**Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Programowanie w C++**

**Dokumentacja projektu:**

***Zmodyfikowane Kółko i Krzyżyk***

**Wykonała:**

**Patrycja Nędza**

**Rzeszów 2019**

***Założenia projektu zmodyfikowanej gry “Kółko   
i Krzyżyk”***

1. Przy starcie aplikacji użytkownik ma możliwość wyboru trybu gry:

* gracz vs gracz
* gracz vs komputer



2. Po wyborze trybu gry, użytkownik jest zobowiązany do wyboru symbolu i koloru danego symbolu oraz do wybrania opcji, który z graczy zaczyna pierwszy.

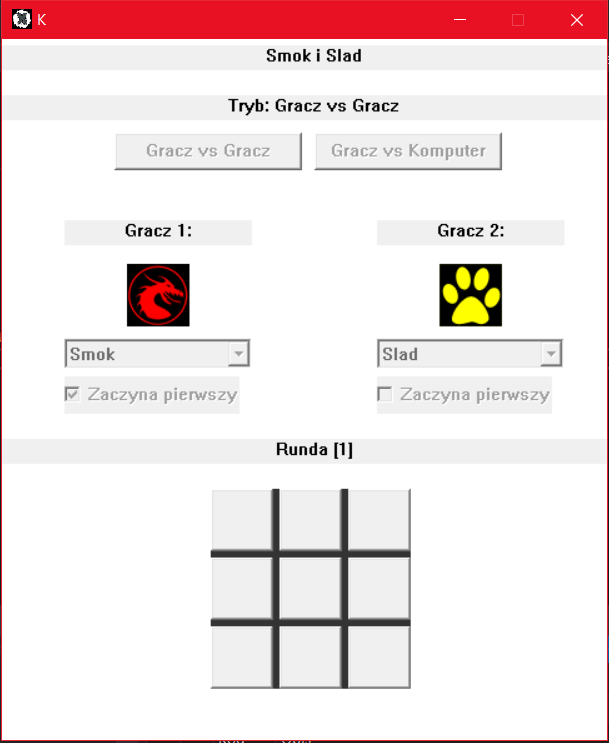


3. Gracze mają do dyspozycji kilka symboli i kolorów, którymi mogą grać:

***(Gra nie pozwala na wybranie tej samej ikonki i koloru obu graczom)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | ***KOLOR*** | | | | | |
| niebieski | zielony | czerwony | rożowy | turkusowy | żółty |
| ***S***  ***Y***  ***M***  ***B***  ***O***  ***L***  ***E*** | C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\kolko.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\krzyzyk.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\trojkat.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\kwadrat.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\BMW.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\skoda.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\ferrari.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\smok.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\slad.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\francois.bmp |  |  |  |  |  |  |
| C:\Users\Patrycja Nędza\source\repos\Kolko_i_Krzyzyk\Kolko_i_Krzyzyk\symbols\templariusz.bmp |  |  |  |  |  |  |

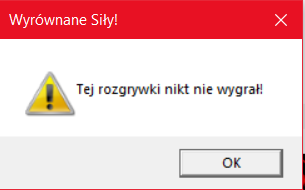
Gra po wyborze rozpoczyna się, pole gry pokazuje się.



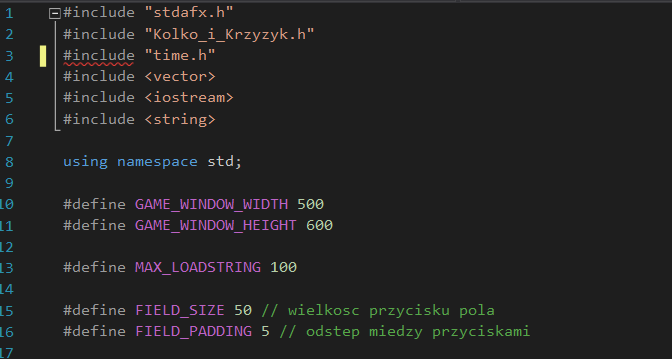
Przy każdym kliknięciu zmienia się numer rundy.

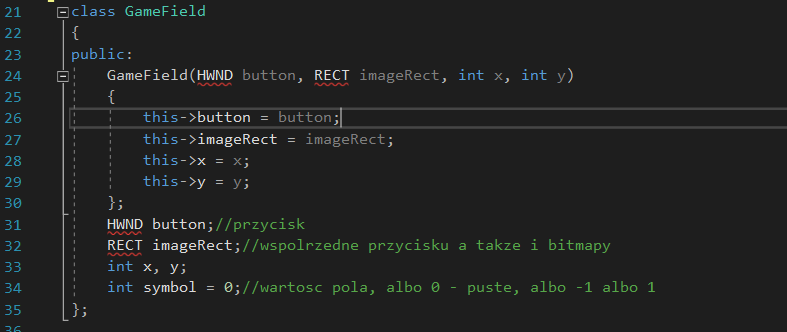


W zależności od rezultatu i trybu gry, na ekranie pokazują się odpowiednie komunikaty, np:

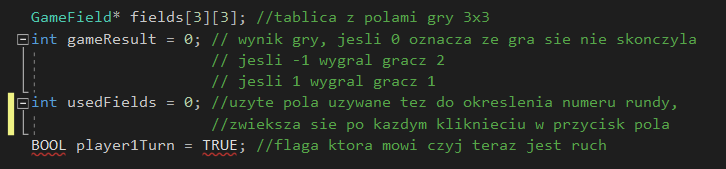


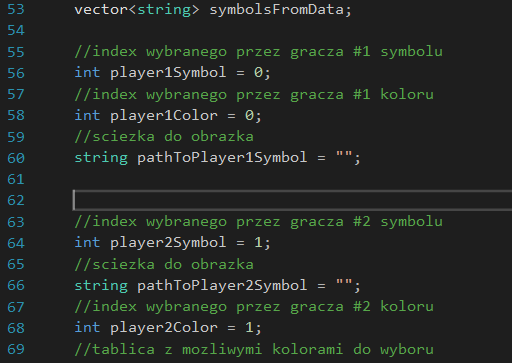
***Opis kodu programu:***

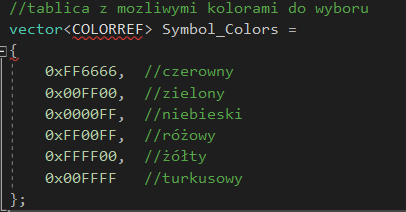


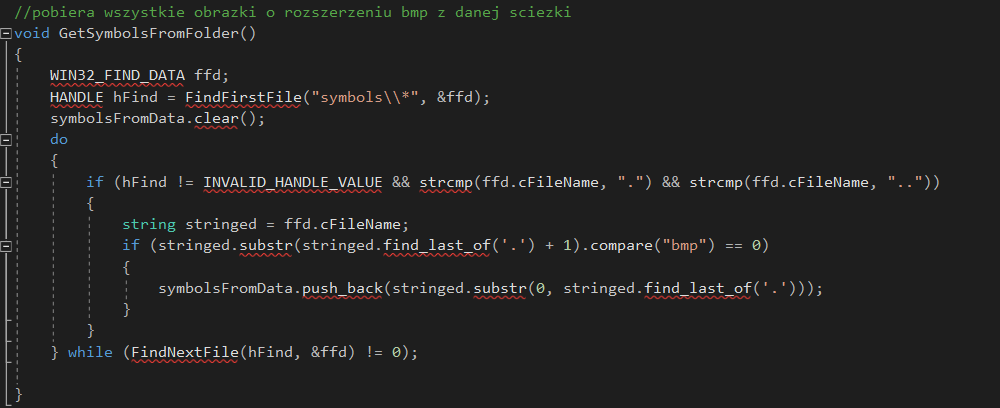


Klasa GameField ma w sobie przycisk, rozmiar (RECT) przycisku, pozycje x i y, a także co się w nim znajduje (0, 1 lub -1).

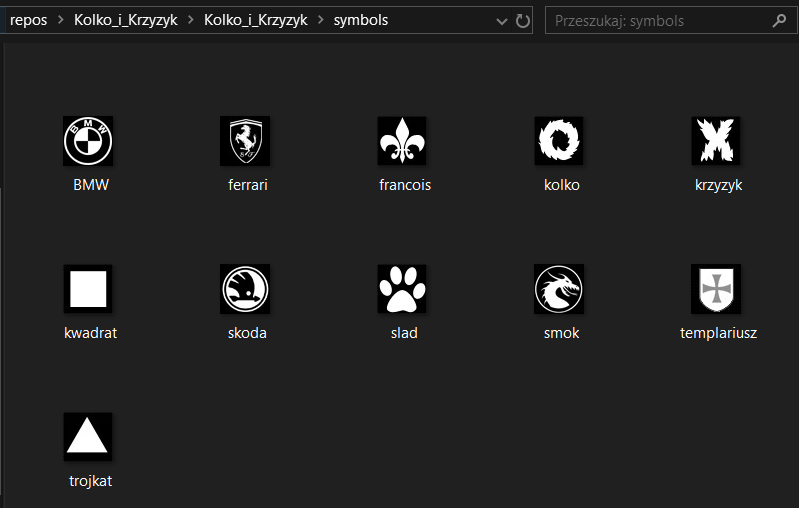


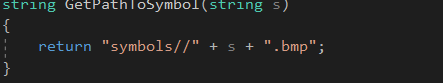




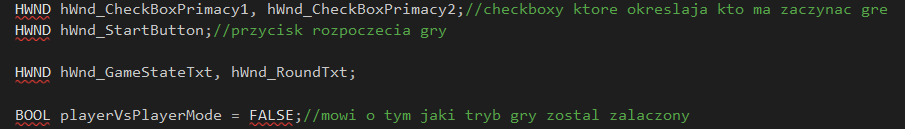


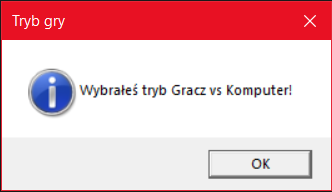
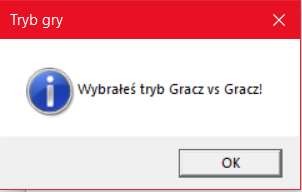
* Folder w którym znajdują się symbole .bmp

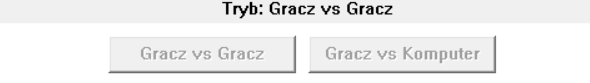




Łączenie string ze ścieżką, dodając do niego rozszerzenie i folder w którym znajdują się ikona.





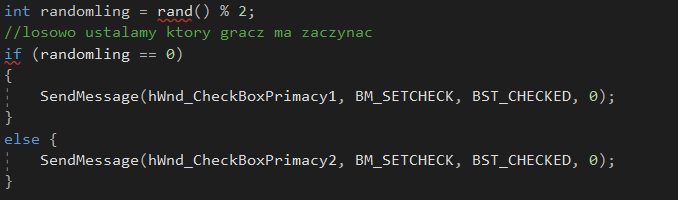


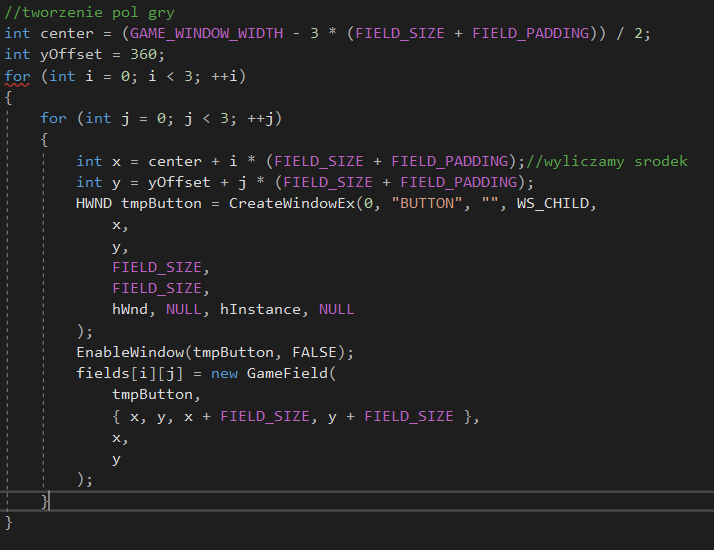
Teraz nie można już zmienić trybu gry. Tak samo jak poprzednich wariantów, po naciśnięciu przycisku „rozpocznij grę!”

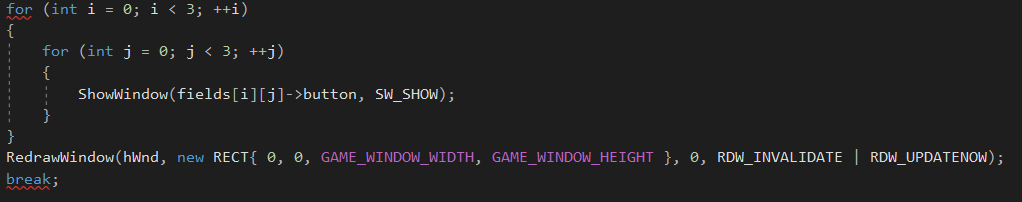




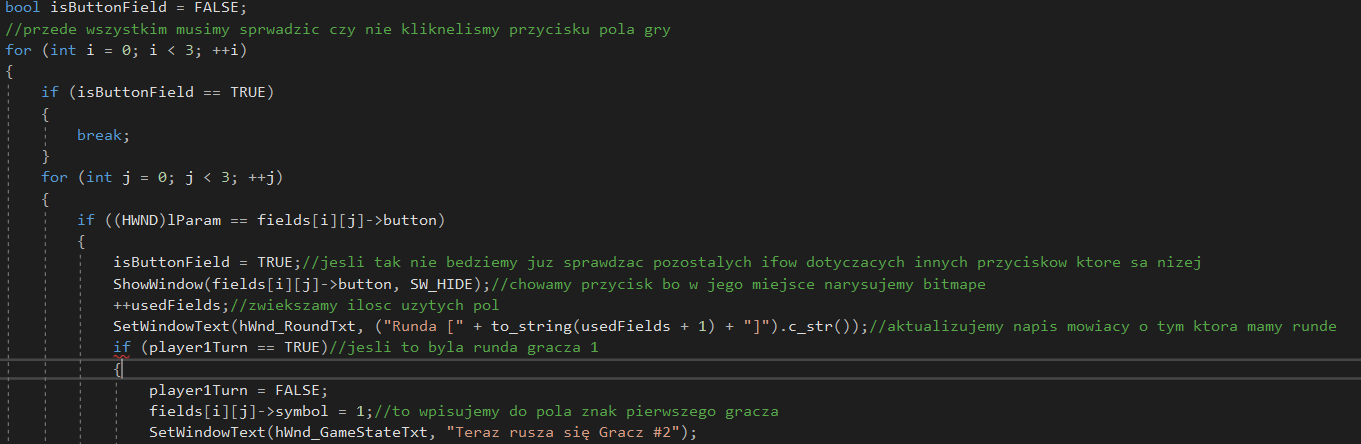
Przy starcie pierwszeństwo jest wyznaczane losowo, lecz użytkownik może to zmienić, klikając w odpowiednie miejsce.

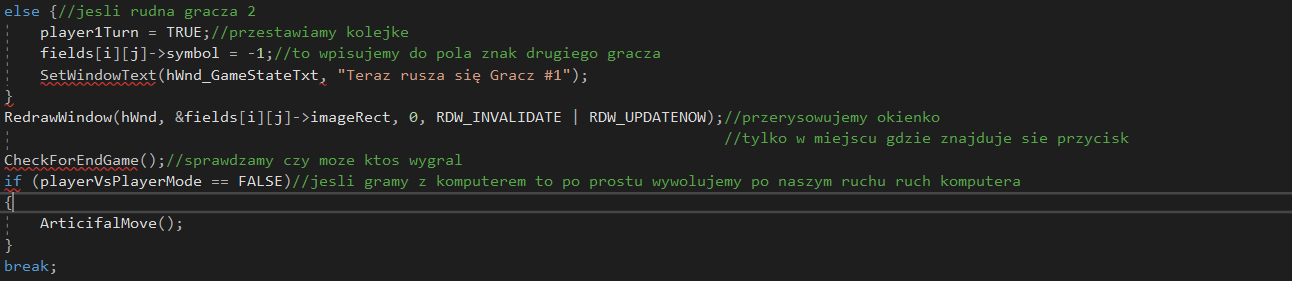


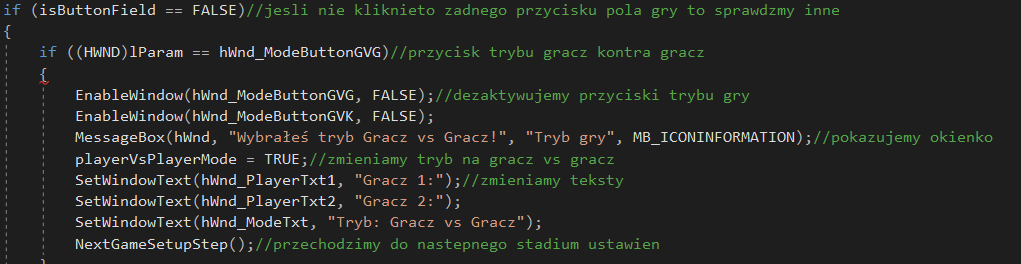


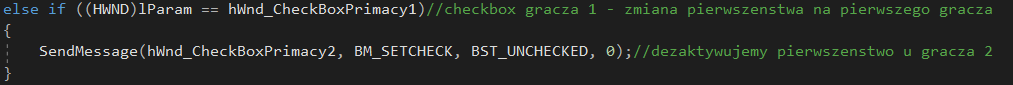


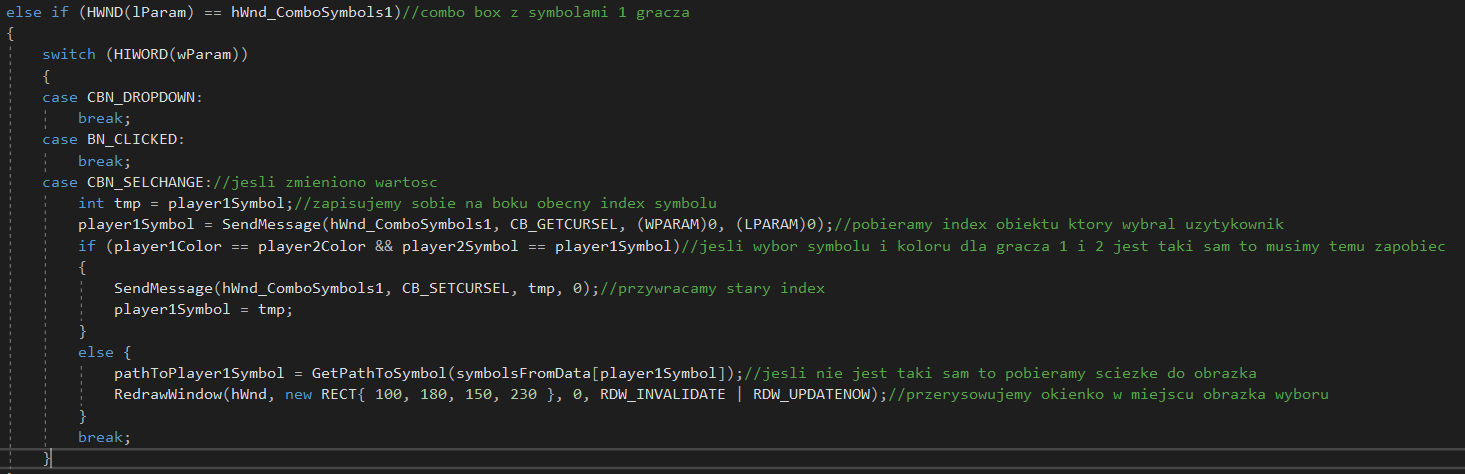
Pokazanie pola gry 3x3.

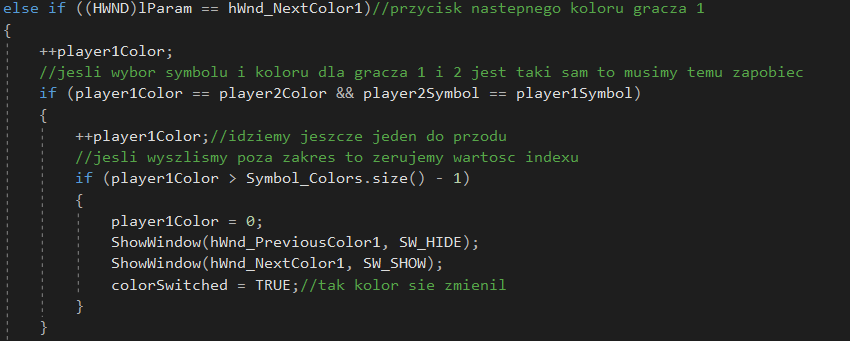


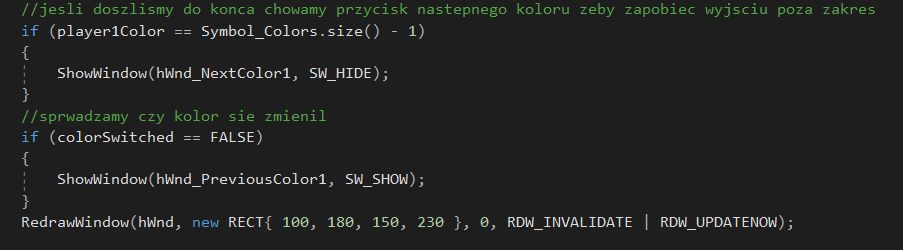


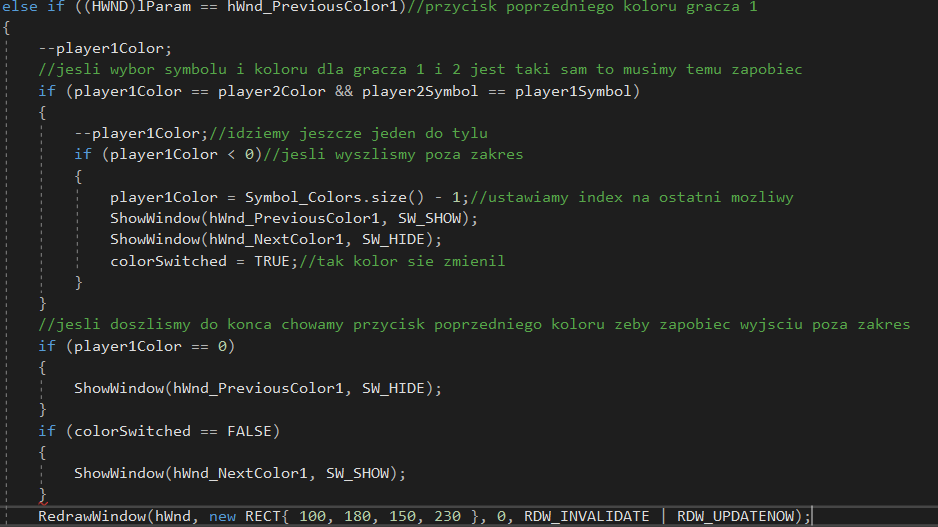








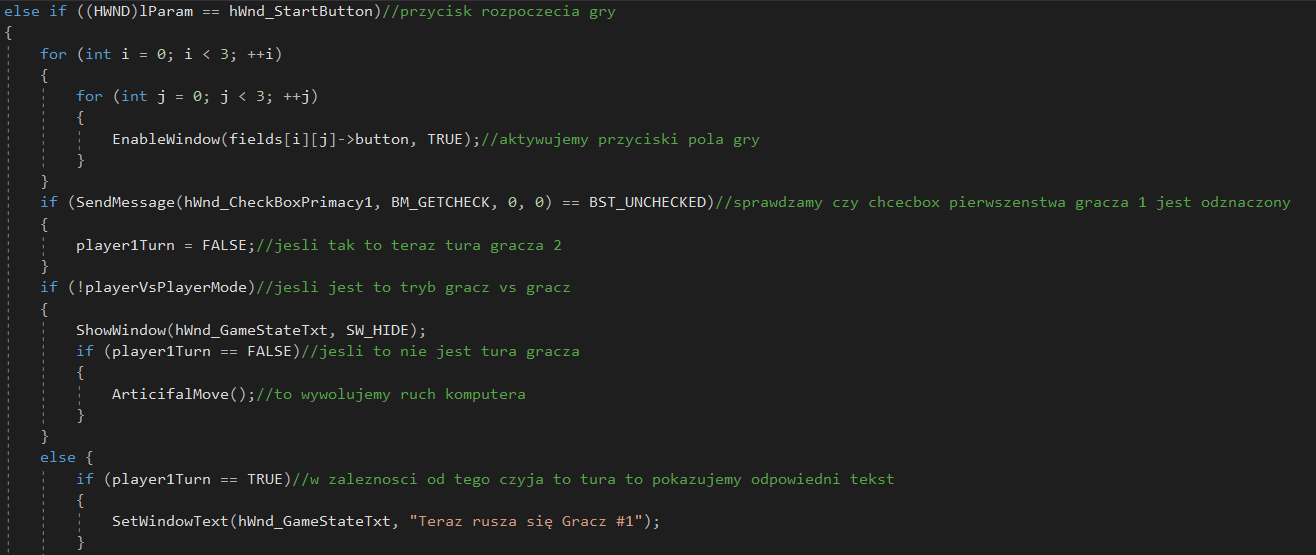


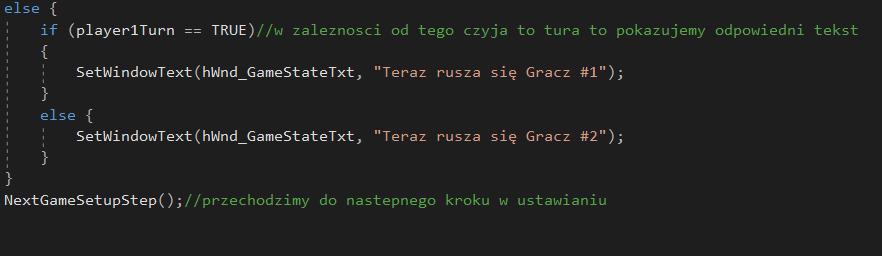


Powyższe screeny są analogiczne do gracza nr.2 lub komputera, ponieważ komputer zawsze jest graczem nr.2

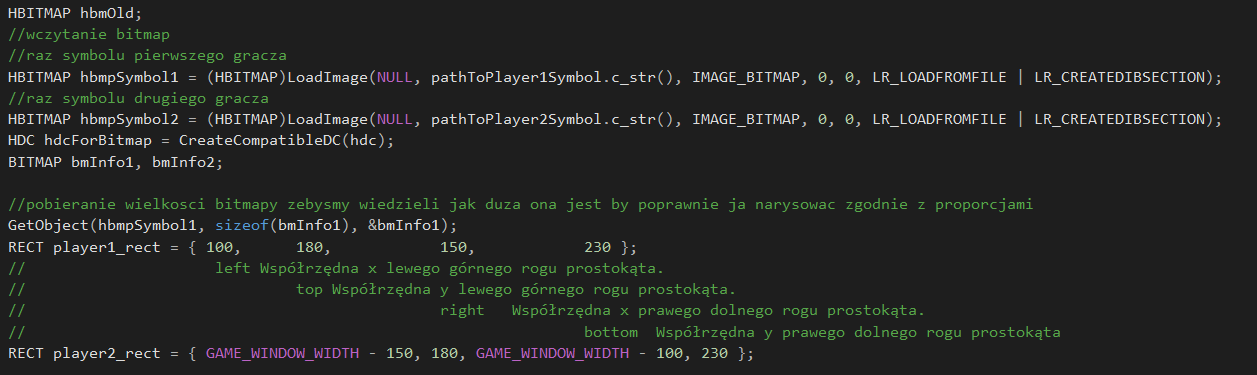
* Czas rozpocząć grę:



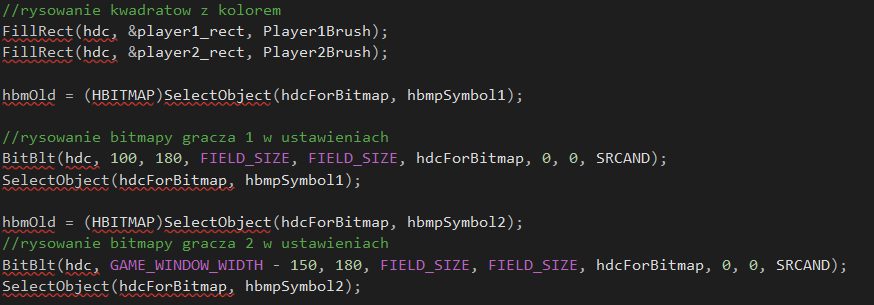


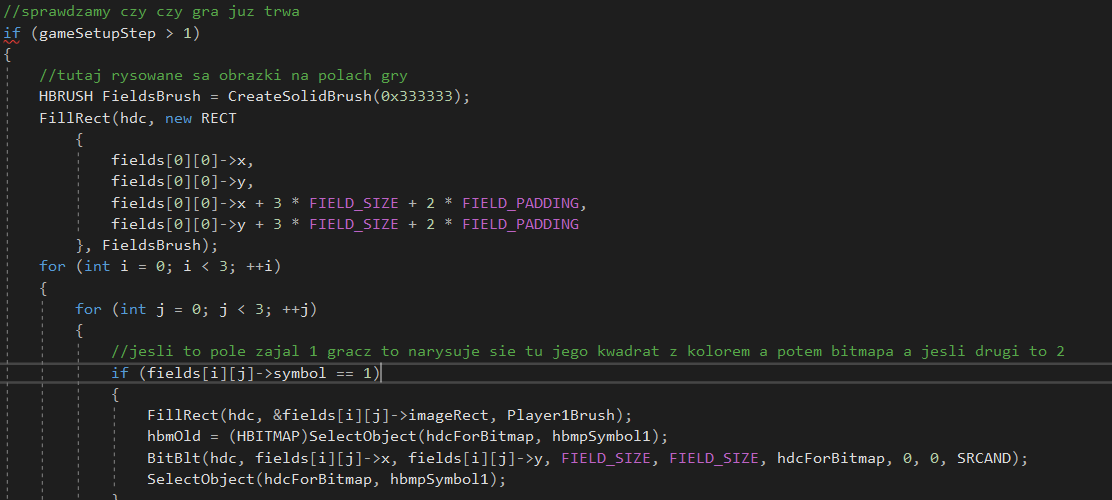


Jak powstają symbole bitmapy?



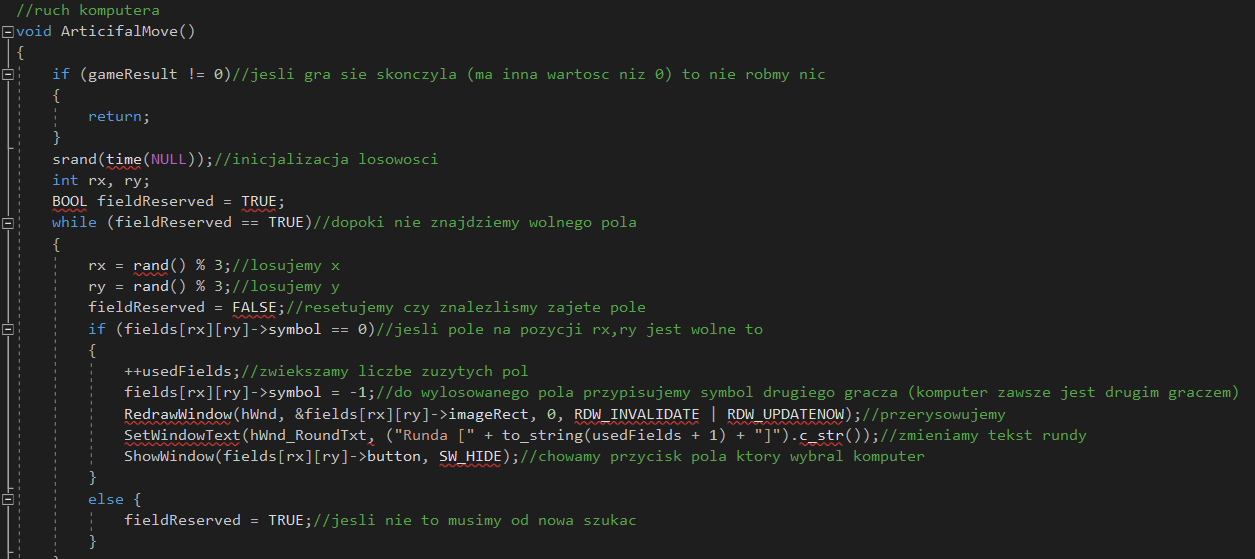
Malowanie symbolu na dany kolor polega na tym, że najpierw rysowany jest kwadrat o danym kolorze a nastepnie nakładany jest na niego symbol bitmapy z odpowiednim parametrem SRCAND (pobieranie pikseli z obrazka).



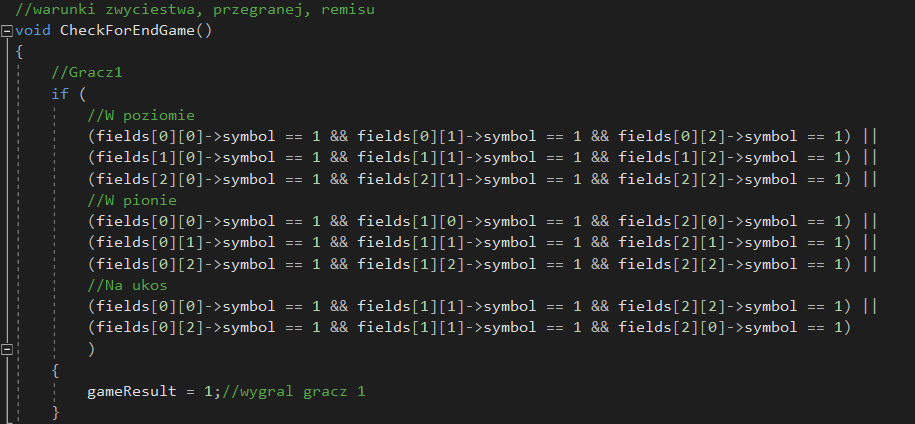


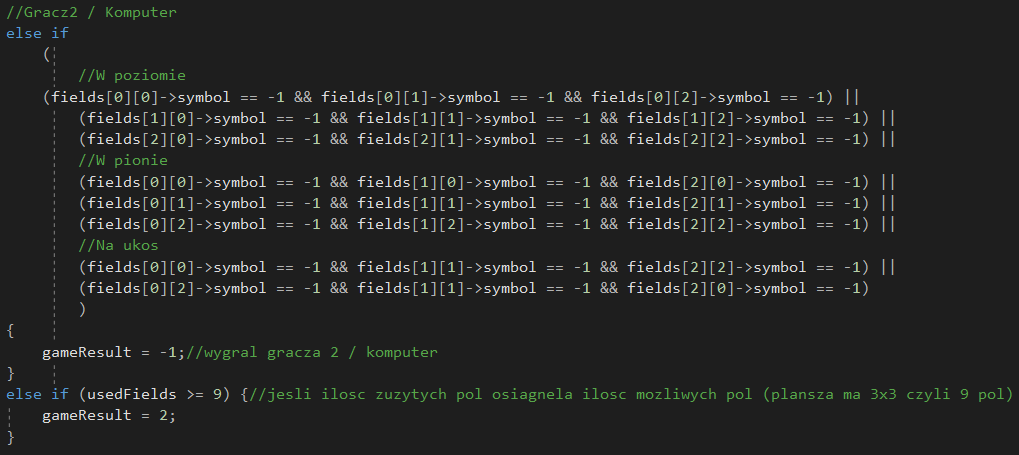
Komputer wybiera swoje pola na zasadzie losowości i zawsze jest graczem nr.2.

W tabeli 3x3 nie pojawi się żaden symbol dopóki gracz nr.1 nie naciśnie żadnego pola.

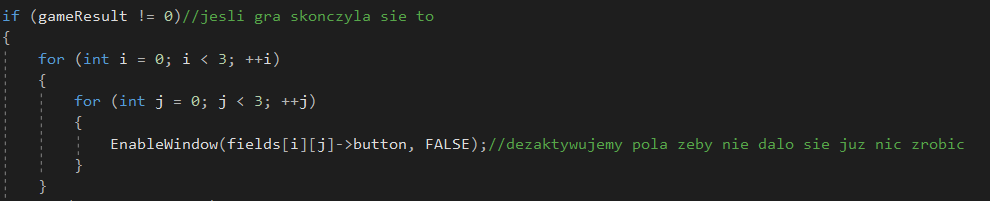


* Warunki wygranej, przegranej, remisu:

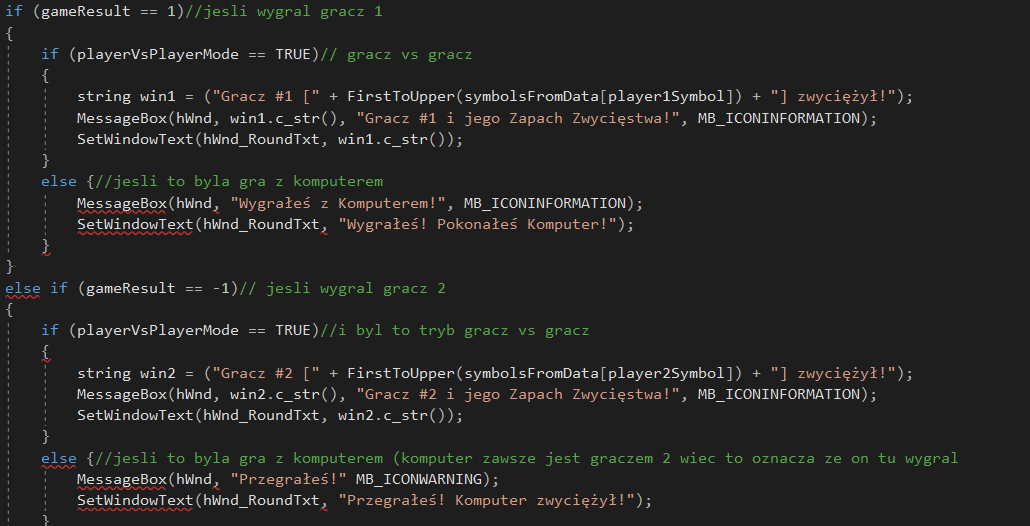


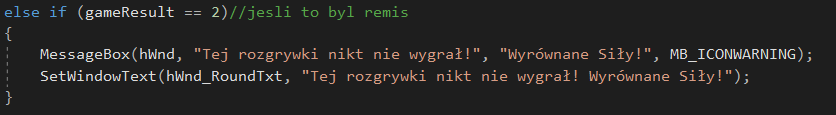


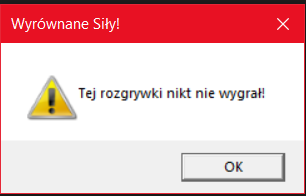
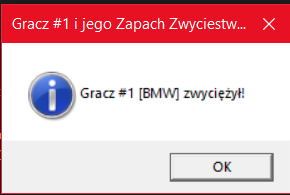
* Koniec gry:



* Komunikaty:







W zależności od wyniku (w trybie gracz vs gracz i gracz vs komputer), pojawiają się odpowiednie komunikaty.

***Technologie wykorzystane w projekcie***

***Język C++***

Język C++ to jeden z najpopularniejszych, kompilowanych języków programowania. Wywodzi się on bezpośrednio z języka C, aczkolwiek ma z nim niewiele wspólnego. Jest on często określany jako język niskopoziomowy.

Ma on bardzo dużo wspólnego z językiem binarnym, co znacznie utrudnia pisanie aplikacji, ale jednocześnie - znacznie zwiększa możliwości programistyczne.

**Gdzie wykorzystuje się język C++?**

* W aplikacjach zarządzanych z poziomu systemu operacyjnego Windows
* W sterownikach sprzętowych, wykorzystywanych np. pod Windows.
* W grach komputerowych.

**Cechy języka C++:**

* Język kompilowany. Oznacza to, że pracuje on bardzo szybko (nawet w aplikacjach zawierających dużą liczbę wierszy)
* Umożliwia częściowe operowanie na urządzeniach hardware. Jest to możliwe dzięki obsługiwanych wstawkach ASM.
* Jest kompatybilny wstecznie z językiem C. Oznacza to, że aplikacje napisane pod C, prawidłowo funkcjonują również pod jego rozwiniętą wersją.
* Obsługuje mnóstwo bibliotek, dzięki czemu jego możliwości są jeszcze większe, niż w przypadku konkurencyjnych środowisk.

***Microsoft Visual Studio***

**Microsoft Visual Studio** – [zintegrowane środowisko](https://pl.wikipedia.org/wiki/Zintegrowane_%C5%9Brodowisko_programistyczne) programistyczne firmy [Microsoft](https://pl.wikipedia.org/wiki/Microsoft). Jest używane do tworzenia oprogramowania konsolowego oraz z [graficznym interfejsem użytkownika](https://pl.wikipedia.org/wiki/Graficzny_interfejs_u%C5%BCytkownika).

Aplikacje mogą być pisane na platformy:

* [Microsoft Windows](https://pl.wikipedia.org/wiki/Microsoft_Windows),
* [Windows Phone](https://pl.wikipedia.org/wiki/Windows_Phone),
* [Windows CE](https://pl.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://pl.wikipedia.org/wiki/.NET_Framework),
* [Microsoft Silverlight](https://pl.wikipedia.org/wiki/Microsoft_Silverlight),

***Jak uruchomić projekt?***

* Sposób 1:

Plik z rozszerzeniem sln (Visual Studio Solution File) otworzymy za pomocą programu Visual Studio, przy pomocy wbudowanego tam kompilatora.

***Jeżeli*** wystąpiłby błąd podczas kompilacji powiadamiający o braku zestawu Windows 10 SDK, należy postępować zgodnie  
z podpowiedziami ukazanymi w programie Visual Studio.

* Sposób 2:

Otworzyć folder debug i uruchomić plik z rozszerzeniem .exe