# Sprawozdanie z Projektu

# Narzędzia Pracy Grupowej

1. **Nazwa aplikacji:**

**TIC TOE GAME**

1. **Opis funkcjonalności aplikacji:**

Aplikacja stworzona w ramach projektu zaliczeniowego to interaktywna gra planszowa, umożliwiająca rozgrywkę w trybie PvP (gracz kontra gracz) oraz PvE (gracz kontra komputer). Głównym celem projektu było opracowanie samodzielnej, czytelnej i w pełni funkcjonalnej aplikacji, która umożliwia użytkownikom intuicyjną grę w środowisku graficznym stworzonym od podstaw.

Po uruchomieniu aplikacji użytkownik zostaje przeniesiony na ekran startowy, na którym może wybrać tryb gry oraz jeden z dostępnych typów plansz. Interfejs jest przyjazny, przejrzysty i został przygotowany z użyciem autorskich grafik, które odpowiadają za estetykę i spójność wizualną całego systemu.

W trybie **PvP** dwóch użytkowników może rozgrywać partię lokalnie na tym samym urządzeniu, wykonując naprzemiennie ruchy zgodnie z zasadami gry. W przypadku trybu **PvE**, przeciwnikiem jednego z graczy jest komputerowy przeciwnik, którego logika została zaimplementowana za pomocą prostych algorytmów AI. Sztuczna inteligencja analizuje sytuację na planszy i wykonuje ruchy, które w zależności od poziomu zaawansowania mogą symulować mniej lub bardziej „inteligentne” zachowanie.

**Mechanika gry obejmuje:**

* **obsługę ruchów graczy,**
* **sprawdzanie poprawności wykonanych ruchów,**
* **wykrywanie końca gry (zwycięstwo jednego z graczy lub remis),**
* **resetowanie planszy i możliwość powrotu do menu głównego po zakończonej partii.**

Użytkownik ma również możliwość wyboru różnych układów plansz, co wpływa na przebieg rozgrywki i dodaje grze różnorodności.

1. **Kod źródłowy:**

https://github.com/Gajewskii/projekt-zaliczenie-npg

1. **Technologie użyte w projekcie:**

**Języki programowania:** Python

**Frameworki:** Flask (do tworzenia aplikacji webowej)

**Biblioteki:**

SQLAlchemy (do obsługi bazy danych)

Jinja2 (do szablonów HTML)

Werkzeug (do obsługi żądań HTTP)

1. **Narzędzia:**

 **GitHub** – wersjonowanie kodu, repozytorium, system zgłoszeń, Wiki

 **Git** – lokalna kontrola wersji, gałęzie, commity

 **Discord** – codzienna komunikacja, szybkie zebrania

 **Google Docs** – współtworzenie dokumentacji i backlogu

 **Trello** – zarządzanie zadaniami (opcjonalnie, jeśli było używane)

1. **Dokumentacja projektu:**

Dokumentacja techniczna oraz instrukcje instalacji i uruchomienia aplikacji znajdują się w pliku README.md w głównym katalogu repozytorium. Dodatkowe informacje dotyczące struktury projektu i używanych technologii można znaleźć w folderze docs.

1. **Organizajcja pracy:**

W projekcie zastosowano metodykę Scrum, z podziałem prac na sprinty oraz regularną oceną postępu. Na początku ustalono wymagania funkcjonalne oraz utworzono wstępny backlog produktu, który następnie był rozszerzany i porządkowany w trakcie pracy.

**Sprint 1 – Planowanie i podstawy projektu**

**Cel: Zaplanowanie funkcji, przygotowanie backlogu, rozpoczęcie implementacji szkieletu aplikacji**

**Zadania:**

* + **Utworzenie repozytorium GitHub**
  + **Stworzenie struktury katalogów i plików**
  + **Określenie głównych funkcjonalności i technologii**
  + **Przydzielenie zadań dla każdego członka**

**Przydziały:**

* + **Z. Sygiet – start aplikacji, ekran powitalny**
  + **M. Smoliński – projekt trybu gry PvP**
  + **J. Taborowski – projekt logiki AI**
  + **D. Matras, P. Gajewski – design graficzny, dokumentacja**

**Czas trwania: 1 tydzień**

**Zebrania: 1 spotkania zespołowe (online)**

**Sprint 2 – Implementacja funkcji gry**

**Cel:** Implementacja trybów gry PvP i PvE, rozpoczęcie integracji z grafiką

**Zadania:**

* + Implementacja mechaniki gry na planszy
  + Logika SI
  + Integracja interfejsu z grafiką
  + Weryfikacja backlogu

**Przydziały:**

* + M. Smoliński – PvP / PvE na różnych planszach
  + J. Taborowski – SI przeciwnika
  + D. Matras, P. Gajewski – dalsze elementy graficzne, edycja backlogu

**Czas trwania:** 1,5 tygodnia

**Zebrania:** codzienne statusy na Discordzie + 1 spotkanie podsumowujące

**Sprint 3 – Testy, poprawki i dokumentacja**

**Cel:** Finalizacja kodu, testowanie działania, przygotowanie dokumentacji końcowej

**Zadania:**

* + Testy techniczne i użytkowe
  + Poprawki w UI i logice
  + Uzupełnienie README i dokumentacji
  + Ocena wkładu zespołu

**Przydziały:**

* + Wszyscy – testowanie
  + Z. Sygiet – podsumowanie prac i organizacja sprawozdania
  + D. Matras, P. Gajewski – końcowa dokumentacja, grafika

**Czas trwania:** 1 tydzień

**Zebrania:** spotkanie końcowe + komunikacja bieżąca

1. **Szczegółowy BACKLOG produktu (PBL) – ewolucja, priorytety, porządkowanie – 2 str.**

Chce dac tutaj tableki od każdeg:

Piotr :

Tabelka

Dawid:

Tabelka

.

.

.

Itd.

1. 3. Dla każdego wykonawcy zespołu (każdy przygotowuje dla siebie) – wkład – 2-3 str.:  
   a) Imię, nazwisko, mail – dewelopera  
   b) Zadania zrealizowane (projektowe, programistyczne, organizacyjne, dokumentacyjne) – wymienić  
   c) Raporty – linki (standup, commit) – aktywność udokumentowana narzędziami (kalendarze, tabele, statystyki, wykresy)  
   d) Parametry oceny pracy, np. procentowa ocena wkładu własnego w projekt (tabela)

## Tabela Oceny Wkładu Własnego

|  |  |  |
| --- | --- | --- |
| Aspekt | Parametry | Wkład |
| Role | Wymienić |  |
| Kodowanie | Liczba linii kodu |  |
| Kodowanie | Funkcje (wymienić) |  |
| Repozytorium | Liczba commit-ów |  |
| Repozytorium | Liczba utworzonych gałęzi |  |
| Repozytorium | Gałąź (używana – nazwa) |  |
| Repozytorium | Liczba połączonych gałęzi |  |
| Repozytorium | Liczba dni aktywności GIT |  |
| Dokumentowanie | Liczba standup-ów |  |
| Aktywność | Opisy na Wiki |  |
| Aktywność | Liczba zrealizowanych zadań |  |
| Aktywność | Szacowana liczba godzin |  |
| Aktywność | Ocena procentowego wkładu |  |