Komunikacja Człowiek Komputer 2021
Dokumentacja projektu 2
Adam Dębek

1.Opis Projektu
Założeniem projektu było stworzenie prostej gry polegającej na przejściu losowo generowanego labiryntu za pomocą przechylania telefonu (czujniki). Początkiem trasy jest górny lewy róg, a końcem lewy dolny. Przejście każdego z labiryntów generuje kolejny poziom utrudniony o bardziej zawiły labirynt.

2. Opis funkcjonalności
Głównymi funkcjonalnościami są ruch bohatera (kwadratu) za pomocą sensorów w systemie Android - akcelerometru i magnetometru, menu główne, gdzie można rozpocząć grę oraz ustawić głośność. Do wygenerowania losowego labiryntu został użyty specjalny algorytm - recursive back tracker.

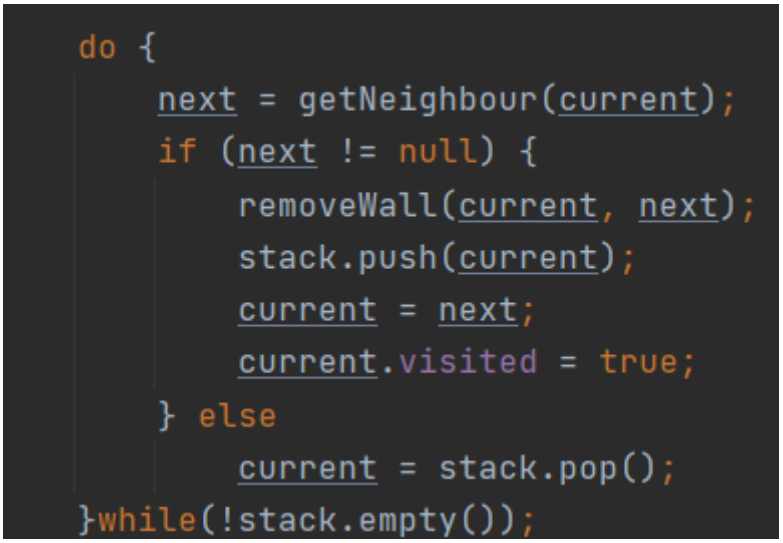
3. Szczególnie interesujące zagadnienia projektowe
Interesującą częścią kodu jest algorytm recursive back tracker. Cały labirynt jest utworzony na podstawie komórek. Na początku trzeba wybrać pozycję startową w moim przypadku lewy górny róg (komórka[0][0]). Następnie dodaniu do listy wszystkich możliwych dróg (sąsiadów) , wybór losowo jednego z nich. Ta wybrano losowo komórka staje się obecną komórką oraz jest wrzucana na stos. Algorytm ten jest powtarzany w kółko, aż do momentu braku sąsiadów, wtedy za pomocą stosu wracamy po komórkach, aż do napotkania komórki z sąsiadem. Na koniec algorytm powinien wrócić do pozycji początkowej.

```
private Cell getNeighbour(Cell cell){
    ArrayList<Cell> neighbours = new ArrayList<>();

    //left
    if(cell.col > 0)
        if(!cells[cell.col - 1][cell.row].visited)
            neighbours.add(cells[cell.col-1][cell.row]);
    //right
    if(cell.col < COLS - 1)
        if(!cells[cell.col + 1][cell.row].visited)
            neighbours.add(cells[cell.col+1][cell.row]);
    //top
    if(cell.row > 0)
        if(!cells[cell.col][cell.row-1].visited)
            neighbours.add(cells[cell.col][cell.row-1]);
    //bottom
    if(cell.row < ROWS - 1)
        if(!cells[cell.col][cell.row+1].visited)
            neighbours.add(cells[cell.col][cell.row+1]);

    if(neighbours.size() > 0) {
        int index = random.nextInt(neighbours.size());
        return neighbours.get(index);
    }
    return null;
```

Funkcja sprawdzająca sąsiadów i zwracająca losowego jeśli istnieje.



```
do {
    next = getNeighbour(current);
    if (next != null) {
        removeWall(current, next);
        stack.push(current);
        current = next;
        current.visited = true;
    } else
        current = stack.pop();
}while(!stack.empty());
```

Główna część algorytmu, przechodząca po komórkach.

4.Instrukcja instalacji
Należy podłączyć telefon przez wejście USB do komputera i skompilować projekt w Android Studio. Trzeba pamiętać aby ustawić odpowiednie ustawienia developerskie w systemie Android aby telefon był widzialny przez Android Studio.

5.Instrukcja konfiguracji - brak.

6.Instrukcja użytkownika
Aby poruszać się głównym obiektem należy przechylać telefon. Odpowiednie przechylenie w lewo - ruch w lewo, analogicznie dla pozostałych kierunków.

7.Wnioski
Projekt działa tak jak to sobie zaplanowałem. W trakcie pracy nad nim nauczyłem się paru ciekawych i przydatnych rzeczy.

8.Samoocena
Nie było to łatwe zadanie, trochę czasochłonne, natomiast warte pracy potrzebnej na jego wykonanie. Projekt w moim przekonaniu działa tak jak to sobie wymyśliłem na początku.