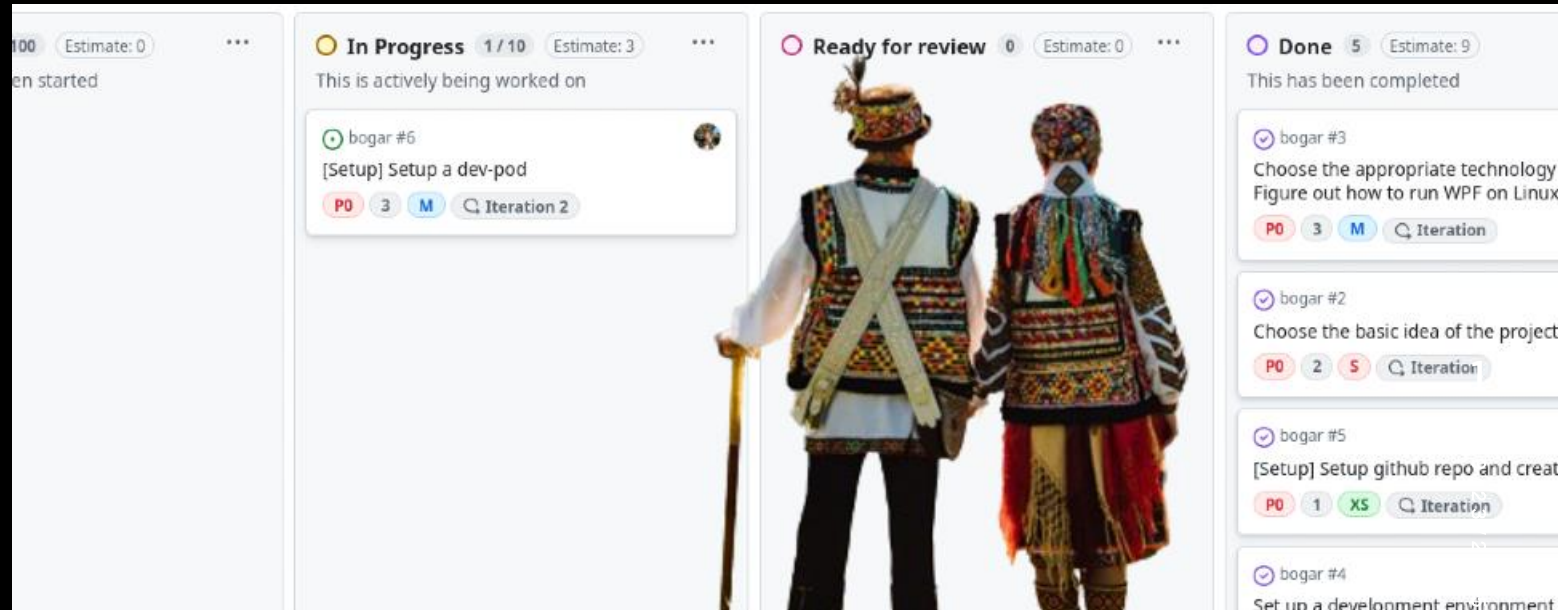


# BOGAR



# ІДЕЯ ПРОЕКТУ:

Шахоподібна гра, орієнтована на змагання між написаними користувачами ботів(програм), які грають за них.

Бот взаємодіє з грою через stdin/stdout, що робить його мовнонезалежним: можна писати як на Python чи C, так і на Haskell/F#.

У самій програмі можна організовувати турніри між користувачами.

# АНАЛІЗ РИНКУ



#### Переваги:

- Підтримка понад 25 мов програмування
- Різноманітні ігрові середовища
- Можливість індивідуальних та командних змагань
- Всесвітня платформа, доступна для всіх без обмежень
- Є форум учасників для обговорень

#### Недоліки:

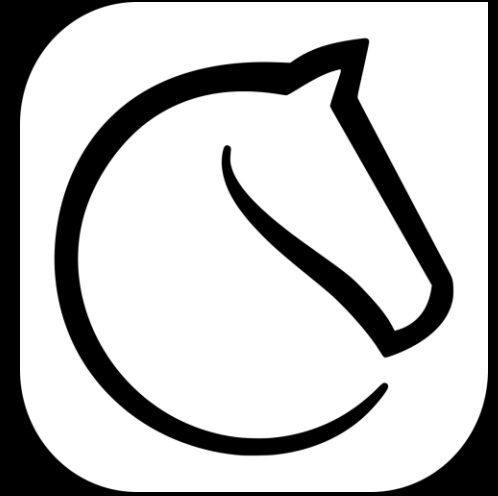
- Тільки онлайн, немає офлайн-режиму
- Більшість ігор або занадто складні в алгоритмічному плані(тобто орієнтовані на тренування ШІ), або занадто прості: без простору для змагань.
- Не дуже зручний вбудований редактор коду

#### Переваги:

- Безкоштовна та відкрита платформа
- Можливість грати онлайн проти людей або ботів
- Підтримка підключення власних ботів через API
- Велика спільнота та активний форум для обговорень та турнірів

#### Недоліки:

- Не підтримує турніри між ботами\*



#### Переваги:

- Орієнтовано на командні змагання
- Проводяться навчання перед змаганнями
- Орієнтована на RTS
- Можливість отримання кредитів MIT та бонусів

#### Недоліки:

- Орієнтована на студентів MIT
- Не всі мови програмування підтримуються. (Java, Python)
- Турніри проводяться заздалегідь визначеним розкладом
- Лише онлайн формат
- Алгоритмічно складні ігри, орієнтовані на ШІ

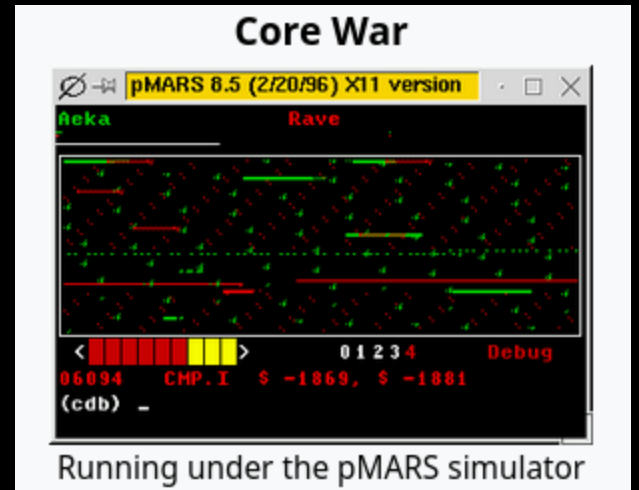


### Переваги:

- Орієнтована на алгоритмічне рішення
- Велика міжнародна спільнота

### Недоліки:

- Власна ASM-подібна мова програмування *Redcode*

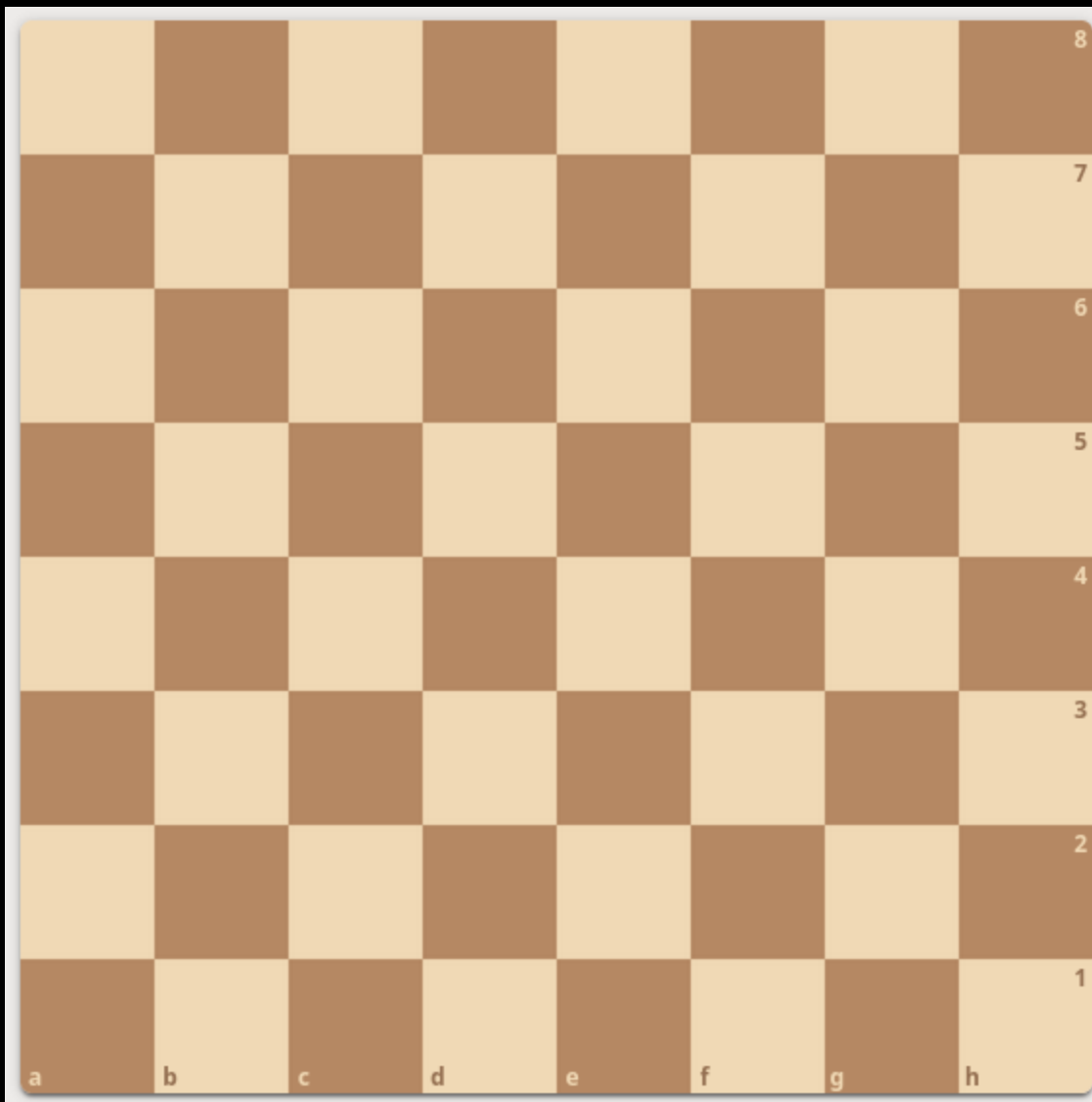


	CodinGame	Lichess	Battlecode	CoreWar
<b>Гра</b>	Широкий вибір	Шахи/подібні	RTS	Алгоритмічна гра
<b>Широкий вибір мов програмування</b>	+	+	-	-
<b>Інтерфейс для ботів</b>	Код виконується на стороні сервера	API	Код виконується на стороні сервера	Програма виконує переданий код
<b>Доступність</b>	+	+	-	+
<b>Можливість проводити змагання</b>	+	-	+	+
<b>Освітньо-орієнтована</b>	+	-	++	-
<b>Орієнтована не на ШІ-рішення</b>	+/-	+	-	+



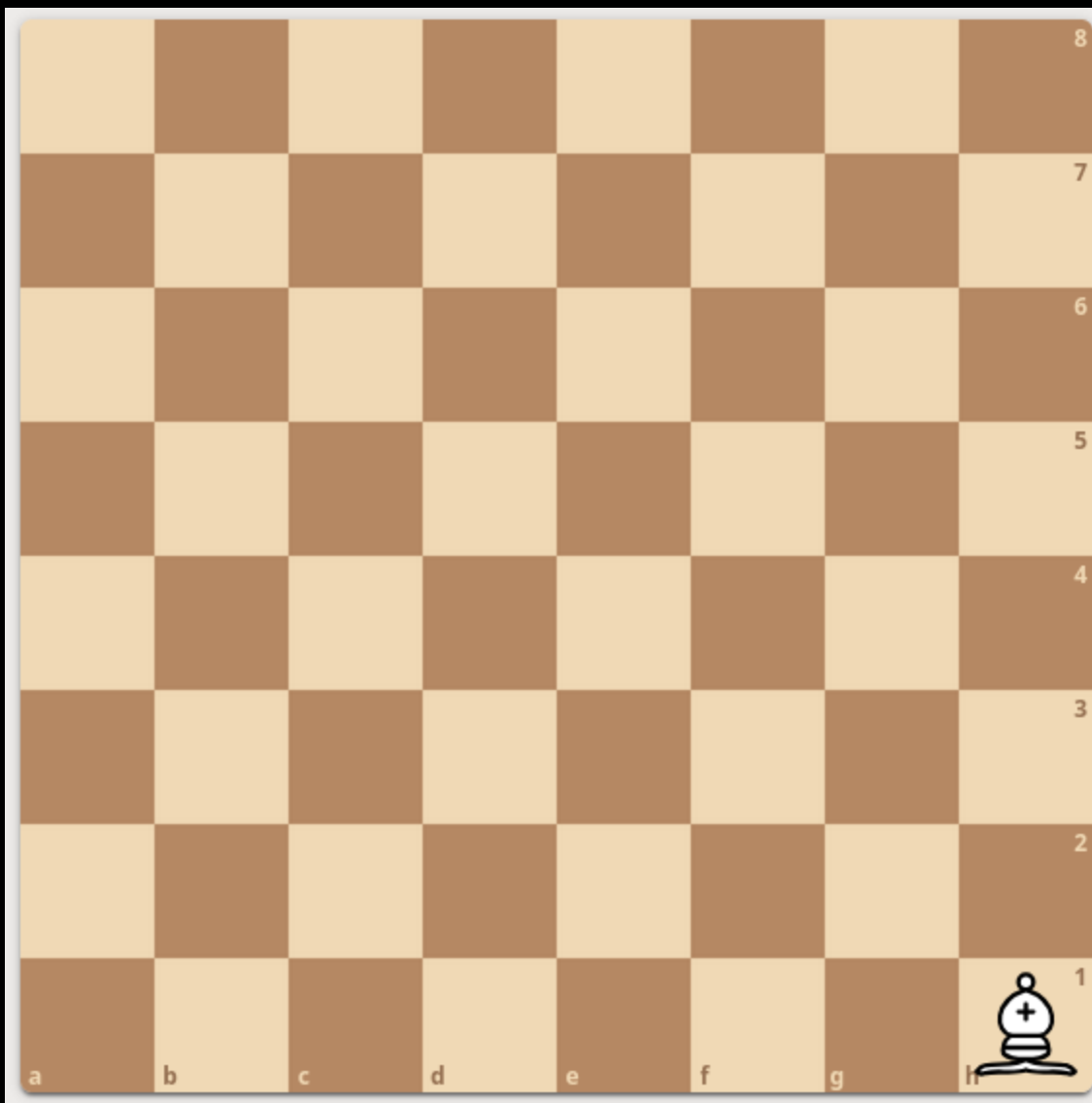
# ПРАВИЛА

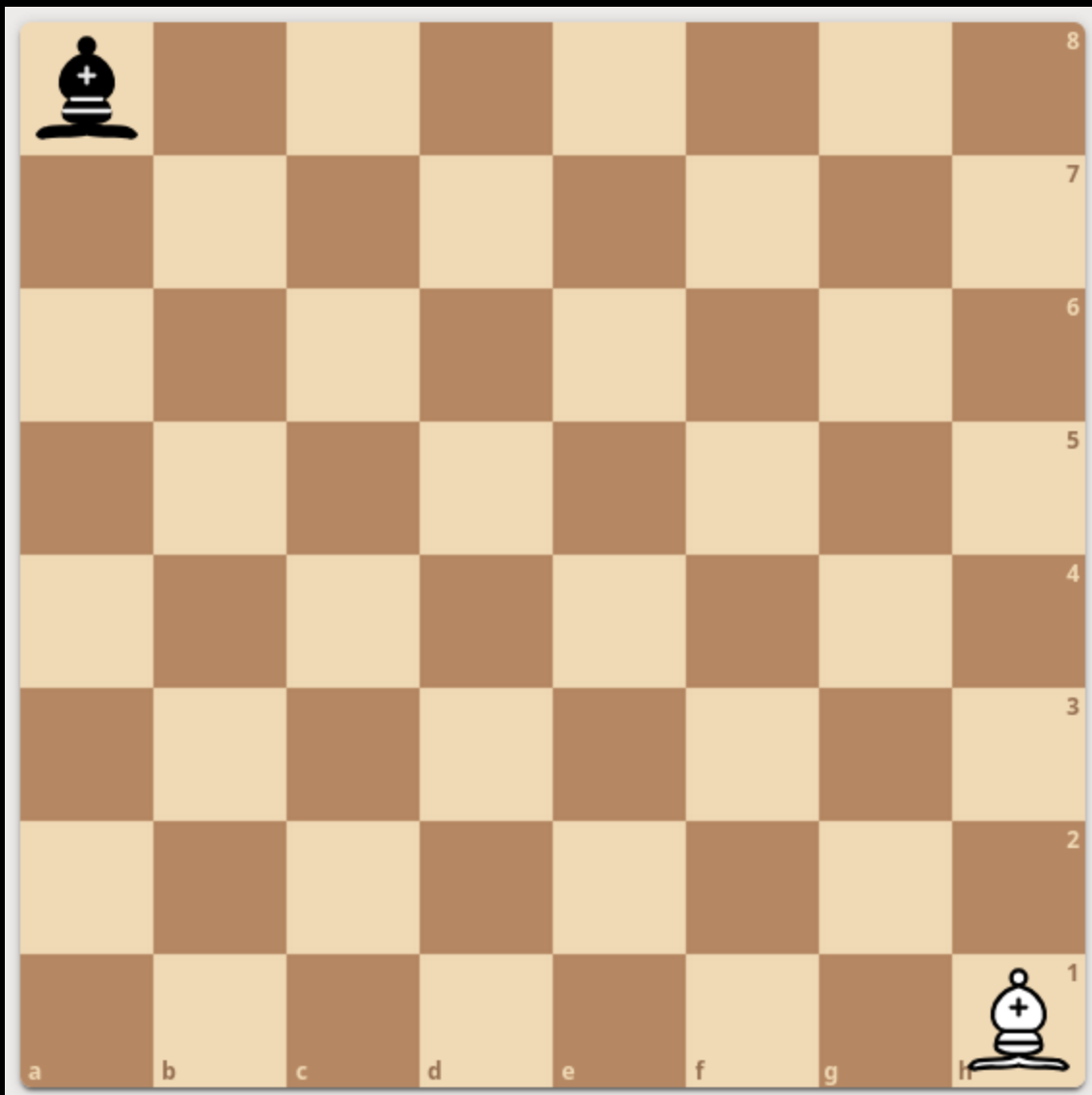
Гра починається з порожньої дошки.  
У обоих гравців у запасі по стандартному набору шахових фігур.



# ПРАВИЛА

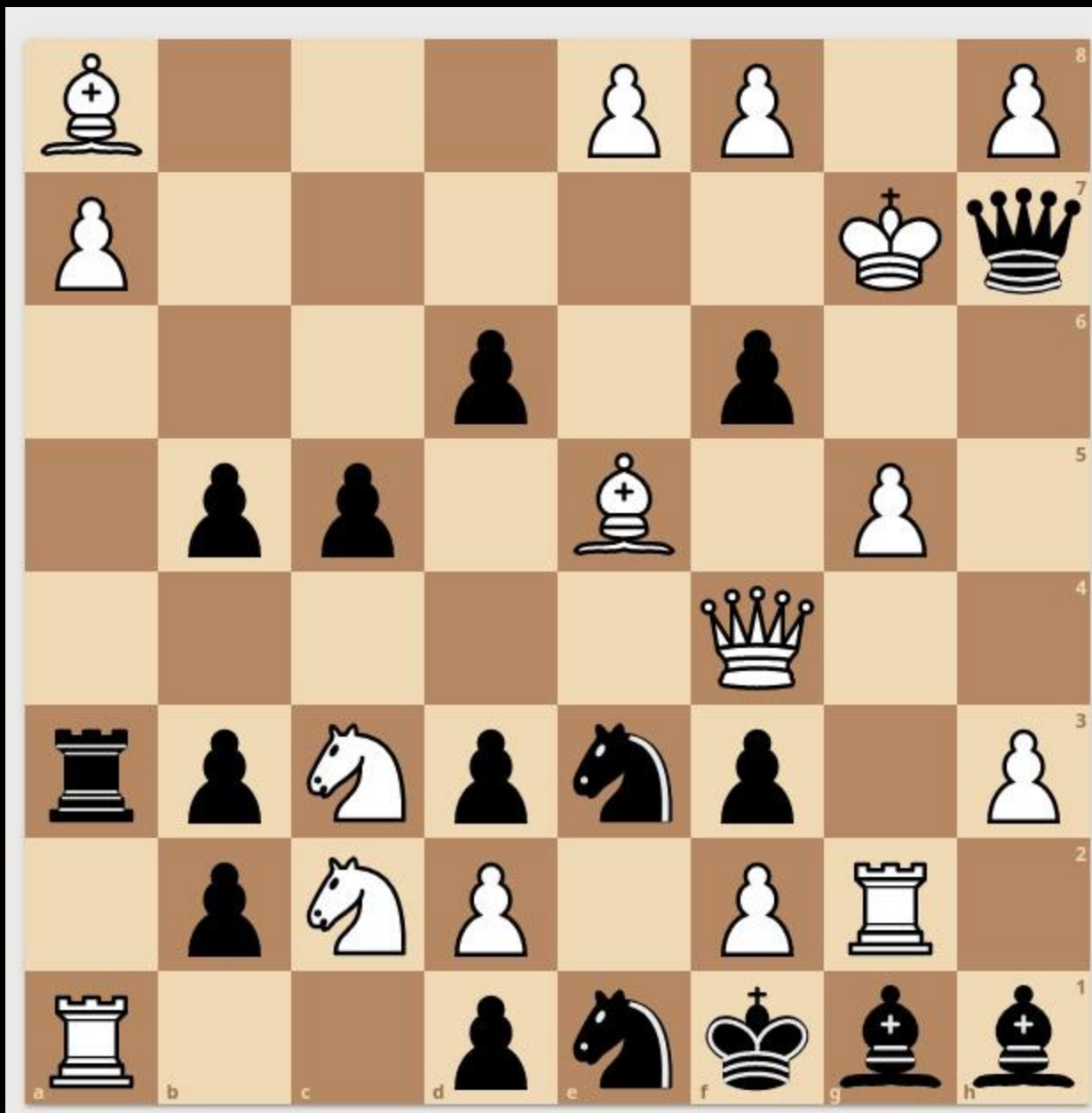
Перший хід - за білими. Вони ставлять довільну доступну фігуру на довільну порожню клітину.





# ПРАВИЛА

Другий хід - чорних, за аналогічними правилами.



Гра завершується, коли на дошці розміщені усі фігури. Після цього йде підрахунок очок кожного гравця. Якщо довільна фігура противника під атакою довільної фігури гравця, то цей гравець отримує певну кількість очок:

Pawn – 1 point  
Bishop/Knight -- 3 points  
Rook – 5 points  
Queen – 8 points  
King – 9 points

# ТЕХНОЛОГІЇ

- WPF
- Entity Framework Core + LINQ
- SQLite
- Windows Job Objects
- WebSocket



# КОРИСТУВАЧІ

Загалом, якщо брати офлайн частину програми, то чіткого ділення на типи користувачів немає. Проте, у будь який момент користувач може запустити сервер на локальному ІР: тоді він автоматично стає адміністратором, який може:

- Створювати/керувати турніром
- Починати матчі між під'єднаними користувачами
- Вилучати гравця з лоббі
- Переглядати загальні результати матчів
- Керувати базою даних(локальною) з результатами матчів

# КОРИСТУВАЧІ

Відповідно, після запуску сервера довільний користувач може до нього під'єднатись, і тоді він отримує роль "учасника".

Учасник лише вказує, з яким з локальних ботів він під'єднується до сервера.

Він може:

- Переглядати результати своїх матчів
- Слідкувати за перебігом матчу за участю власного бота

ЩИРО ДЄКУЄМО ЗА УВАГУ ДЄКУЄМО ДЄКУЄМО

