

# Wymagania projektu

- **Graficzny interfejs użytkownika (GUI)**

*Interface składający się z 3 paneli: menu, ustawień, gry.*

- **Poprawne działanie na 3 systemach operacyjnych**

*Linux. MacOS, Windows.*

- **Możliwość wyboru jednego z 4 trybów czasowych**

*Ograniczenia czasowe powszechnie używane w grach turniejowych. Blitz 3+2, Bullet 1+0, Rapid 10+0 czy też Classical 15+15.*

- **Szybka odpowiedź komputera**

*Odpowiedź w czasie nie większym niż 5 sekund.*

- **Materiały creative commons**

*Wszystkie assety (grafika) będą na licencji cc.*

- **Automatyczne zegary czasowe**

*Dwa zegary po stronie obu graczy służące do kontroli czasu. Wraz z końcem ruchu będzie się automatycznie zatrzymywać.*

- **Historia ruchów**

*Każdy wykonany ruch w danej partii szachów będzie widoczny w bocznym panelu z grą. Na koniec gry historia będzie zapisywana do pliku.*

- **Własne SI**

*Program nie będzie podłączany do istniejących już baz SI, lecz zostanie utworzony nowy 'komputer' posługujący się prostymi algorytmami.*

- **Kompletna implementacje reguł gry**

*Wszystkie istniejące reguły turniejowe zostaną poprawnie zaimplementowane i będą mogły zostać użyte przez użytkownika w aplikacji. W to wchodzi reguły mniej znane jak 'en passant' czy roszada.*

- **Możliwość gry z innymi graczami oraz komputerem**

*Dostępność dwóch trybów rozgrywki gracz vs gracz i gracz vs komputer.*

- **Możliwość poddania gry**

*W każdym momencie gra może zostać poddana.*

- **Możliwość wyboru strony podczas gry**

*Gracz może określić czy chce rozpocząć daną partię.*

- ◆ **Ocena gry danego gracza**

*Dodanie tej funkcji skutkowałoby koniecznością prowadzenia specjalnych statystyk dotyczących poszczególnych etapów gry danego gracza oraz mechanizmów analizujących zgromadzone dane. Stworzenie tego typu systemu punktującego wykracza znacznie poza zamysł naszego projektu.*

## ❖ Łamigłówki/Puzzle szachowe

*Do 10 łamigłówek szachowych w postaci 'znajdź najlepszy ruch'.*

## ❖ Wybranie poziomu trudności

*Nasz 'komputer' w zamyśle będzie kierowany przez proste algorytmy które nie będą mogły zostać podzielone na poszczególne poziomy trudności. Jedynym rozwiązaniem byłoby napisanie bardziej zaawansowanego systemu, jednak wtedy spełnienie wymagania dotyczącego minimalnego czasu odpowiedzi komputera staje się niemożliwe przy obecnych zasobach czasowych. Również w naszych wymaganiach sprecyzowaliśmy prostotę systemu komputerowego.*