

Dokumentacja projektu na przedmiot:

Tworzenie gier na urządzenia mobilne

Autor: Michał Musioł

I. Dokumentacja użytkownika

1. Opis założeń gry

Gra ma umożliwiać grę gomoku zarówno dla dwóch graczy jak i dla jednego gracza z sztuczną inteligencją. W założeniu plansza ma generować się poprzez odpowiedni algorytm a nie poprzez wcześniejsze ustawienie przycisków. Program ma umożliwiać wybór rozmiaru planszy a także możliwość zapisu historii rozgrywek wraz z późniejszym ich odtworzeniem.

2. Cel rozgrywki

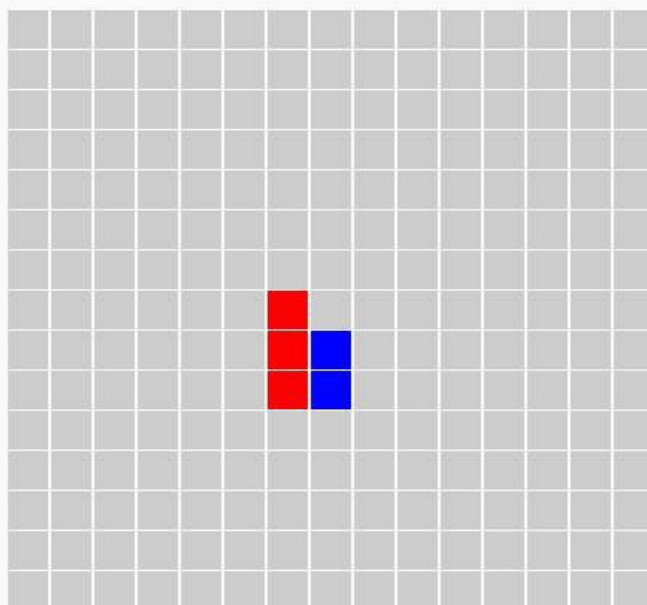
Celem rozgrywki jest pokonanie przeciwnika poprzez ustawienie 5 znaków obok siebie(poziomo, pionowo i na skos).

3. Sterowanie

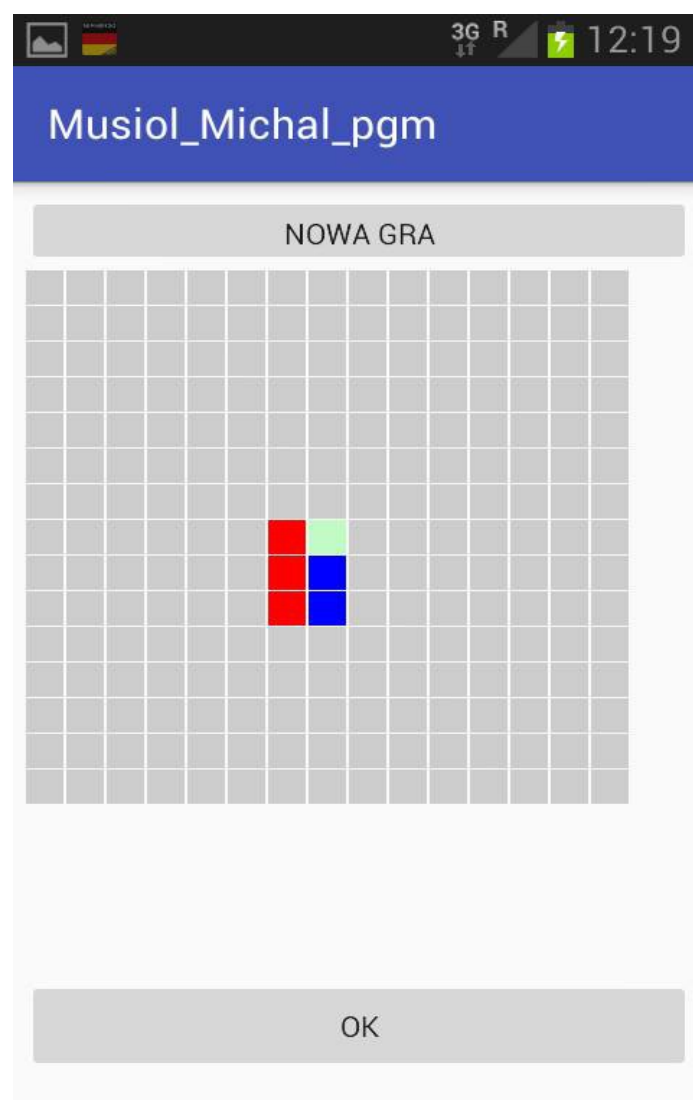
Sterowanie grze możliwe jest wyłącznie dzięki jednemu palcowi, poprzez klikanie w przyciski. Podczas gry, gracz najpierw zaznacza pole, na które chce postawić znak a następnie potwierdza to poprzez naciśnięcie przycisku „ok”.

Musiol_Michal_pgm

NOWA GRA



OK



Przycisk „ok” jest niedostępny do momentu zaznaczenia odpowiedniego pola.



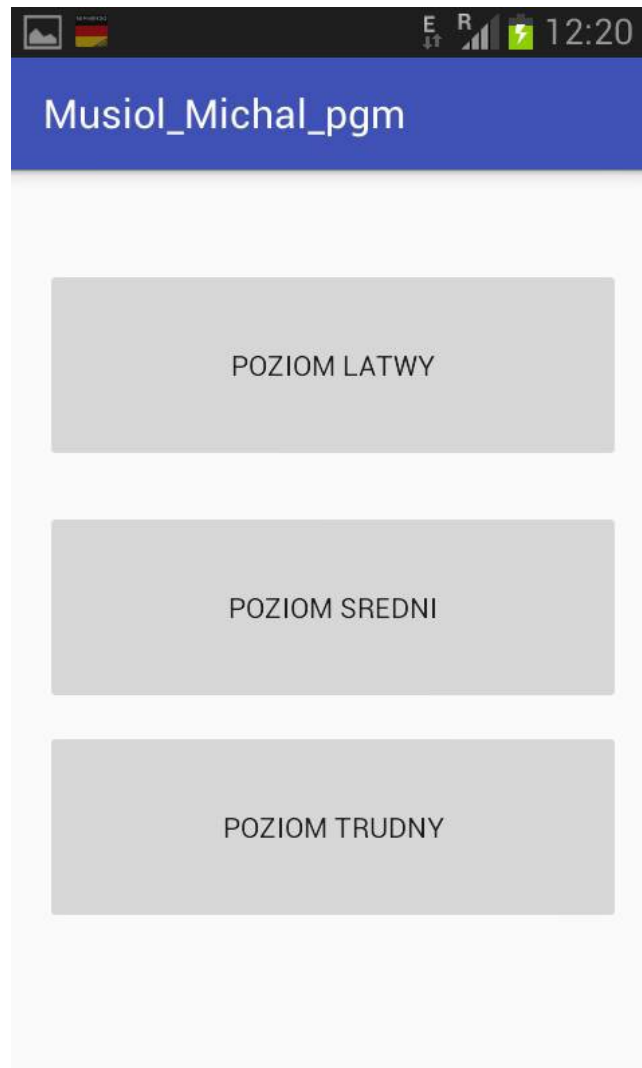
Musiol_Michal_pgm

NOWA GRA DWUOSOBOWA

NOWA GRA Z KOMPUTEREM

HISTORIA GIER

OPCJE



4. Algorytm AI.

Po wyborze poziomu do okna gry przekazywany jest parametr z wartoscia int określający poziom gry. Na początku gry tworzony jest obiekt obsługujący zachowanie ai.

```

public void makeMove() {
    SecureRandom r = new SecureRandom();
    int a = 0, b = 0;
    if(lvl == 1) {
        do {
            a = r.nextInt(15);
            b = r.nextInt(15);

        } while (gameBoard[a][b] == aiSign || gameBoard[a][b] == playerSign);

        setXY(a, b);
    }

    if(lvl == 2)
    {
        searchBoardAi();
        findMax();
    }
}

```

```

public void findSpec(String direction, int row, int column, String s) {

    int counter = 0;
    for(int a = 0; a < 4; a++) {
        if (direction == "vertical") {
            if (gameBoard[row + a][column] == s)
                counter++;
        }
        public void searchBoardAi()
        {
            int counter = 0;
            if
            for(int a = 0; a < sizeo; a++)
            {
                for(int b = 0; b < sizeo; b++)
                {
                    if(b < sizeo-4)
                    {
                        findSpec("horizontal",a,b,aiSign);
                    }

                    if(a < sizeo - 4)
                    {
                        findSpec("vertical",a,b,aiSign);
                    }

                    if(b < sizeo -4 && a < sizeo -4)
                    {
                        findSpec("crossedr",a,b,aiSign);
                    }

                    if(a < sizeo - 4 && b > 3)
                    {
                        findSpec("crossedup",a,b,aiSign);
                    }
                }
            }
        }
    }
}

```

Dotychczas został zaimplementowany algorytm dla dwóch poziomów. W poziomie łatwy algorytm dodaje ruchy dla losowych pól, natomiast dla dwóch kolejnych algorytm dodaje znak tam gdzie jest najwięcej pól w rzędzie.

II. Dokumentacja techniczna

1. Funkcjonalności dostępne w grze.

- 1.1 Wybór poziomu trudności – obsługiwany poprzez 3 przyciski z informacją o poziomie (łatwy, średni, trudny).
- 1.2 Obsługa gry – Gracz ma możliwość sterowania poprzez naciskanie na odpowiednie przyciski. Mechanika gry zapobiega przypadkowym ruchom, poprzez podświetlenie wybranego pola a następnie potwierdzenie tego przyciskiem „ok”.
- 1.3 Nowa gra – gracz ma możliwość zresetowania planszy poprzez naciśnięcie przycisku „nowa gra”.

2. Funkcjonalności które nie zostały zaimplementowane

W finalnej wersji gry zabrakło możliwości zapisywania historii gry oraz zaawansowanego algorytmu AI.