvent &event)
- void **InitialPanelUpdateUI** (wxUpdateUIEvent &event)

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

- **InitialScreenDlg.h**
- **InitialScreenDlg.cpp**

## 4.3 Pole\_przyspieszen\_czastki\_GFKFrm Class Reference

Inherits wxFrame.

### Public Member Functions

- **Pole\_przyspieszen\_czastki\_GFKFrm** (wxWindow parent, wxWindowID id=1, const wxString &title=wxT("Pole przyspieszen czastki GFK"), const wxPoint &pos=wxDefaultPosition, const wxSize &size=wxDefaultSize, long style=Pole\_przyspieszen\_czastki\_GFKFrm\_STYLE)
- void **MapPanelUpdateUI** (wxUpdateUIEvent &event)
- void **SaveButtonClick** (wxCommandEvent &event)
- void **TimeBarScroll** (wxScrollEvent &event)
- void **VectorsButtonClick** (wxCommandEvent &event)
- void **WxButton1Click** (wxCommandEvent &event)
- void **Pole\_przyspieszen\_czastki\_GFKFrmActivate** (wxActivateEvent &event)
- void **SurfaceChoiceSelected** (wxCommandEvent &event)
- void **ContoursBarScroll** (wxScrollEvent &event)
- void **BallButtonClick** (wxCommandEvent &event)
- void **HelpButtonClick** (wxCommandEvent &event)

### Protected Member Functions

- float **fun1** (int x, int y)
- float **fun\_1\_deriv\_x** (int x, int y)
- float **fun\_1\_deriv\_y** (int x, int y)
- float **fun2** (int x, int y)
- float **fun\_2\_deriv\_x** (int x, int y)
- float **fun\_2\_deriv\_y** (int x, int y)
- void **OnClose** (wxCloseEvent &event)
- void **CreateGUIControls** ()
- void **DrawMap** ()
- void **DrawContour** (wxDC &dc, int N, float **min**, float **max**)
- void **putContour** (wxDC &dc, int x, int y, int code)
- void **DrawVectors** ()
- void **DrawVectorsToFile** ()
- void **PrepareToSave** (wxMemoryDC &dc)
- void **DrawArrow** (wxDC &dc, int x1, int y1, int x2, int y2)
- void **DrawBall** (wxDC &dc, float x, float y)

### Protected Attributes

- int **y\_coordinates\_max**
- int **x\_coordinates\_max**
- int **NoLevels**
- int **time**
- int **scroll**
- floatboard
- float **min**
- float **max**
- bool **draw\_vectors**
- bool **draw\_ball**
- Point **points** [1400]
- wxImage **ball\_image**
- wxBitmap **ball\_bitmap**
- wxBitmap **ContourBitMap**
- wxBitmap **VectorBitMap**
- HelpDlg **help**
- InitialScreenDlg **initial\_screen**

### 4.3.1 Member Function Documentation

#### 4.3.1.1 DrawArrow()

```
void Pole_przyspieszen_czastki_GFKFrm::DrawArrow (
    wxDC & dc,
    int x1,
    int y1,
    int x2,
    int y2 )      [protected]
```

Method responsible for drawing an arrow (vector) between given points (x1,y1) and (x2,y2)

#### 4.3.1.2 DrawBall()

```
void Pole_przyspieszen_czastki_GFKFrm::DrawBall ( wxDC & dc,
    float x,
    float y )      [protected]
```

Method responsible for displaying the ball on the drawn map  
dc - drawing context on which the ball should be displayed  
x - X coordinate of the ball position  
y - Y coordinate of the ball position

#### 4.3.1.3 DrawContour()

```
void Pole_przyspieszen_czastki_GFKFrm::DrawContour ( wxDC & dc,
    int N, float
    min,
    float max )      [protected]
```

Method responsible for drawing the contours on the map  
dc - drawing context used to draw the map  
N - amount of contours that are to be placed on the map  
min - minimal height value on the map  
max - maximal height value on the map

#### 4.3.1.4 DrawMap()

```
void Pole_przyspieszen_czastki_GFKFrm::DrawMap ( )      [protected]
```

Method responsible for drawing the map

#### 4.3.1.5 DrawVectors()

```
void Pole_przyspieszen_czastki_GFKFrm::DrawVectors ( ) [protected]
```

Method responsible for drawing vectors on the contour map

#### 4.3.1.6 DrawVectorsToFile()

```
void Pole_przyspieszen_czastki_GFKFrm::DrawVectorsToFile ( ) [protected]
```

Method responsible for drawing vectors on contour map saved to a file

#### 4.3.1.7 fun1()

```
float Pole_przyspieszen_czastki_GFKFrm::fun1 ( int x,  
int y ) [protected]
```

First surface function, returning the Z-coordinate of a (x,y) point

#### 4.3.1.8 fun2()

```
float Pole_przyspieszen_czastki_GFKFrm::fun2 ( int x,  
int y ) [protected]
```

Second surface function, returning the Z-coordinate of a (x,y) point

#### 4.3.1.9 fun\_1\_deriv\_x()

```
float Pole_przyspieszen_czastki_GFKFrm::fun_1_deriv_x ( int x,  
int y ) [protected]
```

X derivative of the first surface function in the (x,y) point

#### 4.3.1.10 fun\_1\_deriv\_y()

```
float Pole_przyspieszen_czastki_GFKFrm::fun_1_deriv_y ( int x,  
int y ) [protected]
```

Y derivative of the first surface function in the (x,y) point

#### 4.3.1.11 fun\_2\_deriv\_x()

```
float Pole_przyspieszen_czastki_GFKFrm::fun_2_deriv_x ( int x,  
int y ) [protected]
```

X derivative of the second surface function in the (x,y) point

#### 4.3.1.12 fun\_2\_deriv\_y()

```
float Pole_przyspieszen_czastki_GFKFrm::fun_2_deriv_y ( int x,  
int y ) [protected]
```

Y derivative of the second surface function in the (x,y) point

#### 4.3.1.13 PrepareToSave()

```
void Pole_przyspieszen_czastki_GFKFrm::PrepareToSave (  
wxMemoryDC & dc ) [protected]
```

Method preparing a displayed map to be saved to a file as contours +  
vectors dc - drawing context from which a map is taken

#### 4.3.1.14 putContour()

```
void Pole_przyspieszen_czastki_GFKFrm::putContour ( wxDC & dc,  
int x, int  
y,  
int code ) [protected]
```

Method analysing the Z-coordinates of points and deciding if a contour should be drawn between  
them dc - drawing context used to draw the contours

x - X coordinate of a point

y - Y coordinate of a point

code - variable defining positioning of the contours

### 4.3.2 Member Data Documentation

#### 4.3.2.1 ball\_bitmap

```
wxBmp Pole_przyspieszen_czastki_GFKFrm::ball_bitmap [protected]
```

Bitmap with ball drawn on the map

#### 4.3.2.2 ball\_image

wxImage Pole\_przyspieszen\_czastki\_GFKFrm::ball\_image [protected]

Image of the ball from which a bitmap is made

#### 4.3.2.3 board

float Pole\_przyspieszen\_czastki\_GFKFrm::board [protected]

Array containing every pixel's data on the drawing panel

#### 4.3.2.4 ContourBitMap

wxBitmap Pole\_przyspieszen\_czastki\_GFKFrm::ContourBitMap [protected]

Main bitmap presenting the colored map with contours

#### 4.3.2.5 draw\_ball

bool Pole\_przyspieszen\_czastki\_GFKFrm::draw\_ball [protected]

Variable defining if the moving ball animation should be present on the map

#### 4.3.2.6 draw\_vectors

bool Pole\_przyspieszen\_czastki\_GFKFrm::draw\_vectors [protected]

Variable defining if the vectors are present on the map

#### 4.3.2.7 help

**HelpDlg** Pole\_przyspieszen\_czastki\_GFKFrm::help [protected]

Dialog box containing vital information about the program and how to use it

#### 4.3.2.8 initial\_screen

**InitialScreenDlg** Pole\_przyspieszen\_czastki\_GFKFrm::initial\_screen [protected]

Initial screen window, displayed at the start up of program

#### 4.3.2.9 max

float Pole\_przyspieszen\_czastki\_GFKFrm::max [protected]

Maximal height value

#### 4.3.2.10 min

float Pole\_przyspieszen\_czastki\_GFKFrm::min [protected]

Minimal height value

#### 4.3.2.11 NoLevels

int Pole\_przyspieszen\_czastki\_GFKFrm::NoLevels [protected]

Number of contours shown on the map

#### 4.3.2.12 points

Point Pole\_przyspieszen\_czastki\_GFKFrm::points[1400] [protected]

Array of points included in acceleration vectors computations

#### 4.3.2.13 scroll

int Pole\_przyspieszen\_czastki\_GFKFrm::scroll [protected]

Variable defining the maximum scroll value of the time scrollbar

#### 4.3.2.14 time

int Pole\_przyspieszen\_czastki\_GFKFrm::time [protected]

Variable modifying the surface

#### 4.3.2.15 VectorBitMap

wxBitmap Pole\_przyspieszen\_czastki\_GFKFrm::VectorBitMap [protected]

Bitmap overlaying the contour map, responsible for drawing the vectors

#### 4.3.2.16 x\_coordinates\_max

int Pole\_przyspieszen\_czastki\_GFKFrm::x\_coordinates\_max [protected]

X Coordinate of the highest point of map

#### 4.3.2.17 y\_coordinates\_max

int Pole\_przyspieszen\_czastki\_GFKFrm::y\_coordinates\_max [protected]

Y Coordinate of the highest point of map

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

- Pole\_przyspieszen\_czastki\_GFKFrm.h
- Pole\_przyspieszen\_czastki\_GFKFrm.cpp

## 4.4 Pole\_przyspieszen\_czastki\_GFKFrmApp Class Reference

Inherits wxApp.

### Public Member Functions

- bool **OnInit** ()
- int **OnExit** ()

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

- Pole\_przyspieszen\_czastki\_GFKApp.h
- Pole\_przyspieszen\_czastki\_GFKApp.cpp

# Chapter 5

## File Documentation

### 5.1 HelpDlg.cpp File Reference

```
#include "HelpDlg.h"
```

#### 5.1.1 Detailed Description

##### Author

Pitala&Marszalek&Matuszewski Created: 2017-06-14 20:46:45

#### 5.1.2 DESCRIPTION

**HelpDlg** (p. ??) class implementation

### 5.2 HelpDlg.h File Reference

```
#include <wx/wx.h> #include  
<wx/dialog.h>  
#include <wx/richtext/richtextctrl.h> #include  
<wx/button.h>
```

#### Classes

- class **HelpDlg**

#### Macros

- #define **HelpDlg\_STYLE** wxCAPTION j wxSYSTEM\_MENU j wxDIALOG\_NO\_PARENT j wxMINIMIZE\_ -  
BOX j wxCLOSE\_BOX

### 5.2.1 Detailed Description

#### Author

Pitala&Marszalek&Matuszewski Created: 2017-06-14 20:46:45

### 5.2.2 DESCRIPTION

**HelpDlg** (p. ??) class declaration

## 5.3 InitialScreenDlg.cpp File Reference

```
#include "InitialScreenDlg.h"
```

### 5.3.1 Detailed Description

#### Author

Pitala&Marszalek&Matuszewski Created: 2017-06-14 20:55:17

### 5.3.2 DESCRIPTION

**InitialScreenDlg** (p. ??) class implementation

## 5.4 InitialScreenDlg.h File Reference

```
#include <wx/wx.h> #include  
<wx/dialog.h> #include  
<wx/panel.h> #include  
<wx/button.h> #include  
<unistd.h>
```

### Classes

- class **InitialScreenDlg**

### Macros

- `#define InitialScreenDlg_STYLE wxCAPTION j wxSYSTEM_MENU j wxDIALOG_NO_PARENT j wxMINIMIZE_BOX j wxCLOSE_BOX`

### 5.4.1 Detailed Description

#### Author

Pitala&Marszalek&Matuszewski Created: 2017-06-14 20:55:17

### 5.4.2 DESCRIPTION

**InitialScreenDlg** (p. ??) class declaration