Użyte biblioteki

`stdio.h` `stdlib.h` `time.h` `gtk/gtk.h` `stdbool.h` `gdk/gdkkeysyms.h`

Definicje i makra:

C_RAD	promien koła reprezentującego gracza
TIMEOUT	opóźnienie animacji w ms / ilość milisekund na jedną klatkę
VECT	długość wektora poruszającego graczami
deltaAlpha	o jaki kat obroci sie gracz po nacisnieciu klawisza
WinWidth	szerokość okna
WinHeight	wysokość okna
BORDER	długość wewnętrzenej ramki, „martwego pola”
AREA_MULTIPLIER	mnożnik macierzy kolizji, zwiększa dokładność
setBLUEcol(x)	ustawia kolor x - obiektu cairo na niebieski
setGREENcol(x)	ustawia kolor x – obiektu cairo na zielony

Funkcje programu.

`void activate(GtkApplication *app, gpointer data);`
Inicjuje okno GTK.

`void drawBackground(char *msg);`
Rysuje planszę i wyświetla komunikat *msg*.

`void drawing();`
Uaktualnia plaszę o ruchy graczy.

`void drawCircle(cairo_t *cr, int posX, int poxY, double radius);`
Rysuje punkt reprezentujący aktualną pozycje gracza.

`void nextRound(int winner);`
Aktualizuje wynik gracza *winner* oraz na nowo inicjalizuje tablice kolizji oraz pozycje graczy.

`void playerInit();`
Losuje pozycję graczy, ich początkowy kąt oraz czas pozostały do kolejnego przerwania śladu.

`void initCrashTab();`
Inicjalizuje macierz kolizji.

`void fillCrashTab(double x, double y);`
Uzupełnia macierz kolizji o (*x,y*) – aktualne współrzędne gracza. (A dokłanie wypełnia kwadrat opisany na okęgu o promieniu C_RAD i środku (*x,y*)).

`bool aboutToCrash(double x, double y);`
Zwraca *true*, jeżeli gracz o aktualnej pozycji (*x,y*) dokonał kolizji. False wpp.

`void playerMove();`
Aktualizuje współrzędne graczy.

`void playerMoveLeft(short who);`
Zmienie kierunek poruszania sie gracza *who* – obraca go w lewo (wywoływane przy naciśnięciu odpowiedniego klawisza)

`void playerMoveRight(short who);`

Zmienie kierunek poruszania sie gracza *who* – obraca go w prawo (wywoływane przy naciścięciu odpowiedniego klawisza)

`int randTimeout();` Zwraca losową ilość jednotek czasu, po którym ślad gracza ulegnie chwilowemu przerwaniu.

`nt randXcoord();` Zwraca losową współrzędną X gracza.

`int randYcoord();` Zwraca losową współrzędną Y gracza.

`int randAlpha();` Zwraca losowy kąt.

`gboolean draw_it(GtkWidget *widget);`

Funkcja odpowiedzialna za animację w grze oraz za stany graczy (zostawia ślad/ nie zostawia) względem jednostek czasu.

```
gboolean drawInit(GtkWidget *widget, GdkEventConfigure *event,
gpointer data);
```

Funkcja inicjująca animację.

```
gboolean drawInitDraw(GtkWidget *widget, cairo_t *cr, gpointer
data);
```

Funkcja rysująca czystą planszę.

```
gboolean on_key_press(GtkWidget *widget, GdkEventKey *event,
gpointer data);
```

Funkcja reagującą na naciścięcie klawisza – rozpoczyna grę lub wywołuję funkcję obracającą graczem (playerMoveRight, playerMoveLeft).

```
gboolean on_key_release(GtkWidget *widget, GdkEventKey *event,
gpointer data);
```

Funkcja reagująca na zwolnienie klawisza.