

**Specyfikacja funkcjonalna:**  
Kwantowe Kółko i Krzyżyk

**Michał Żdanuk**



Wydział Elektryczny  
Politechnika Warszawska  
Marzec 2022

## 1 Cel dokumentu

Dokument powstał, by przybliżyć czytelnikowi działanie gry - **Kwantowe Kółko i Krzyżyk**. Przedstawia on także zasady rozgrywki oraz wyjaśnia zachowanie programu w zależności od działań podejmowanych przez graczy.

Poza tym dokument zawiera:

- szablon interfejsu graficznego okna rozgrywki z uwzględnionymi komunikatami (błędu/informacyjnymi),
- schemat graficzny przykładowej planszy,
- krótki scenariusz poparty grafikami mający na celu przybliżenie zasad gry,

## 2 Cel projektu

Celem projektu jest napisanie programu w języku JAVA, pozwalającego na przeprowadzenie rozgrywki w okienku terminala dla dwóch graczy. W przypadku sprawnego zaimplementowania wersji "konsolowej" gra zostanie rozszerzona o możliwość zagrania w wersji "okienkowej" (tzn. z interfejsem graficznym).

Rozgrywka w **Kwantowym Kółku i Krzyżyk** będzie przebiegać na jednym urządzeniu stacjonarnym. Gracz wybiera i zaznacza swoim znakiem odpowiednią komórkę poprzez podanie do terminalu cyfry odpowiadającej danej komórce.

Kluczowa będzie dla mnie praca z uwzględnieniem norm programowania obiektowego tzn. poprawne korzystanie z dziedziczenia klas, agregacji czy kompozycji. Wysoki nacisk położę również na właściwe wykorzystanie podstawowych zasad obiektowości: hermetyzacji interfejsów, abstrakcji oraz hierarchizacji obiektów.

## 3 Wstęp teoretyczny

### 3.1 Zasady rozgrywki

**Gra Kwantowe Kółko i Krzyżyk** to uogólnienie "kwantowe" standardowej gry Kółko i Krzyżyk, w której ruchy graczy są "superpozycjami" ruchów z klasycznego wariantu gry. Ułożenie odpowiednich trzech znaków w rzędzie: pionowo, poziomo lub po skosie zapewnia graczowi zwycięstwo w rozgrywce.

### 3.2 Wynalazcy gry

Za wymyślenie gry w kwantowe kółko i krzyżyk odpowiadają: **Allan Goff** oraz **Novatia Labs**.

Określili oni grę jako sposób "wprowadzenia fizyki kwantowej bez użycia matematyki" równocześnie tworząc narzędzie pozwalające na bardzo podstawowe zrozumienie idei mechaniki kwantowej.

### 3.3 Podłoże fizyki kwantowej w rozgrywce

Twórcy tworząc kwantowe kółko i krzyżyk chcieli umieścić w rozgrywce koncept trzech zjawisk układów kwantowych:

- **superpozycję (ang. superposition)** - zdolność obiektów kwantowych do bycia w dwóch miejscach jednocześnie;
- **splątanie (ang. entanglement)** - zjawisko, w którym odległe obiekty systemu kwantowego wykazują korelację, której nie można wytłumaczyć żadną zależnością przyczynowo-skutkową;
- **zawalenie się/rozłączanie (ang. collapse)** - zjawisko, w którym następuje redukcja stanów kwantowych do stanów klasycznych.

### 3.4 Komórka

Komórka - najmniejszy, niezależny fragment rozgrywki. Tak samo jak w standardowej wersji gry w komórki wpisujemy znak (kółko lub krzyżyk). Różnica polega na tym, że w kwantowej wersji w jedno pole możemy wpisać wiele "małych" znaków (w szczególnym przypadku maksymalnie aż 8). Kiedy nastąpi **splątanie**, a następnie **zawalenie się** odpowiednie "małe znaki" zamienią się na stałe w "duże" znaki wówczas w każdym takim polu będzie znajdować się już tylko jeden znak.

Każda komórka może znajdować się w jednym ze ściśle określonych stanów:

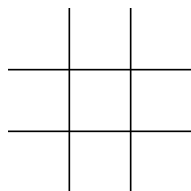
- komórka pusta (początkowa sytuacja, gdy w komórce nie jest wpisany żaden znak)

- komórka wypełniona małymi znakami (sytuacja, gdy gracze wpisali już swoje znaki w komórkę, ale nie wystąpiło jeszcze **splątanie**)
- komórka splątana (sytuacja, gdy właśnie nastąpiło **splątanie** i gracz musi wybrać, które znaki "małe" zamienią się na stałe w "duże")
- komórka po procesie "zawalenia się" (sytuacja, gdy nastąpiło **zawalenie się**; w danej komórce występuje tylko jeden znak, którego nie można zmienić)

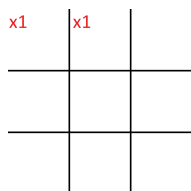
**Uwaga!** Znaki "małe" to znaki, które są wpisywane przez graczy na początku każdej swojej tury. Znaki "duże" to znaki powstałe w procesie **zawalenia się**, których nie można już zmienić, a odpowiednie ich ułożenie zapewnia graczowi zwycięstwo.

### 3.5 Plansza

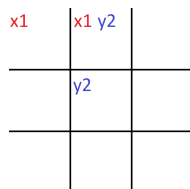
Plansza - rozbudowany element rozgrywki, który składa się z dziewięciu skorelowanych ze sobą komórek (ułożenie 3x3). W każdej turze gracz wpisuje dwa swoje znaki w komórki (**uwaga:** każdy ze znaków musi trafić do innej komórki!). W przypadku gdy wystąpi cykl (z jednej komórki można przejść do innej i wrócić z powrotem do pierwszej, ale inną ścieżką) gracz wybierze, który ze znaków "małych" zamieni się w duży. Takie działanie spowoduje "reakcję łańcuchową", ponieważ przerwany zostanie cykl i pozostałe komórki tworzące cykl wypełnią się odpowiednimi "dużymi" znakami. Poniżej prezentuję krótki scenariusz rozgrywki, gdzie nastąpi **splątanie** - w wyniku powstania cyklu:



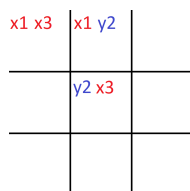
Rysunek 1: plansza na początku rozgrywki (przed wykonaniem pierwszego ruchu)



Rysunek 2: plansza po wykonanym pierwszym ruchu - ruch gracza x



Rysunek 3: plansza po wykonanym drugim ruchu - ruch gracza y

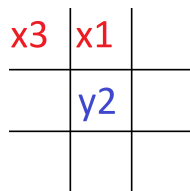


Rysunek 4: plansza po wykonanym trzecim ruchu (splątanie) - ruch gracza x

Cykl tworzą komórki: lewa górna, środkowa górna oraz komórka centralna:

- splątana jest komórka lewa górna z komórką środkową górną poprzez x1
- splątana jest komórka środkowa górna z komórką środkową centralną poprzez y2
- splątana jest komórka centralna z komórką lewą górną poprzez x3

Gracz przeciwny (tzn. nie ten który utworzył cykl) wybiera, które znaki zmieniają się w "duże". Przykładowo:



Rysunek 5: plansza po wybraniu przez gracza y "dużych" znaków

### 3.6 Zakończenie rozgrywki

Pomyślne zakończenie rozgrywki następuje tylko w dwóch przypadkach:

- gdy któryś z graczy ułoży z trzech "dużych" znaków linię pionową/poziomą/po skosie - gracz, któremu udało się to jest zwycięzcą rozgrywki
- wypełnione zostaną wszystkie komórki planszy "dużymi" znakami, a żaden z graczy nie ułoży rzędu trzech znaków - żaden z graczy nie wygrywa

**UWAGA:** może wystąpić sytuacja szczególna, gdy dwóch graczy jednocześnie ułoży linię złożoną z trzech znaków! Wówczas zwycięzcą zostaje gracz, którego linia zawiera znak o mniejszym indeksie.

x3	x1	x4
y6	y2	y4

Rysunek 6: plansza z utworzonymi liniami przez obu graczy

Najmniejszy indeks znaków w linii gracza x to 1, a w przypadku gracza y to 2. Zatem zwycięzcą rozgrywki zostaje gracz x.

## 4 Sterowanie

Sterowanie grą jest trywialne. Gracz by wykonać ruch jedyne co musi zrobić to podać z klawiatury w terminalu cyfrę odpowiadającą polu, które chce wybrać (cyfra z zakresu 0-8). Poniżej zaprezentowany jest schemat obrazujący przypisanie cyfr do konkretnych pól:

0	1	2
3	4	5
6	7	8

Rysunek 7: plansza na początku rozgrywki

W przypadku wystąpienia splątania przeciwny gracz wybiera, które znaki zmieniają się w "duże".

## 5 Komunikaty aplikacji

W aplikacji komunikaty podzieliłem na dwa typy:

- komunikaty informacyjne
- komunikaty błędów

Komunikaty informacyjne mają na celu pomaganie graczom rozgrywki (szczególnie tym, którzy nie są zaznajomieni z zasadami rozgrywki). W konsoli na bieżąco wyświetla się komunikat z informacją czyja jest tura. W przypadku wystąpienia "zawalenia się" (patrz podsekcja 3.3) gra poinformuje gracza o tym oraz podpowie, które znaki może wybrać.

Komunikaty błędów wystąpią w przypadku wykonania przez któregoś z graczy nieprawidłowego ruchu. Przykładowe nieprawidłowe zachowania gracza:

- gracz próbuje wybrać numer pola, który nie występuje w planszy
- gracz próbuje wybrać dwa razy te same pole w jednej swojej turze
- gdy wystąpi splątanie gracz próbuje wybrać pole puste lub pole, które nie wchodzi w skład cyklu

W każdym z powyższych scenariuszy gracz zostanie poinformowany o nieprawidłowości w jego ruchu z wyjaśnieniem w jaki sposób wykonać poprawny ruch, błędy ruch nie zostanie wykonane, a gracz zostanie poproszony o wykonanie poprawnego działania.

## 6 Testowanie programu

Aplikacja będzie podlegała testowaniu na dwa sposoby - testy statyczne oraz testy jednostkowe. Testy statyczne - ręczna analiza pisanego kodu będzie wykonywana konsekwentnie w ciągu trwania całego projektu. Testy jednostkowe będą powstawać po wykonaniu pełnego, niezależnego fragmentu aplikacji (np. po stworzeniu klasy Gracz wraz z całą jej funkcjonalnością) zostaną napisane testy jednostkowe do danego fragmentu kodu. Na koniec projektu nastąpi weryfikacja pokrycia kodu testami jednostkowymi. W przypadku stwierdzenia, że jakiś fragment kodu/jakaś sytuacja nie jest sprawdzana zostaną napisane dodatkowe testy uzupełniające takowe braki. Testy jednostkowe zostaną napisane przy pomocy **Maven'a** przy użyciu biblioteki **JUnit**.

## 7 Bibliografia

Do tworzenie powyższego dokumentu pomocne okazały się niżej wymienione źródła:

- [https://en.wikipedia.org/wiki/Quantum\\_tic-tac-toe](https://en.wikipedia.org/wiki/Quantum_tic-tac-toe)
- <http://qttt.rohanp.xyz/>
- <https://quantumfrontiers.com/2019/07/15/tiqtaqtoe/>