

# Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Filip Kaśkos 249009

29.05.2020

Prowadzący: Mgr inż. Marcin Ochman

Termin: wtorek 15.15

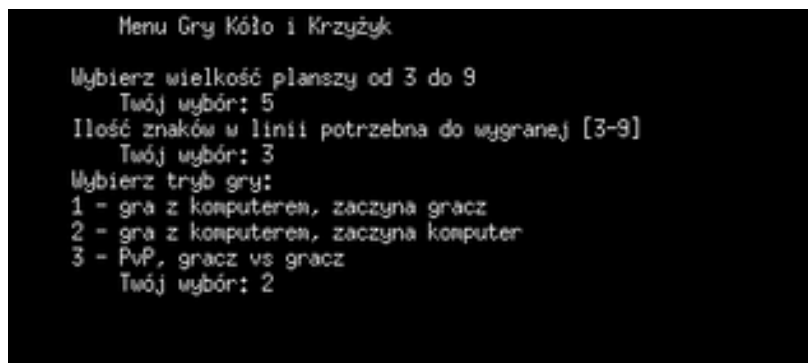
## 1. Wprowadzenie

Zadaniem do wykonania było stworzenie gry, w której dzięki odpowiednim algorytmom możemy zmierzyć się z komputerem. Wybrana przeze mnie gra to kółko i krzyżyk. Elementem sztucznej inteligencji jest algorytm oparty o strategię MinMax. Do stworzenia interfejsu graficznego gry wykorzystałem bibliotekę SFML.






## 2. Opis gry.

Gra w kółko i krzyżyk jest wszystkim dobrze znana. Aby wygrać na planszy należy umieścić ciąg takich samych znaków w pionie, poziomie lub po przekątnej. Podstawowa wersja gry składa się z planszy 3x3 i rozgrywka na niej jest dosyć prosta. W projekcie rozmiar planszy oraz liczba znaków konieczna do wygrania jest jednak modyfikowalna i zależy od gracza. Gracz wprowadza rozmiar planszy (od 3 do 9) oraz liczbę znaków określających wygraną (od 3 do 9, liczba znaków nie może być większa od rozmiaru planszy). W przypadku większej planszy i sumy znaków na przykład 3, wygrana możliwa jest w pionie, poziomie oraz na **głównych** przekątnych. Gracz ma do wyboru także sposoby rozgrywki. Może grać z komputerem sam zaczynając rozgrywkę, grać z komputerem kiedy to on rozpoczyna rozgrywkę oraz grę PvP, w której może zmierzyć się z innym graczem. W grze przyjętym znakiem rozpoczynającym rozgrywkę jest zawsze kółko („O”). Menu gry znajduje się w konsoli, jednak po wybraniu opcji uruchamia się okno z grą. Umieszczanie znaków na planszy możliwe jest poprzez klikanie w odpowiednie pola. Po zakończonej rozgrywce pojawia się komunikat o wygranej lub remisie.









































### Obrazy z rozgrywki



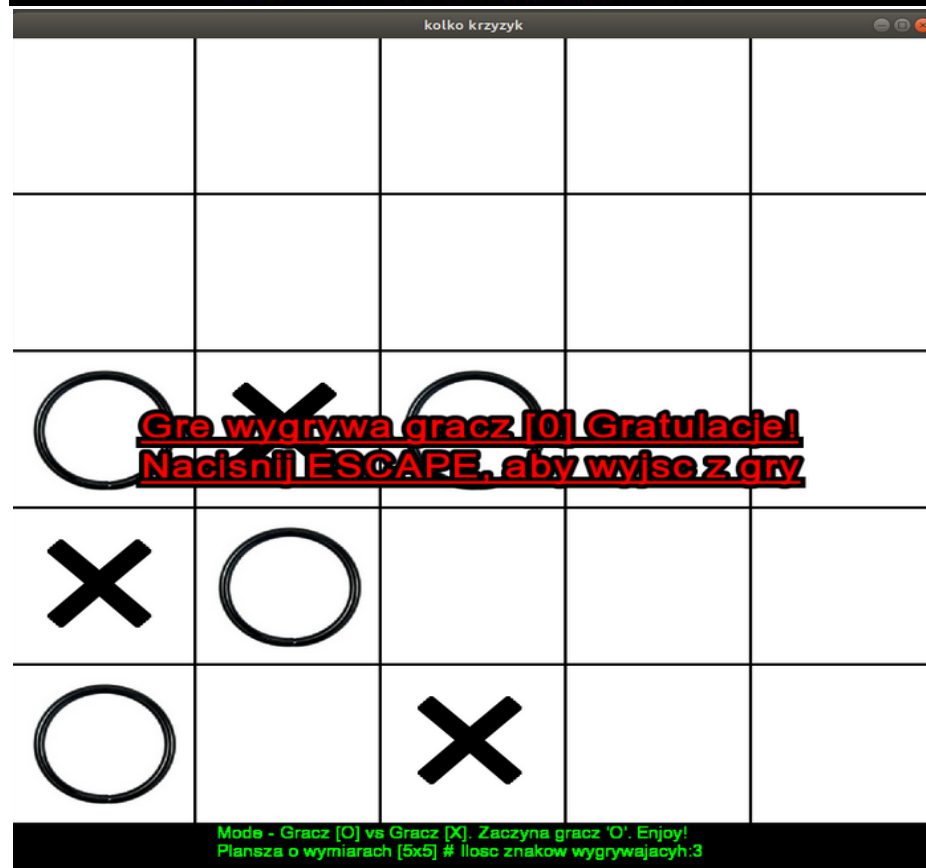
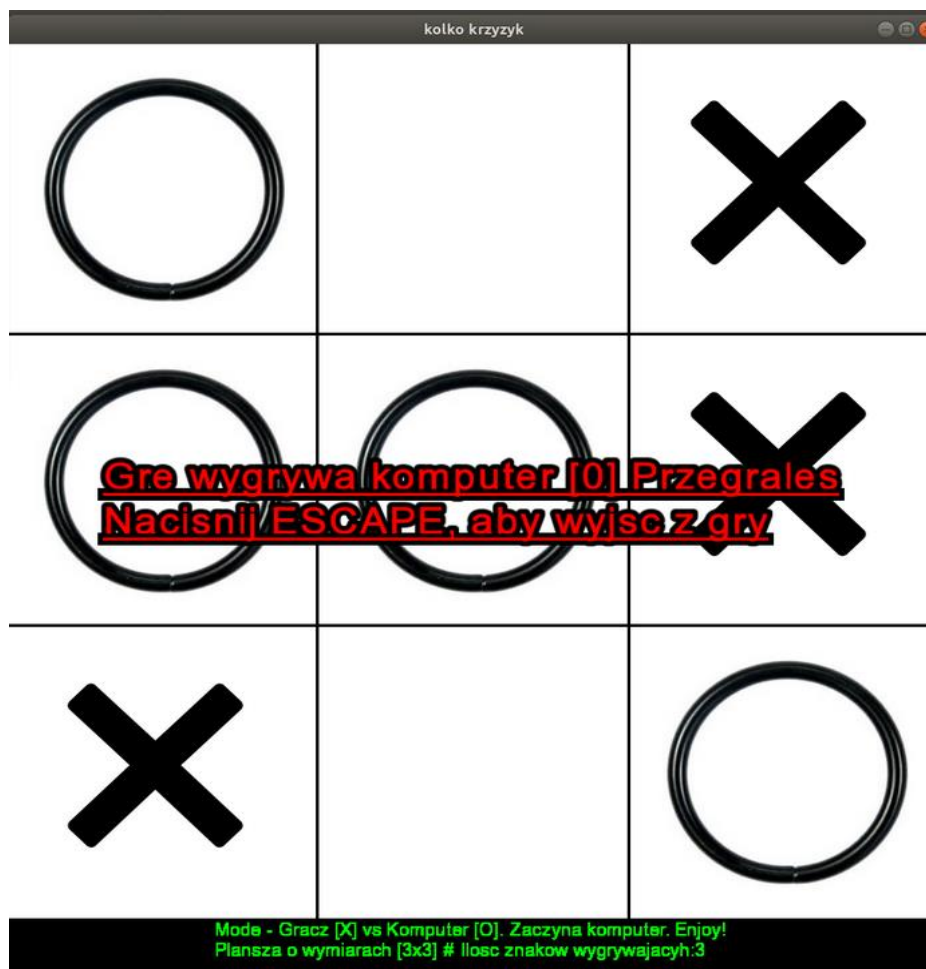
```
Menu Gry Kółko i Krzyżyk
Wybierz wielkość planszy od 3 do 9
Twój wybór: 5
Ilość znaków w linii potrzebna do wygranej [3-9]
Twój wybór: 3
Wybierz tryb gry:
1 - gra z komputerem, zaczyna gracz
2 - gra z komputerem, zaczyna komputer
3 - PvP, gracz vs gracz
Twój wybór: 2
```

kolko krzyzyk				
				
				

Mode - Gracz [X] vs Komputer [O]. Zaczyna komputer. Enjoy!  
Plansza o wymiarach [5x5] # Ilosc znakow wygrywajacyh:3

kolko krzyzyk								
								
								
								
								
								
								
								
								
								

Mode - Gracz [O] vs Gracz [X]. Zaczyna gracz 'O'. Enjoy!  
Plansza o wymiarach [9x9] # Ilosc znakow wygrywajacyh:9



Okno gry ma rozmiar 900x960. Pola do gry generowane są jako kwadraty z czarną obwódką, z rozmiarem dostosowanym do rozmiaru okna oraz wielkości planszy do gry, zapisane są w wektorze. Znaki gry czyli kółko i krzyżyk są obrazkami (sf::Sprite – biblioteka SFML), które zapisane są w wektorze. Następnie zarówno pola jak i znaki renderowane są w oknie gry. Z racji tego, że pola rozmieszczone są obok siebie a proces wychwytywania kliknięcia odbywa się za pomocą wczytania położenia współrzędnych kursora myszy zdarza się, że klikając na czarne linie oddzielające pola, tworzone są dwa znaki na sąsiadujących ze sobą polach. Zakończenie rozgrywki sygnalizowane jest poprzez wyświetlenie napisu(sf::Text) na środku okna. Zamknięcie okna z grą obsługiwane jest zarówno poprzez przycisk Escape oraz kliknięcie „x” w prawym górnym rogu. Operacje uzyskane są za pomocą odpowiednich części biblioteki SFML (sf::Event). Cała gra bazuje na jednej klasie „Game”, w której zawarte są odpowiednie funkcje zarówno do obsługi gracza oraz sztucznej inteligencji. Rozgrzywka możliwa jest dzięki pojedynczemu obiektowi klasy „Game” oraz zewnętrznej głównej pętli gry.

### 3. Sztuczna inteligencja

Sztuczna inteligencja w grze zaprojektowana jest z wykorzystaniem algorytmu min-max. Algorytm w grze takiej jak kółko i krzyżyk (suma zerowa) polega na przeszukiwaniu drzewa rozwiązań i wybieraniu rozwiązania, które daje graczowi wykonującemu maksymalne zyski przy minimalizacji zysków przeciwnika. Sprawdzając wygraną na planszy zwracany jest wynik 10, -10 lub 0 odpowiednio dla wygranych poszczególnych znaków i remisu. Daje nam to możliwość określenia wartości wykonywanego ruchu. Jeśli po wykonaniu ruchu osiągamy zwycięstwo ruch ten ma wartość 10 lub -10. Wywołując algorytm rekurencyjnie sprawdzamy dla każdego naszego ruchu możliwe ruchy przeciwnika oraz drogę do końcowej wygranej. Kalkulujemy, która droga do wygrania jest najkorzystniejsza i wykonujemy dany ruch. Jednym z usprawnień algorytmu jest odejmowanie głębokości drogi o wartości ruchu. W przypadku różnych możliwych dróg prowadzących do zwycięstwa wybierana jest droga najkrótsza. W pełni wykonująca się algorytm powoduje, że wygrana z komputerem jest niemożliwa, gdyż jest on perfekcyjny. Jednak z racji tego, iż przy większej wielkości planszy drzewo rozwiązań jest niezwykle rozbudowane, a długość sprawdzenia wszystkich możliwych ruchów jest ogromna, głębokość wywołań rekurencyjnych jest ograniczona. Dla uzyskania optymalnej rozgrywki maksymalna głębokość wywołań zależna jest od wielkości planszy. Dla planszy 3x3 jest ona równa, dla pozostałych mniejszych od 7x7 równa się 3, a dla reszty 2. Pozwala to zoptymalizować rozgrywkę i zrównać czas reakcji komputera na wszystkich wielkościach planszy. Zmniejszenie maksymalnej głębokości „ogłupia” komputer jednak jest ono konieczne.

#### **4. Wnioski**

Testując grę we wszystkich możliwych opcjach rozgrywki, mogę stwierdzić, że jest ona zoptymalizowana i działa bez większych problemów. Sztuczna inteligencja na planszy 3x3 jest nie do pokonania. Algorytm MinMax spełnia więc swoje zadanie. Przy większej planszy oraz mniejszej od jej wymiarów liczby znaków wygrywających możliwa jest wygrana. Na podstawie tej prostej rozgrywki możemy wysnuć wniosek, że posiadając odpowiednie zasoby sprzętowe możliwe jest stworzenie przeciwnika perfekcyjnego, którego pokonanie jest dla człowieka rzeczą niemożliwą. Grafika rozgrywki nie jest zaawansowana, aczkolwiek efekty pracy z moim pierwszym interfejsem graficznym uważam za zadowalające. Projektowanie jakiegokolwiek grafiki wymaga przyswojenia wielu nowych informacji jak i poznania mechanizmów biblioteki, w tym przypadku SFML.