



Projekt Zespołowy Systemu Informatycznego

„Test of might”

Top-down gra 2D typu dungeon crawler

Informatyka rok III semestr V

Autorzy projektu - Grupa Developerska P1D1

Maciej Jankowski

Adrian Rudek

Kamil Pawlak

Dawid Kuta

Prowadzący przedmiot:

Dr inż. Artur Pała

Spis treści

1.	OPIS CELU PROJEKTU	3
2.	WYMAGANIA FUNKCJONALNE	4
3.	WYMAGANIA NIEFUNKCJONALNE.....	4
4.	GRAFIKI KONCEPCYJNE	5
5.	OPIS WYBRANEGO ŚRODOWISKA.....	7
6.	USER STORIES	7
7.	NARZĘDZIA PRACY ZESPOŁOWEJ.....	9
8.	PRODUCT BACKLOG.....	11

1. Opis celu projektu

Gra “Test of Might” będzie dwuwymiarową produkcją typu dungeon crawler, utrzymaną w stylistyce pixel art, skupioną na eksploracji, walce i rozwoju postaci. Celem projektu jest stworzenie wciągającej gry akcji z elementami RPG, w której gracz przemierza zróżnicowane poziomy — od mrocznych podziemi, przez zapomniane kopalnie i opustoszałe wioski, aż po cmentarze i zamki pełne potworów.

Projekt zakłada:

- opracowanie mechaniki eksploracji i walki w środowisku pełnym niebezpieczeństw i ukrytych przejść,
- stworzenie systemu rozwoju bohatera, umożliwiającego zdobywanie doświadczenia, ulepszanie ekwipunku i odkrywanie nowych umiejętności,
- zaprojektowanie różnorodnych poziomów i lokacji, stopniowo zwiększających poziom trudności,
- przygotowanie spójnej oprawy graficznej i dźwiękowej w klimacie fantasy,
- zapewnienie graczowi poczucia wyzwania i satysfakcji z pokonywania kolejnych etapów.

Ostatecznym celem projektu jest stworzenie kompletnej, grywalnej wersji gry, która łączy prostotę klasycznych gier 2D z nowoczesnym podejściem do rozgrywki i estetyki.

Gra Test of Might będzie dwuwymiarową grą akcji typu dungeon crawler w stylistyce pixel art, rozgrywaną z perspektywy z góry. Celem gracza będzie eksploracja niebezpiecznych lokacji pełnych potworów, pułapek i ukrytych skarbów, a także rozwój swojej postaci poprzez walkę, zdobywanie doświadczenia i ulepszanie ekwipunku.

Gra będzie składała się z kilku poziomów tematycznych, takich jak podziemia, kopalnia, cmentarz, opustoszała wioska czy zamek, które reprezentować będą kolejne etapy rozwoju postaci i stopniowo zwiększający się poziom trudności. Każda lokacja będzie zawierała unikalnych przeciwników oraz sekrety, których odkrycie nagradzane będzie nowym wyposażeniem lub umiejętnościami.

Podstawą rozgrywki będzie system walki w czasie rzeczywistym oparty na zręczności gracza i taktycznym wykorzystaniu otoczenia. Gracz będzie mógł korzystać z różnych rodzajów broni — od mieczy i toporów po zaklęcia magiczne — a także tworzyć nowe przedmioty z zasobów znalezionych podczas eksploracji, dzięki prostemu systemowi tworzenia (craftingu).

W grze pojawi się również system rozwoju postaci, w którym gracz, zdobywając punkty doświadczenia, będzie mógł zwiększać swoje statystyki, takie jak siła, szybkość ataku czy ilość zdrowia. Poziomy gry będą zróżnicowane nie tylko pod względem wizualnym, ale także mechanicznym — każdy z nich wprowadzi nowe zagrożenia i wyzwania logiczne wymagające sprytu i refleksu.

Całości dopełni nastrojowa oprawa dźwiękowa i muzyczna, budująca klimat tajemnicy i napięcia podczas eksploracji podziemi oraz spokoju po zakończonych walkach. Głównym celem gracza będzie dotarcie do ostatecznego poziomu — zamku, w którym czeka potężny przeciwnik stanowiący próbę ostatecznej siły („Test of Might”). Pokonanie go zakończy grę, symbolizując zwycięstwo bohatera i zdobycie miana prawdziwego wojownika.

2. Wymagania funkcjonalne

1. Menu pozwalające na wybór postaci, stworzenie nowej, usunięcie postaci oraz możliwość dostosowania głośności gry i możliwość wyjścia z gry
2. Możliwość sterowania bohaterem w dwóch płaszczyznach
3. Opcja cofania się do poprzednich poziomów
4. Dodanie opcji zdobywania przedmiotów „wypadających” z pokonanych wrogów
5. Możliwość zwiększania poziomu trudności
6. Zwiększenie poziomu trudności będzie zwiększać szansę na zdobycie lepszych przedmiotów
7. Wrogowie podążają za graczem atakując go w zwarcie lub też wystrzeliwując pociski
8. Pokonani wrogowie upuszczają doświadczenie
9. Możliwość ulepszania postaci poprzez wbijanie poziomów
10. Pasek zdrowia, gdy spadnie do zera kończy grę

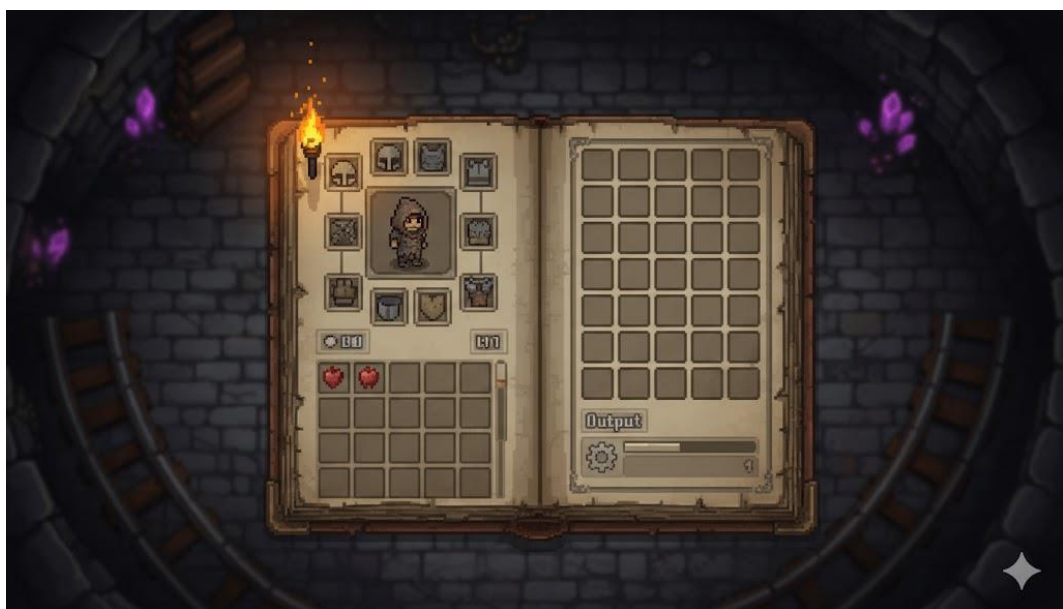
3. Wymagania niefunkcjonalne

1. Gra będzie działać offline
2. Gra będzie działać na komputerach z systemem Windows 10-11

4. Grafiki Konceptyjne



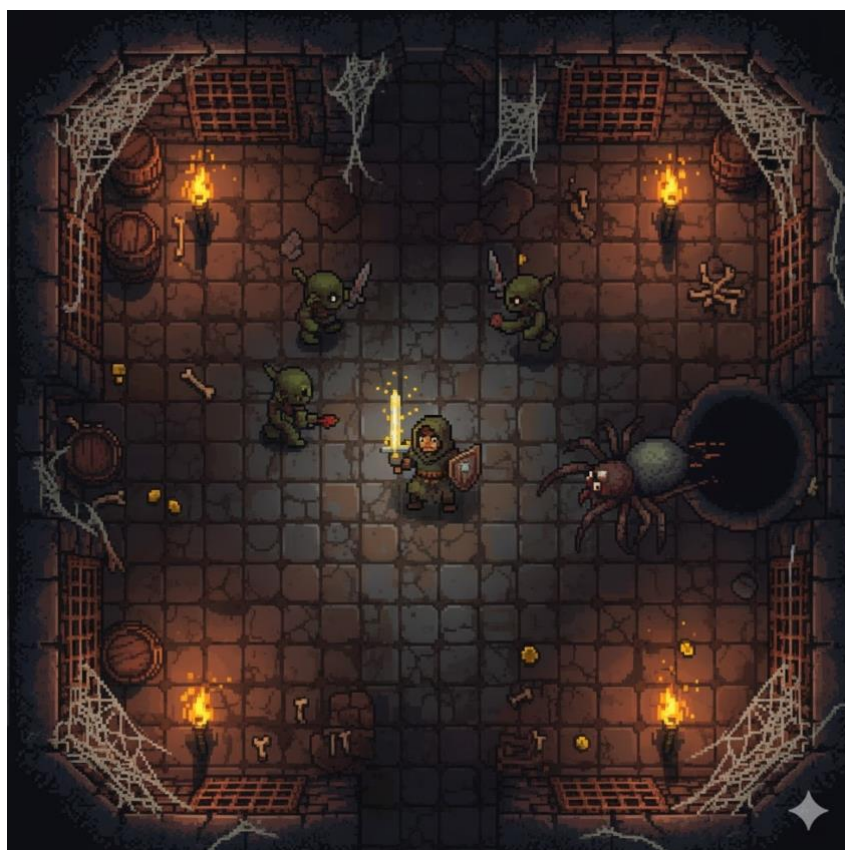
Rysunek 4.1- menu główne



Rysunek 4.2 – ekwipunek w trakcie rozgrywki



Rysunek 4.3 – menu wyboru postaci/zapisu



Rysunek 4.4 – grafika koncepcyjna poziomu

Rys. 4.1. przedstawia koncepcje głównego menu, które zawiera opcje kontynuowania rozgrywki, przejścia menu ustawień oraz opuszczenia gry. Rys. 4.2. przedstawia koncepcje ekwipunku, w którym gracz będzie mógł zarządzać zdobytymi przedmiotami i je ekwipować. Rys. 4.3. przedstawia koncepcje ekranu wyboru postaci, każda z nich jest reprezentacją innego zapisu gry. Rys 4.4. przedstawia koncepcję stylu graficznego przykładowego lochu, w którym gracz może zdobywać dodatkowe poziomy, skarby i ekwipunek.

5. Opis wybranego środowiska

- Visual Studio Code – popularne środowisko programistyczne stworzone przez firmę Microsoft. Pozwalające na tworzenie aplikacji w wielu językach (Python, JavaScript, C++, itd.), środowisko to może zostać użyte do prostych projektów jak i profesjonalnych aplikacji desktopowych, webowych jak i serwerowych.
Cechami tego środowiska są:
 - Duża ilość rozszerzeń możliwych do pobrania pomagających w pracy
 - Wsparcie społeczności
- Godot – jest środowiskiem do tworzenia gier dwuwymiarowych jak i trójwymiarowych przy użyciu wielu języków programowania jak C#, C++ i GDScript. Silnik ten działa na systemach Windows, macOS i Linux, oferując narzędzia do tworzenia scen, animacji, interfejsów jak i logiki gry.
Cechy silnika:
 - Możliwość pisania skryptów w językach C#, C++ oraz GDScript
 - Całkowicie darmowa licencja
 - Duże wsparcie społeczności po przez liczne samouczki, dokumentacje i gotowe pluginy oraz assety
- GitHub – jest internetową platformą mającą zadanie przechowywanie, zarządzanie i współdzielenie kodu źródłowego projektów. Platforma pozwala na śledzenie zmian w kodzie, pozwala na cofanie do wcześniejszych wersji kodu i umożliwia pracę wielu programistą nad jednym kodem.

6. User stories

6.1. Jako “Gracz” chcę móc zapisać stan gry w dowolnym momencie, żeby móc przerwać rozgrywkę i wrócić do niej bez utraty postępu.

Warunki satysfakcji:

- W menu pauzy dostępna jest opcja "Zapisz grę".
- Gracz może wczytać zapisany stan gry z menu głównego.
- Po wczytaniu gry, gracz kontynuuje rozgrywkę od momentu zapisu z zachowaniem wszystkich postępów.

6.2. Jako “Gracz” chcę rozwijać swoją postać, by czuć satysfakcję z robienia postępów

Warunki satysfakcji:

- System rozwoju postaci poprzez zwiększenie statystyk

- System ekwipunku, który można pozyskać z eksploracji
- System umiejętności, pozwalających na zmianę podejścia do walki

6.3. Jako “Gracz” chcę zarządzać swoim ekwipunkiem, aby móc zakładać lepszą broń i pancerz oraz używać przedmiotów.

Warunki satysfakcji:

- Gracz ma dostęp do ekranu ekwipunku.
- Gracz może wyposażać postać w przedmioty z ekwipunku (broń, pancerz, amulety).
- Użycie przedmiotu jednorazowego (np. mikstury) usuwa go z ekwipunku i aktywuje jego efekt.

6.4. Jako “Gracz” chcę dostosować ustawienia gry, w celu zapewnienia sobie jak największego komfortu rozgrywki

Warunki satysfakcji:

- Ustawienia grafiki, dźwięku, wyświetlania, itd.
- Ustawienia dostępności
- Ustawienia dostosowania poziomu rozgrywki
- Ustawienia można zmieniać z widoku menu głównego jak i samej gry

6.5. Jako “Gracz” chcę móc swobodnie eksplorować poziomy, by szukać ukrytych smaczków i skarbów

Warunki satysfakcji:

- Możliwość wyboru poziomu do rozegrania z poziomu menu
- Ukryte skarby i easter egg'i na poziomach zachęcające do eksploracji
- Tracker postępu w szukaniu znajdziek (jakieś osiągnięcie, albo statystyka)
- Nagroda/pochwała za znalezienie znajdzki i wszystkich znajdziek

6.6. Jako “Gracz” chcę atakować potwory przy użyciu różnych rodzajów broni (np. miecze, łuki), aby móc dostosować styl walki do przeciwnika.

Warunki satysfakcji:

- Gracz może wyposażać się w broń białą, bądź dystansową
- Atak bronią białą zadaje obrażenia pobliskim przeciwnikom
- Atak bronią dystansową zadaje obrażenia celom na odległość, ale wymaga trafienia.

6.7. Jako “Gracz” chcę móc wchodzić w interakcję z otoczeniem, bo pozwala mi to na czerpanie większej satysfakcji z grania

Warunki satysfakcji:

- Kapliczki, statuetki, napisy czy fontanny albo inne obiekty które dają buffy
- NPC z którymi można rozmawiać
- Ślady które informują o obecności pułapki
- Pułapki

- Skróty do odblokowania na poziomach

6.8. Jako „Gracz” chcę móc dostosowywać swój styl walki, by przetestować co jest najlepsze dla mnie

Warunki satysfakcji:

- Opcja resetu statystyk
- Opcja zmiany umiejętności dostępnych w walce
- Opcja zmiany umiejętności pasywnych

6.9. Jako „Gracz” chcę wiedzieć więcej o przeciwnikach, w celu lepszego dostosowania swojej strategii walki

Warunki satysfakcji:

- Bestiariusz z informacjami na temat przeciwników
- Wygląd przeciwnika daje informacje jakiego typu ataków będzie używać
- Przeciwnicy telegrafują swoje ataki
- Mapy lub znajdzki dostarczają informacji na temat słabości bossa

6.10. Jako „Gracz” chciałbym mieć dostęp do pokoju testowego, aby sprawdzić różne mechaniki, obrażenia itp.

Warunki satysfakcji:

- Utworzenie poziomu testowego, nie wpływającego na resztę gry.
- Utworzenie „manekinów” do sprawdzania zadawanych obrażeń lub możliwość wybrania dowolnego przeciwnika z gry
- Gracz w tym poziomie może swobodnie wybierać przedmioty, umiejętności z gry

7. Narzędzia pracy zespołowej

7.1.



GitHub Projects to narzędzie zintegrowane z platformą GitHub, służące do planowania i organizacji pracy zespołu programistycznego. Umożliwia tworzenie tablic projektowych, które pozwalają na śledzenie postępów w realizacji zadań oraz przypisywanie ich do konkretnych członków zespołu.

Dzięki temu rozwiązaniu zespół mógł w łatwy sposób koordynować działania, unikać konfliktów w kodzie oraz zapewnić ciągłość prac nad projektem.

7.2.



Discord pełni rolę głównego narzędzia komunikacji zespołowej. Jest to aplikacja umożliwiająca prowadzenie rozmów tekstowych i głosowych w czasie rzeczywistym, a także udostępnianie ekranu i plików.

Dzięki zastosowaniu Discorda zespół mógł sprawnie współpracować niezależnie od miejsca przebywania, co znacząco usprawniło tworzenie gry oraz umożliwiło efektywne zarządzanie czasem i zadaniami.

8. Product backlog

1. Założyć repozytorium na GitHub
2. Konfiguracja projektu na GitHub
3. Założenie kanału na Discord
4. Projekt koncepcji stylu graficznego
5. Projekt map
6. Projekt głównego menu
7. Projekt interfejsu użytkownika w grze
8. Projekt mapy startowej (tutorial)
9. Projekt spritów postaci
10. Implementacja logiki poruszania się
11. Implementacja systemu kolizji
12. Implementacja systemu kamery
13. Implementacja logiki przeciwników
14. Implementacja systemu zdrowia i punktów życia
15. Implementacja systemu punktacji/doświadczenia
16. Implementacja ekwipunku
17. Projekt przedmiotów możliwych do zdobycia
18. Implementacja walki w zwarcu
19. Implementacja systemu zapisu gry
20. Implementacja efektów dźwiękowych
21. Implementacja muzyki w tle
22. Implementacja systemu audio w grze
23. Implementacja ekranu ładowania
24. Implementacja opcji wyboru save'ów
25. Projekt systemu dropu itemów z przeciwników
26. Implementacja logiki odblokowywania poziomów
27. Implementacja systemu dialogów
28. Implementacja quest log (journal – lore, informacje o poziomie, itd.)
29. Implementacja systemu resetu statystyki/umiejętności
30. Implementacja systemu sklepu
31. Projekt przedmiotów specjalnych typu różne mikstury
32. Projekt spritów przeciwników
33. Projekt spritów bossów
34. Implementacja bestiariusza
35. Implementacja systemu walki dystansowej
36. Implementacja prostych cutscenek
37. Implementacja systemu umiejętności
38. Implementacja kilku umiejętności możliwych do odblokowania
39. Dopracowanie animacji, particle itp (poprawa oprawy graficznej)
40. Implementacja meachaniki tooltips
41. Implementacja opcji dostępności
42. Implementacja lore
43. Implementacja ukrytych pokoi i easter egg-ów

- 44. Testowanie poszczególnych poziomów
- 45. Implementacja bug fixów
- 46. Testowanie interfejsu użytkownika
- 47. Publikacja wersji DEMO (DEV)
- 48. Testowanie przed wydaniem pierwszej wersji
- 49. Implementacja łatek optymalizacyjnych
- 50. Wydanie pierwszej wersji gry (RELEASE)