

Projekt: Gra Tetris

Numer grupy 2ID13A	Data wykonania ćwiczenia:	Ocena
	Data oddania sprawozdania:	
Zespół realizujący: <div><div>1.</div><div>Viktor Tykhonov</div></div> <div><div>2.</div><div>Grzesiek Kalarus</div></div>		

Opis projektu i gry

Tematem naszego projektu był „Tetris”. Tetris to jedna z najsłynniejszych gier wideo stworzona przez Alekseja Pażytnowa i dwóch jego współpracowników Dimitrija Pawłowskiego i Wadima Gierasimowa. Tytuł ten pojawił się na rynku w Związku Radzieckim, w czerwcu 1984 roku, a jego powstanie zawdzięczamy pracom zespołu Pażytnowa na komputerze Elektronika 60 prowadzonym dla Akademii Nauk ZSRR w Moskwie.

Tetris był pierwszą grą ze Związku Radzieckiego, która zawitała do Stanów Zjednoczonych i mimo że pojawiła się ona na kilku różnych platformach do grania oraz na automatach do gier, to dopiero wersja tego tytułu, która została wydana w 1989 roku na konsoli Game Boy przyczyniła się do ogromnego sukcesu tej gry. Od tamtego czasu Tetris, w takiej czy innej formie, pojawił się na praktycznie na każdej współczesnej platformie do grania i już na stałe wszedł do historii komputerowej rozrywki. Jego najnowszą, mainstreamową odsłoną jest zapowiedziana w marcu 2014 roku wersja gry na konsole Xbox One i PlayStation 4.

Do stworzenia gry używamy prostej biblioteki SFML oraz środowiska Visual Studio 2019. Za pomocą tej biblioteki całkowicie rysujemy część graficzną, a także dodajemy dźwięki do gry. Czynnikiem, który zaważył nad wyborem tej właśnie biblioteki było to że ta biblioteka jest jedna z najprostszych w zastosowaniu, przy sporych możliwościach. Poza wspomnianą przed chwilą biblioteką nie użyliśmy żadnych innych skomplikowanych rozszerzeń. Do poprawnego działania naszych funkcji wystarczyły podstawowe biblioteki języka C++.

Opis gry

Po uruchomieniu programu wyświetlone zostanie okienko gry w stałym rozmiarze 800x1000 pixeli. Następnie, musisz nacisnąć przycisk **Enter**, aby rozpocząć grę. Aby sterować tetraminami (figury, które spadają) musisz naciskać strzałki (góra, dół, prawo, lewo) na klawiaturze. Na koniec gry otrzymujesz swój wynik (Punkty za zniszczenie linijek tetraminów). Każdy zniszczony rząd jest wart 100 punktów.

Powodzenia w grze!!!

Krótki opis elementów kodu

Bardziej szczegółowa dokumentacja kodu przygotowana w programie Doxygen.

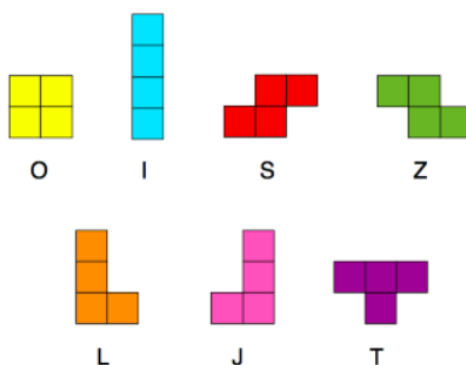
Tutaj opiszę tylko krótki opis kodu.

1. main.c

- **int main()**- główna funkcja odpowiadająca za cały program. Uruchamia bibliotekę „**SFML**” i włącza jej funkcje.

2. Reprezentacja naszych figur

Z Wikipedii możesz dowiedzieć się, że nasze figurki nazywają się **tetramino**, ponieważ każda taka figura składa się z 4 kwadratów. W sumie Tetris używa 7 takich figurek tetromino:



Dlatego całkiem logiczne jest opisanie ich jako całkowitej **dwuwymiarowej tablicy** z 7 rzędami (w zależności od liczby figur) i 4 kolumnami, które określają kształt każdej konkretnej figury. Powstaje jeszcze jedno pytanie: „Jak ustawimy kształt tetramino?”.

Odpowiedź znajdziesz poniżej:

(3, 5, 4, 6)	(3, 4, 5, 7)	(3, 5, 7, 6)																								
<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7
0	1																									
2	3																									
4	5																									
6	7																									
0	1																									
2	3																									
4	5																									
6	7																									
0	1																									
2	3																									
4	5																									
6	7																									

Warto zauważyć, że kolejność wypełnianych kwadratów nie ma dla nas znaczenia. Ważny jest tylko sam fakt: czy kwadrat jest zacieniony, czy nie. Dlatego pierwsza figura ze zbiorem „punktów”(3, 5, 4, 6) może być powiązana zarówno ze zbiorem jak(5, 3, 4, 6) i zbiorem

(3, 5, 6, 4). Pole gry może być również reprezentowane jako tablica liczb całkowitych $\text{Field}[M][N]$, gdzie M jest wysokością pola, a N jego szerokością. Dodajmy te elementy do naszego kodu:

```
11  const int M = 20; // wysokość zabawowego pola
12  const int N = 10; //szerokość zabawowego pola
13  int field[M][N] = { 0 }; // zabawowe pole
14  int points = 0;
15  string disScore;
16
17
18  // Tablica figur tetrimino
19  int figures[7][4] =
20  {
21      1,3,5,7, // I
22      2,4,5,7, // Z
23      3,5,4,6, // S
24      3,5,4,7, // T
25      2,3,5,7, // L
26      3,5,7,6, // J
27      2,3,4,5, // O
28  };
```

3. Wiązanie Tetrimino ze współrzędnymi pola gry

Aby powiązać tetromino ze współrzędnymi pola gry, połączyliśmy współrzędne lokalne, określone przez figury tetrimino, z globalnymi współrzędnymi pola gry.

4. Rotacja Tetrimino

Do realizacji tej funkcjonalności potrzebna jest znajomość algebry liniowej, a mianowicie, że obrót **duszka** wokół danego punktu o współrzędnych (x_0 ; y_0) jest opisany równaniami o następującej postaci:

$$X = x_0 + (x - x_0) * \cos(a) - (y - y_0) * \sin(a);$$

$$Y = y_0 + (y - y_0) * \cos(a) - (x - x_0) * \sin(a);$$

Gdzie (X ; Y) są stare (pierwotne) współrzędne punktu, (X ; Y) są nowymi współrzędnymi (po obrocie) i a jest kątem obrotu. Ponieważ wszystkie zakręty, które mamy, to wyłącznie 90° , a ze szkolnego kursu algebry wiemy, że:

$$\sin(90^\circ) = 1;$$

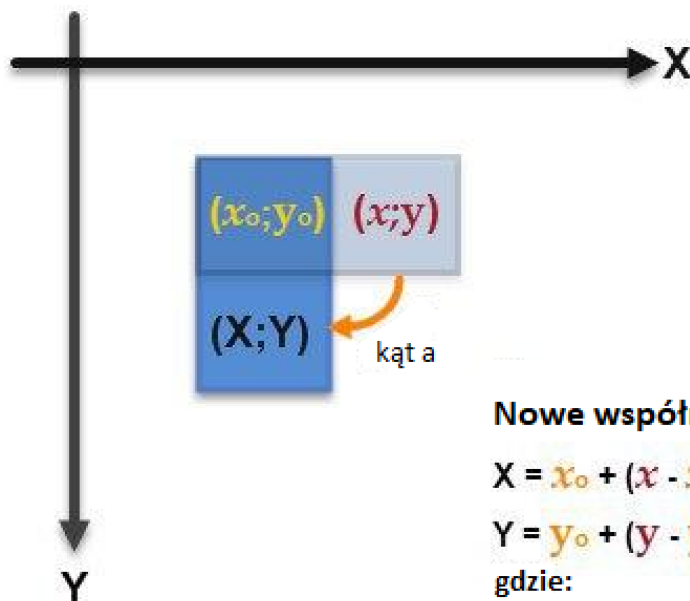
$$\cos(90^\circ) = 0;$$

następnie, zastępując odpowiednie wartości sinusa i cosinusa, oryginalne równania upraszcza się do następującej postaci:

$$X = x_0 - (y - y_0);$$

$$Y = y_0 + (x - x_0);$$

Zdjęcie dla jasności:



Nowe współrzędne (X ;Y) obiektu:

$$X = x_0 + (x - x_0) * \cos(a) - (y - y_0) * \sin(a);$$

$$Y = y_0 + (y - y_0) * \cos(a) + (x - x_0) * \sin(a);$$

gdzie:

(x_0, y_0) - współrzędne punktu, wokół którego następuje obrót;

(x, y) - początkowe współrzędne obiektu; a jest kątem obrotu

Podział prac

Postaraliśmy się, aby podział prac w zespole był równy i sprawiedliwy. Na początku korzystając ze środków masowej komunikacji przeprowadziliśmy burzę mózgów i przygotowaliśmy plan budowy projektu. Zaczęliśmy myśleć o tym, jak działa sam Tetris.

Viktor Tykhonov był odpowiedzialny za napisanie głównego ekranu gry. Także za napisanie mechaniki gry:

- Tworzenie pola
- tworzenie tetraminy
- przesuwanie , obracanie i usuwania linijki tetraminy.

Grzesiek Kalarus był odpowiedzialny za część graficzną i muzyczną, a także za licznik pokazujący wynik na koniec gry i sam koniec gry.

Lista kroków, aby uruchomić grę:

1. Trzeba pobrać skompresowany folder ze wszystkimi elementami gry, dokumentacją i sprawozdaniem.
2. Wypakować go bezpośrednio na dysku D. Ścieżka pliku powinna wyglądać następująco: D:\ , tam powinien znajdować się folder Tetris, a w folderze katalogi: Build, Debug, html, latex, Tetris oraz plik doxywizard i projekt Tetris z rozszerzeniem .sln
3. Należy przejść do folderu Debug, który znajduje się w głównym folderze tetris: D:\Tetris\Debug, w nim trzeba odszukać plik o nazwie Tetris z rozszerzeniem .exe