

Dokumentacja Projektu „Problem liszaja”

Wiktor Mazur, 173669

2EF-DI

Cel projektu

Celem projektu jest przeprowadzenie symulacji zarażania komórek, napisanej w języku programowania C++.

Technologie

Język programowania:	C++ (Standard ISO C++ 14)
Biblioteka graficzna:	SFML v2.6.1
Środowisko programistyczne:	Visual Studio 2022 v143
Wersja Zestawu Windows SDK:	10.0

Klasa Board

Opis

Klasa Board służy do wyświetlania tablicy na której symulacja oraz logika symulacji.

Pola

unsigned int size; - zmienna obsługująca wielkość planszy

float cellSize; - zmienna obsługująca wielkość komórki

float boardSize = 550.0f; - zmienna obsługująca wielkość planszy. Jest ona stała

float offsetX; - zmienna obsługująca wielkość miejsca po prawej jak i po lewej stronie programu

float offsetY; - zmienna obsługująca wielkość miejsca u góry jak i u dołu programu

Duration* duration; - Obiekt klasy Duration

sf::Font font; - Obiekt klasy Font. Obsługuje czcionkę

vector<vector<HealthStatus>> healthStatuses; - Wektor statusu komórki

vector<vector<sf::Color>> colors; - Wektor koloru komórki

sf::Text title; - Obiekt klasy tekst biblioteki SFML.

sf::Text roundText; - Obiekt klasy tekst biblioteki SFML.

sf::Text deltaTimeText; - Obiekt klasy tekst biblioteki SFML.

sf::Text allTimeText; - Obiekt klasy tekst biblioteki SFML.

sf::Text allCellText; - Obiekt klasy tekst biblioteki SFML.

sf::Text infectedCellText; - Obiekt klasy tekst biblioteki SFML.

sf::Text immuneCellText; - Obiekt klasy tekst biblioteki SFML.

sf::Text healthCellText; - Obiekt klasy tekst biblioteki SFML.

vector<tuple<int, int, int>> toStore; - Wektor zawierający informacje o komórkach zarażonych i chorych.

Metody

`int countCells(HealthStatus status, int boardSize);` - Metoda zliczająca komórki

`void updateSize(int newSize);` - Metoda która aktualizuje status komórek

`void drawBoard(sf::RenderWindow& window);` - Metoda która rysuje plansze

`void calculateboardSize(sf::RenderWindow& window);` - Metoda która oblicza wielkość planszy

`void handleClick(int currentround, sf::RenderWindow& window);` - Metoda która pozwala na zarażenie komórek poprzez klikanie na plansze

`void update(int boardSize, int currentRound, float deltaTime, float allTime, sf::RenderWindow& window, int infectionPercent, int infectedToImmune, int immuneCooldown);` - Metoda która aktualizuje stan komórki

`void findRowAndCol(unsigned int row, unsigned int col, int currentround, float infectionPercent);` - Metoda która zaraża komórki na około komórki chorej.

`void addTotoStore(int newRow, int newCol, int currentround);` - Metoda która dodaje komórki zarażone do listy.

`bool isInToStore(int currentround, int newRow, int newCol);` - Metoda która sprawdza czy komórka jest zarażona

`void drawtoStore(vector<tuple<int, int, int>>& toStore);` - Metoda która wypisuje dodane komórki do listy

`void spreadInfection(vector<tuple<int, int, int>>& toStore, int currentround, int infectedToImmune, int immuneCooldown);` - Metoda która obsługuje rozchodzenie się infekcji

`void removeHealthCells(vector<tuple<int, int, int>>& toStore, int currentround, int infectedToImmune, int immuneCooldown);` - Metoda która usuwa zdrowe komórki z Listy zarażonych/odpornych

Klasa Menu

Opis

Klasa Menu obsługuje Menu programu w którym wywoływane są podstawowe parametry programu.

Pola

int menuSize=50; - zmienna obsługująca wielkość planszy

int menuMaxround = 1000; - zmienna obsługująca ilość rund

float menuHoldprocess = 0.5; - zmienna obsługująca czas wstrzymania wątku

int menuInfectionPercent = 50; - zmienna obsługująca procent szans zarażenia

int menuInfectedToImmune = 1; - zmienna obsługująca czas zarażenia komórki

int menuImmuneCooldown = 1; - zmienna obsługująca czas odporności komórki

sf::Font font; - Obiekt klasy tekst biblioteki SFML.

sf::Text title; - Obiekt klasy tekst biblioteki SFML.

sf::Text sizeText; - Obiekt klasy tekst biblioteki SFML.

sf::Text buttonText; - Obiekt klasy tekst biblioteki SFML.

sf::Text holdprocessText; - Obiekt klasy tekst biblioteki SFML.

sf::Text infectionPercentText; - Obiekt klasy tekst biblioteki SFML.

sf::Text infectedToImmuneText; - Obiekt klasy tekst biblioteki SFML.

sf::Text immuneCooldownText; - Obiekt klasy tekst biblioteki SFML.

sf::Text exit; - Obiekt klasy tekst biblioteki SFML.

Board* board; - Obiekt klasy Board

Duration* duration; - Obiekt klasy Duration

Metody

`void drawMenu(sf::RenderWindow& window);` - metoda która rysuje menu

`void drawButton(sf::RenderWindow& window, const std::string& text, const sf::Vector2f& position, const std::function<void()>& onClick, const sf::Color& mainColor, const sf::Color& hoverColor);` - metoda która rysuje przycisk

`void increase(int howMuch, const std::string& variable);` - metoda która zwiększa ilość danej zmiennej

`void decrease(int howMuch, const std::string& variable);` - metoda która zmniejsza ilość danej zmiennej

Klasa Duration

Opis

Klasa ta obsługuje czas występujący w grze.

Pola

int maxround; - zmienna obsługująca ilość rund

unsigned int holdprocess; - zmienna obsługująca czas wstrzymania wątku

int infectionPercent; - zmienna obsługująca procent szans zarażenia

unsigned int infectedToImmune; - zmienna obsługująca czas zarażenia komórki

unsigned int immuneCooldown; - zmienna obsługująca czas odporności komórki

Metody

Duration(int maxround, unsigned int holdprocess, int infectionPercent, unsigned int infectedToImmune, unsigned int immuneCooldown); - Konstruktor ustawiający zmienne w danej klasie

Klasa Game

Opis

Klasa Game obsługuje uruchamianie projektu oraz podstawowe parametry z którymi gra jest uruchamiana.

Pola

Duration* duration; - Objekt klasy Duration

Board* board; - Objekt klasy Board

Menu* menu; - Objekt klasy Menu

sf::Clock clock; - Objekt klasy Clock

float deltaTime; - zmienna obsługująca czas

float allTime; - zmienna obsługująca całkowity czas trwania programu

bool isFirst = true; - Sprawdza czy jest pierwsza runda

sf::Font font; - Objekt klasy tekst biblioteki SFML.

sf::Text exitHealth; - Objekt klasy tekst biblioteki SFML.

sf::Text exitInfected; - Objekt klasy tekst biblioteki SFML.

Metody

int run(); - metoda która uruchamia program

void drawEndAllHealth(sf::RenderWindow& window);

void drawEndAllInfected(sf::RenderWindow& window);

void delay(int time, sf::RenderWindow& window, bool& menuOpen); - metoda która wstrzymuje działanie programu

Zdjęcia Plików Nagłówkowych

```
class Board {
public:
    Board(int size);
    void consoleStart();
    void updateSize(int newSize);
    void drawBoard(sf::RenderWindow& window);
    void calculateBoardSize(sf::RenderWindow& window);
    void handleClick(int currentround, sf::RenderWindow& window);
    void update(int boardSize, int currentRound, float deltaTime, float allTime, sf::RenderWindow& window, int infectionPercent, int infectedToImmune, int immuneCooldown);
    unsigned int size;
    int countCells(HealthStatus status, int boardSize);
private:
    float cellSize;
    float boardSize = 550.0f;
    float offsetX;
    float offsetY;
    sf::Font font;
    Duration* duration;

    vector<vector<HealthStatus>> healthStatuses;
    vector<vector<sf::Color>> colors;

    sf::Text title;
    sf::Text roundText;
    sf::Text deltaTimeText;
    sf::Text allTimeText;
    sf::Text allCellText;
    sf::Text infectedCellText;
    sf::Text immuneCellText;
    sf::Text healthCellText;

    vector<tuple<int, int, int>> toStore;
    void findRowAndCol(unsigned int row, unsigned int col, int currentround, float infectionPercent);
    void addInfoToStore(int newRow, int newCol, int currentround);
    bool isInfoToStore(int currentround, int newRow, int newCol);
    void drawInfoToStore(vector<tuple<int, int, int>>& toStore);
    void spreadInfection(vector<tuple<int, int, int>>& toStore, int currentround, int infectedToImmune, int immuneCooldown);
    void removeHealthCells(vector<tuple<int, int, int>>& toStore, int currentround, int infectedToImmune, int immuneCooldown);
};
```

Rysunek pliku nagłówkowego klasy Board.

```
class Menu{
public:
    void drawMenu(sf::RenderWindow& window);
    void drawButton(sf::RenderWindow& window, const std::string& text, const sf::Vector2f& position, const std::function<void()>& onClick, const sf::Color& color);
    void increase(int howMuch, const std::string& variable);
    void decrease(int howMuch, const std::string& variable);
    void increaseFloat(float howMuch, const std::string& variable);
    void decreaseFloat(float howMuch, const std::string& variable);
    int menuSize=50;
    int menuMaxround = 1000;
    float menuHoldprocess = 0.5;
    int menuInfectionPercent = 50;
    int menuInfectedToImmune = 1;
    int menuImmuneCooldown = 1;
private:
    sf::Font font;
    sf::Text title;
    sf::Text sizeText;
    sf::Text buttonText;
    sf::Text holdprocessText;
    sf::Text infectionPercentText;
    sf::Text infectedToImmuneText;
    sf::Text immuneCooldownText;
    sf::Text exit;
    Board* board;
    Duration* duration;
};
```

Rysunek pliku nagłówkowego klasy Menu.

```
class Duration {
public:
    Duration(int maxround, unsigned int holdprocess, int infectionPercent, unsigned int infectedToImmune, unsigned int immuneCooldown);
    void consoleStart();
    int maxround;
    unsigned int holdprocess;
    int infectionPercent;
    unsigned int infectedToImmune;
    unsigned int immuneCooldown;
};
```

Rysunek pliku nagłówkowego klasy Duration.

```

class Game {
public:
    int run();
    void drawEndAllHealth(sf::RenderWindow& window);
    void drawEndAllInfected(sf::RenderWindow& window);
private:
    Duration* duration;
    Board* board;
    Menu* menu;
    sf::Clock clock;
    float deltaTime;
    float allTime;
    void delay(int time, sf::RenderWindow& window, bool& menuOpen);
    bool isFirst = true;
    sf::Font font;
    sf::Text exitHealth;
    sf::Text exitInfected;
};

```

Rysunek pliku nagłówkowego klasy Game.

```

#include "Game.h"

int main(){
    Game game;
    game.run();
    return 0;
}

```

Rysunek pliku uruchamiającego program.