

Projekt JPWP

Prezentacja 3

Kamil Myćka

Link do repozytorium: [serious-game](#)

CELE

Temat projektu:

Interaktywna gra komputerowa w stylu JRPG

Cel projektu:

Stworzenie gry komputerowej, w której gracz poruszać będzie się po labiryncie i będzie pokonywał przeciwników, poprzez rozwiązywanie prostych równań matematycznych w określonym czasie.

Docelowi użytkownicy gry:

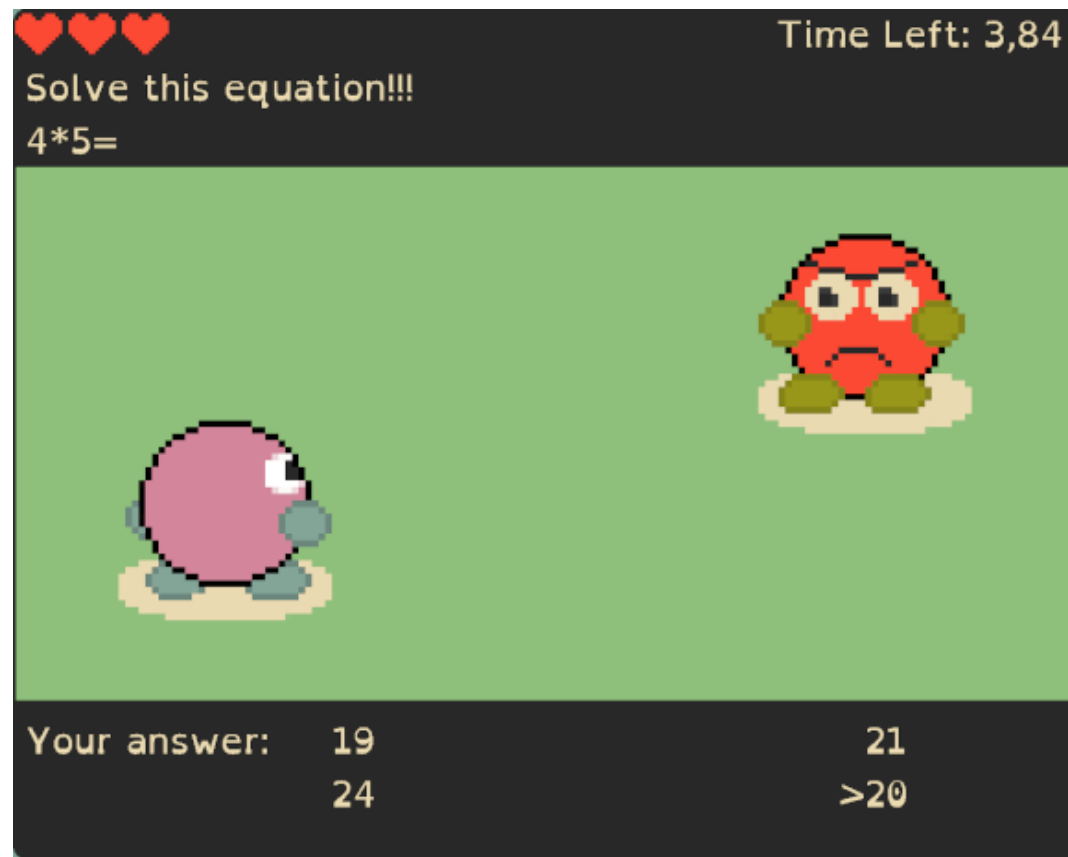
Dzieci w wieku wczesnoszkolnym, rozpoczynający swoją naukę matematyki, oraz ludzie pragnący polepszyć swoje zdolności matematyczne.

Projekt interfejsu graficznego

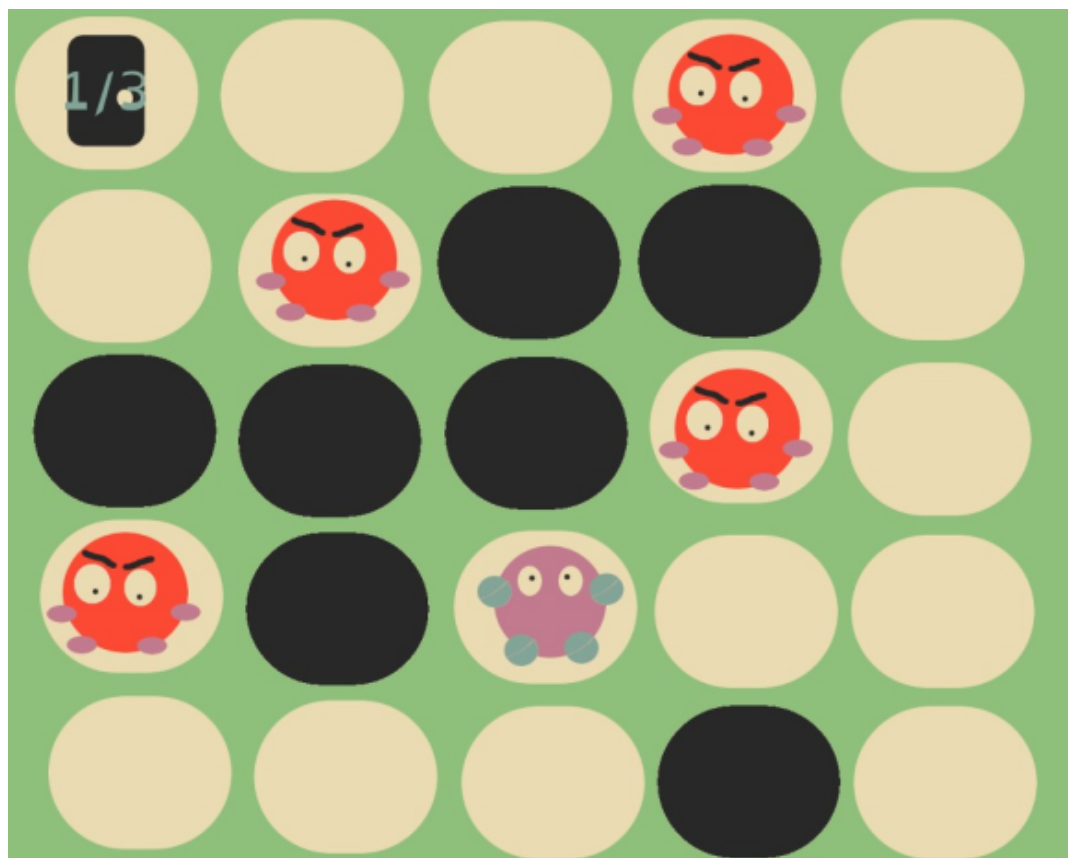
Wczesny zarys interfejsu



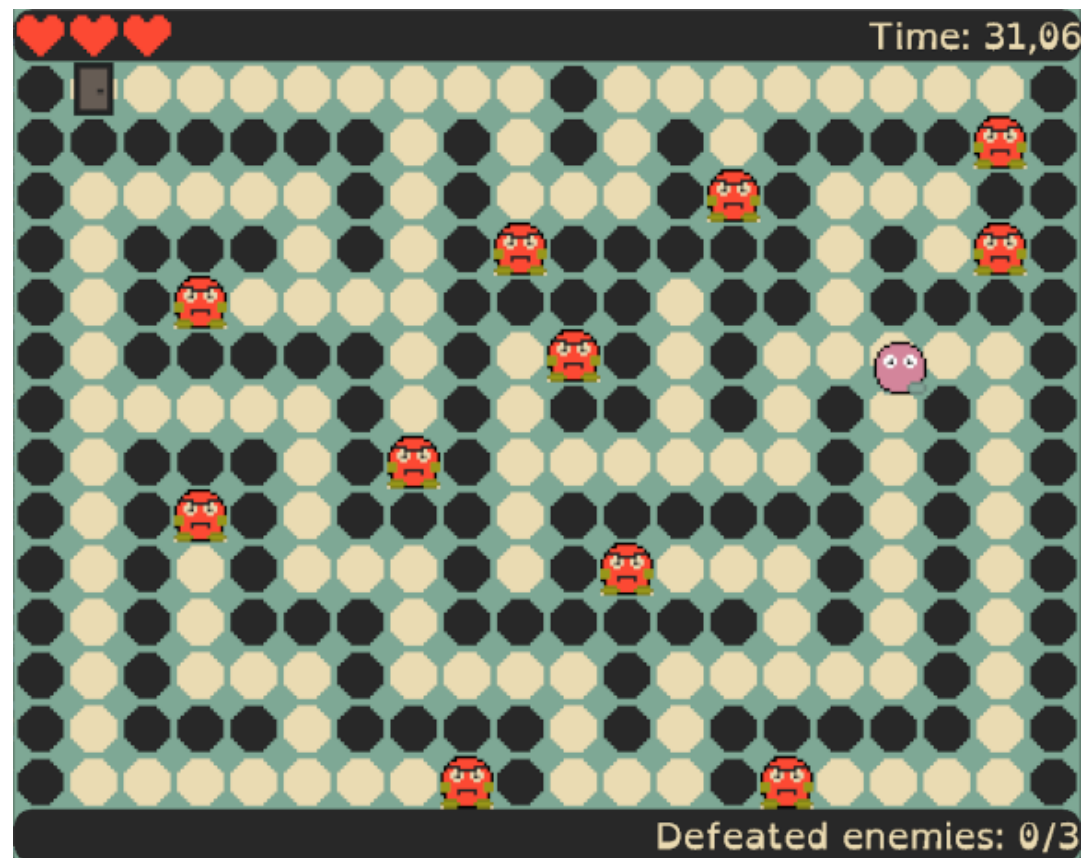
Wersja końcowa



Wczesny zarys interfejsu



Wersja końcowa



Zasoby gry

Graficzne

| Przód 1 | Przód 2 | Tył 1 | Tył 2 | Lewo 1 | Lewo 2 | Prawo 1 | Prawo 2 |
|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |

| Przeciwnik | Pełne serce | Puste serce | Drzwi | Droga | Kamienna ściana |
|---|---|--|---|---|---|
|  |  |  |  |  |  |

| Gracz w walce | Przeciwnik w walce |
|--|--|
|  |  |

Wszystkie grafiki zostały wykonane własnoręcznie wykorzystując GNU Image Manipulation Program.

Czcionka

OpenDyslexic

| OpenDyslexic-Regular.otf | | | | | | | | | | | | | | |
|--------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| A 0065 | B 0066 | C 0067 | D 0068 | E 0069 | F 0070 | G 0071 | H 0072 | I 0073 | J 0074 | K 0075 | L 0076 | M 0077 | N 0078 | |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | |
| O 0079 | P 0080 | Q 0081 | R 0082 | S 0083 | T 0084 | U 0085 | V 0086 | W 0087 | X 0088 | Y 0089 | Z 0090 | | | |
| O | P | Q | R | S | T | U | V | W | X | Y | Z | | | |
| a 0097 | b 0098 | c 0099 | d 0100 | e 0101 | f 0102 | g 0103 | h 0104 | i 0105 | j 0106 | k 0107 | l 0108 | m 0109 | n 0110 | o 0111 |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| p 0112 | q 0113 | r 0114 | s 0115 | t 0116 | u 0117 | v 0118 | w 0119 | x 0120 | y 0121 | z 0122 | | | | |
| p | q | r | s | t | u | v | w | x | y | z | | | | |

Składowanie danych

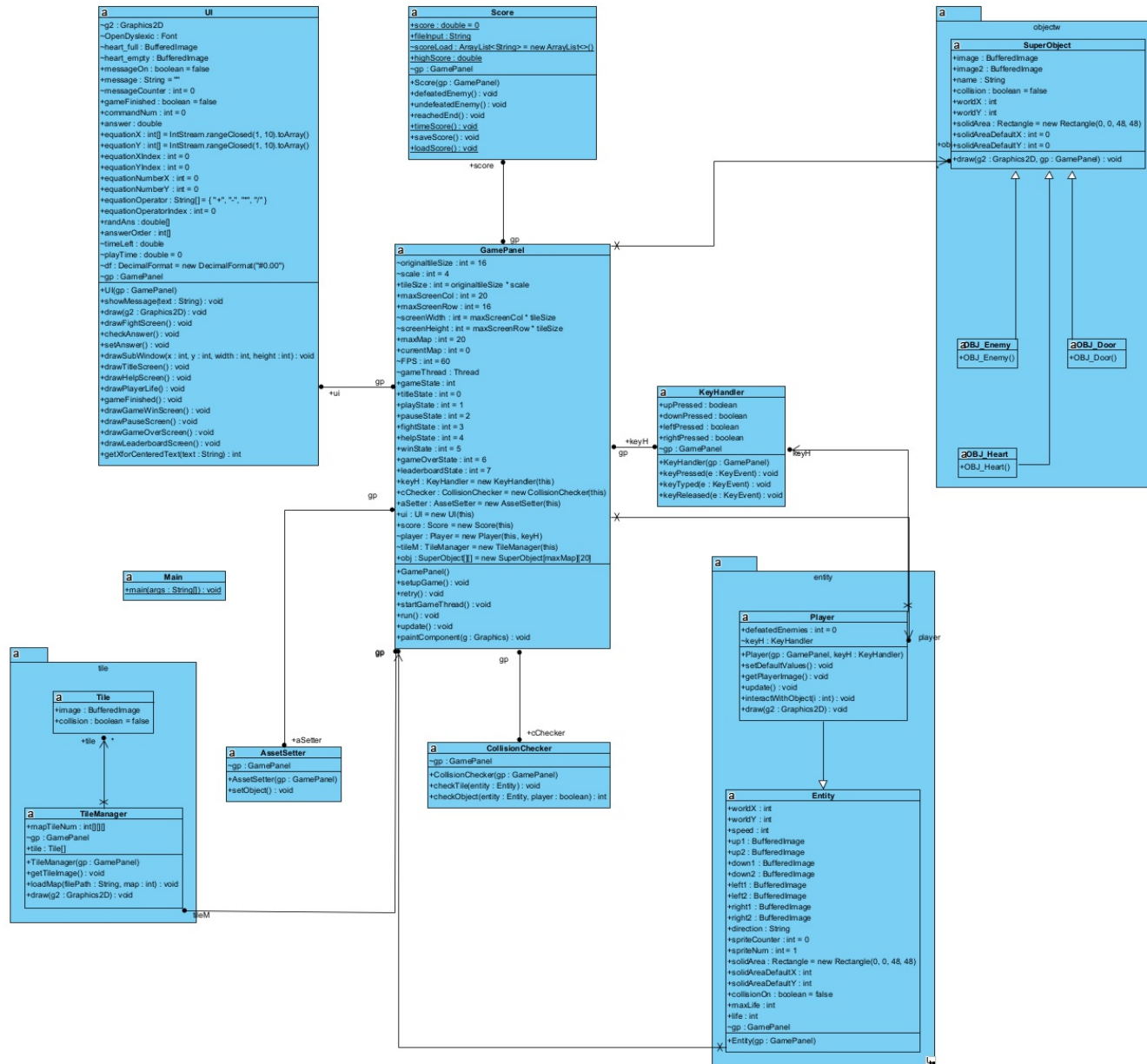
Wyniki gracza przechowywane są w pliku /score.txt, co pozwala na łatwe odczytanie ich w każdej chwili zarówno przez rodzica który chce sprawdzić postęp swojego dziecka, jak i przez samo dziecko.

```
6 1178.4959999999996
5 1078.3839999999961
4 1913.4240000000086
3 2266.5440000000011
2 1811.05600000000628
1 1914.43200000000534
19 3648.63999999992477
1 1796.54400000000417
2 2855.3439999999792
3 1772.0319999999788
4 1695.29600000000621
```

Stan realizacji projektu

1. Zaprojektowano interfejs graficzny gry
2. Zaprojektowano menu gry
3. Zrealizowano wszystkie grafiki, lecz mogą zostać one jeszcze poprawione
4. Zrealizowano wszystkie zaplanowane funkcjonalności

Diagramy klas



Funkcja generująca równanie matematyczne

```
public void randomEquation() {
    // GENERATING RANDOM NUMBERS FOR THE EQUATION
    equationYIndex = (int) (Math.random() * equationX.length);
    equationXIndex = (int) (Math.random() * equationY.length);
    equationOperatorIndex = (int) (Math.random() * equationOperator.length);
    setAnswer();
    randAns = new double[4];

    // FILL RANDANS WITH RANDOM INTEGER NUMBERS
    for (int i = 0; i < randAns.length; i++) {
        randAns[i] = answer + (int) (Math.random() * (5 + 5)) - 5;
        if (randAns[i] == answer) {
            randAns[i] = answer + (int) (Math.random() * (5 + 5)) - 5;
        }
        for (int j = 0; j < i; j++) {
            if (randAns[i] == randAns[j]) {
                randAns[i] = answer + (int) (Math.random() * (5 + 5)) - 5;
            }
        }
    }

    // SHUFFLE ARRAY
    randAns[0] = answer;
    for (int i = 0; i < randAns.length; i++) {
        int rand = (int) (Math.random() * randAns.length);
        double temp = randAns[i];
        randAns[i] = randAns[rand];
        randAns[rand] = temp;
    }

    // CHECK IF ANSWERS REPEAT AND SHUFFLE AGAIN
    for (int i = 0; i < randAns.length; i++) {
        for (int j = 0; j < i; j++) {
            if (randAns[i] == randAns[j]) {
                randAns[j] = answer + (int) (Math.random() * (5 + 5)) - 5;
            }
        }
    }
}

// SOLVIN THE EQUATION AND SETTING THE ANSWER
public void setAnswer() {
    switch (equationOperator[equationOperatorIndex]) {
        case "+":
            answer = equationX[equationXIndex] + equationY[equationYIndex];
            break;
        case "-":
            answer = equationX[equationXIndex] - equationY[equationYIndex];
            break;
        case "*":
            answer = equationX[equationXIndex] * equationY[equationYIndex];
            break;
        case "/":
            if (((double) equationX[equationXIndex] / (double) equationY[equationYIndex] % 1 == 0) {
                answer = equationX[equationXIndex] / equationY[equationYIndex];
            } else {
                equationOperatorIndex = 2;
                answer = (double) equationX[equationXIndex] * (double) equationY[equationYIndex];
            }
            break;
    }
}
```

Efekt funkcji

