**"PABLO MUSHROOMS"**

**Grupa "Metalowe Płuca":**

Kacper Kruszyński - project manager, główny programista

Filip Gąsior - programista

Paweł Biernat - architekt oprogramowania, analityk

Sebastian Król - tester, analityk

**Aktualna wersja projektu: v1.1**

**Data ukończenia: 22.11.2024 r.**

**Wprowadzenie**

Cel projektu:

Stworzenie gry fabularnej RPG, która oferuje rozrywkę, a także uczy gracza myślenia strategicznego.

Adresowane potrzeby:

* Dostarcza angażującej rozrywki i wyzwań.
* Utrzymuje zainteresowanie dzięki dynamicznym walkom.
* Zaspokaja potrzebę dostosowywania gry do własnych potrzeb i podejmowania decyzji.

Opis aplikacji i funkcjonalności:

* Konsolowa gra RPG, której akcja toczy się w średniowieczu. Gracz ma do przejścia cztery różne biomy, które są pełne przeciwników. Na początku rozgrywki gracz wpisuje swoje imię, wybiera klasę bohatera, którym chce grać oraz biom, na który ma zamiar się udać. Na każdym z wybranych biomów gracz podejmuje walkę z przeciwnikami i bossem czekającym na końcu wybranej ścieżki. Na koniec ostatniego biomu gracz musi się zmierzyć z finalnym bossem, aby przejść grę.

Funkcjonalności:

* Tworzenie i rozwój postaci:
* Wybór klasy (Rycerz, Mag, Łucznik),
* Przypisywanie statystyk (punkty zdrowia, siła, szybkość),
* Zdobywanie poziomów.

* System walki:
* Turowe bitwy z przeciwnikami,
* Ciosy krytyczne,
* Minimalna losowość obrażeń ataków.

* Ekwipunek:
  + Losowe przedmioty o różnych rzadkościach (Common, Uncommon, Rare, Epic, Mythical, Legendary, Divine),
  + Zarządzanie wyposażeniem.

* Eksploracja:
* Wybór biomów (pustynia, bagna, góry i zamek) wpływających na przeciwników.

* Zapis postępów:
  + Automatyczne zapisywanie stanu gry i wyników.

**Proces tworzenia projektu**

Podział pracy w zespole:

Kacper Kruszyński - project manager, główny programista

Gąsior Filip - programista

Paweł Biernat - architekt oprogramowania, analityk

Sebastian Król - tester, analityk

Harmonogram działań:

1. Wybranie rodzaju projektu
2. Zastanowienie się nad potencjalnymi oczekiwaniami graczy
3. Wymyślenie fabuły gry
4. Ustalenie architektury programu
5. Rozpoczęcie pisania kodu
6. Wprowadzenie obsługi błędów
7. Ukończenie początkowej wersji gry
8. Testowanie
9. Rozwinięcie gry, dalsze programowanie
10. Ukończenie obecnej wersji gry
11. Testowanie
12. Napisanie dokumentacji
13. Udostępnienie gry graczom

Kluczowe decyzje projektowe:

* Podział ról i zadań dla każdego członka zespołu. W ten sposób praca była bardzo efektywna,
* Przemyślenie potencjalnych błędów i ich obsługi zawczasu, aby potem mieć więcej czasu na rozwój gry,
* Porzucenie pomysłu utworzenia klasy dla bossa. Stał on się zwykłym, ale zmodyfikowanym przeciwnikiem. Na skutek tej decyzji uniknięto powielania się kodu i zmniejszono jego ilość,
* Umiejętne dysponowanie pamięcią operacyjną poprzez wykorzystywanie istniejących zmiennych zamiast tworzenia nowych.

**Szczegółowy opis aplikacji**

Diagram:

Obraz zawierający tekst, zrzut ekranu, projekt graficzny, Grafika

Opis wygenerowany automatycznie

Funkcje aplikacji:

* Kreator postaci:
* Opis: Funkcja ta tworzy postać, w którą wcieli się gracz. Wprowadza on imię i wybiera klasę dla swojej postaci. Imię będzie wykorzystywane przy wyświetlaniu komunikatów podczas walki.
* Przykład:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* Walka:
* Opis: Funkcja ta umożliwia walkę gracza z przeciwnikami. Na początku losuje ona nazwę przeciwnika (jeśli nie jest bossem) na podstawie wybranego wcześniej biomu i jego statystyki. Potem, na podstawie statystyki szybkości ustala ona, która strona pierwsza rozpocznie atak, a także odejmuje punkty zdrowia jednej stronie na podstawie siły drugiej. Jest pewna szansa na zadanie obrażeń krytycznych przeciwnikowi przez gracza. Po wygranej przez gracza walce funkcja dodaje mu część utraconych punktów zdrowia.
* Przykład:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* Przydzielanie punktów doświadczenia
  + Opis: Funkcja ta nalicza graczowi punkty doświadczenia. Po osiągnięciu wymaganej ilości tych punktów zwiększa ona poziom gracza. W nagrodę za osiągnięcie kolejnego poziomu gracz otrzymuje pewną ilość punktów, dzięki którym może on zwiększyć wybrane statystyki swojej postaci o wybraną ilość punktów.
  + Przykład:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* Losowanie przedmiotów:
  + Opis: Zadaniem tej funkcji jest losowanie przedmiotu, jego rzadkości, typu i ulepszeń statystyk, które da on graczowi. Dzieje się to po pokonaniu co 3 zwykłego przeciwnika lub bossa. Funkcja ta przyjmuje wybór gracza, czy chce on wejść w posiadanie przedmiotu, czy też nie.
  + Przykład:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* Wybór biomu:
  + Opis: Funkcja ta przekazuje na podstawie wybranego biomu informację do funkcji “Walka”, z której tablicy nazw przeciwników powinna ona wylosować nazwę dla przeciwnika.
  + Przykład:

Wybór biomu:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Losowanie nazw przeciwników na podstawie biomów:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* Info o ekwipunku:
  + Opis: Funkcja ta jest wywoływana na żądanie gracza. Po wyświetleniu się stosownego komunikatu, po wciśnięciu klawisza I, gracz może zobaczyć swój ekwipunek, czyli przedmioty, które posiada wraz z ich rzadkościami i ulepszeniem statystyk, które dają.
  + Przykład:



Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* Zagraj jeszcze raz / zakończ grę:
  + Opis: Ta funkcja wywołuje się po śmierci gracza lub przejściu gry. Gracz podejmuje wybór, czy chce zagrać jeszcze raz, czy zakończyć grę. Po wyborze 1 opcji, funkcja rozpoczyna nową grę, a po wyborze 2 opcji - kończy algorytm.
  + Przykład:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

* Zapisywanie:
  + Opis: Ta funkcja zapisuje informacje o graczu i przeciwnikach po każdej zmianie.
  + Przykład: Zapisywane są takie informacje o graczu jak jego imię, klasa postaci, statystyki, poziom i punkty doświadczenia, przedmioty z ich rzadkościami i statystykami oraz wybrany biom.

Interfejs użytkownika:

Typ:

* Konsolowy.

Opis:

* Menu główne:
  + wyświetlanie powitania,
  + wprowadzanie imienia gracza,
  + wybór klasy bohatera.
* Walka:
  + komunikaty o atakach, obrażeniach, statystykach przeciwnika i gracza.
* Eksploracja:
  + Komendy konsolowe do wyboru biomów.

* Ekwipunek:
  + Opcje przypisywania przedmiotów za pomocą klawiszy (np. "Y" - użyj, "X" - odrzuć).

* Zapisywanie gry:
  + Automatyczne informacje o zapisanych danych.

**Problemy i wyzwania:**

Napotkany został problem nieprzemyślanego pisania kodu, przez co potem wyraźnie dało się odczuć trudności w jego późniejszej modyfikacji. Utrudniona została także większa rozbudowa gry.

Rozwiązaniem powyższego problemu okazało się wstrzymanie pracy nad rozbudową projektu i skupienie się na uproszczeniu kodu, aby funkcje w nim zawarte były przejrzyste, łatwiejsze do zrozumienia i miały sens.

Czasami wyzwaniem była nieobecność członków zespołu w poszczególnych dniach, przez co komunikacja była utrudniona, na szczęście nie miało to znaczącego wpływu na powstawanie gry.

Trzeba było podjąć kompromis odnośnie podziału obowiązków, ponieważ niekażdy dobrze sobie radził z powierzonym mu zadaniem.

**Wnioski i dalszy rozwój**

Wnioski:

* Warto upraszczać kod, ponieważ im prostszy będzie kod, tym bardziej czytelny,
* Udało się napisać prosty i czytelny kod oraz stworzyć interesującą i podatną na rozbudowę fabułę,
* Udało się wprowadzić automatyczny zapis gry,
* W zespole panowała dobra atmosfera, a współpraca była na wysokim poziomie, co przyczyniło się do sprawnego stworzenia gry.

Przyszłe plany/sugestie:

* Zbalansowanie progresji gracza, m.in. przez dodanie różnych poziomów trudności,
* Zbalansowanie zwiększania statystyk gracza przez przedmioty oraz dodanie dodatkowych umiejętności pasywnych dla rzadszych broni i broni z bossów,
* Dodanie waluty i sklepu, w którym za zgromadzone pieniądze można będzie można nabywać przedmioty

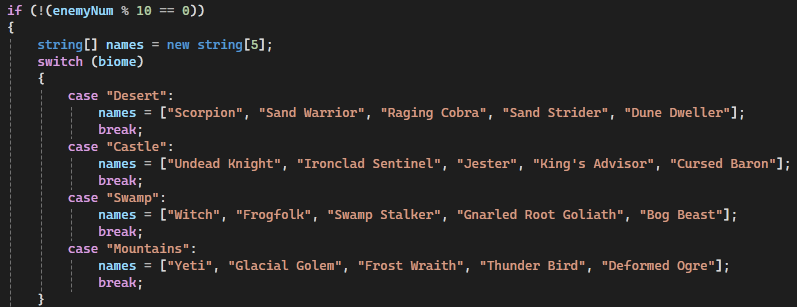
**Załączniki**

Fragment kodu i komentarzy dla walki z przeciwnikiem:

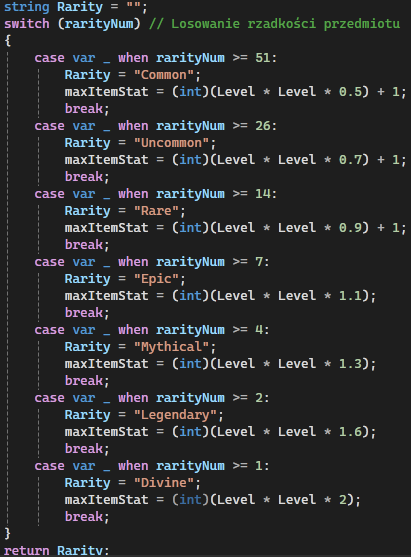
Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

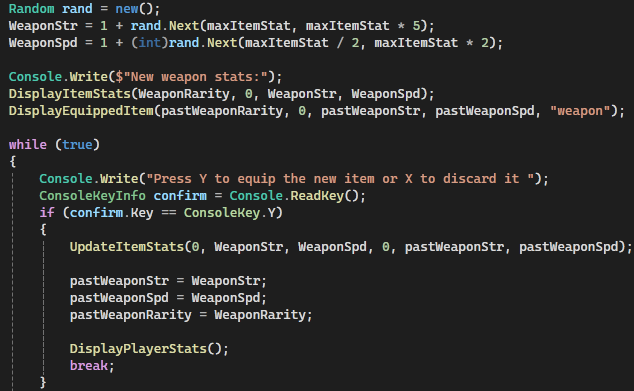
Fragment kodu z tablicami nazw przeciwników:



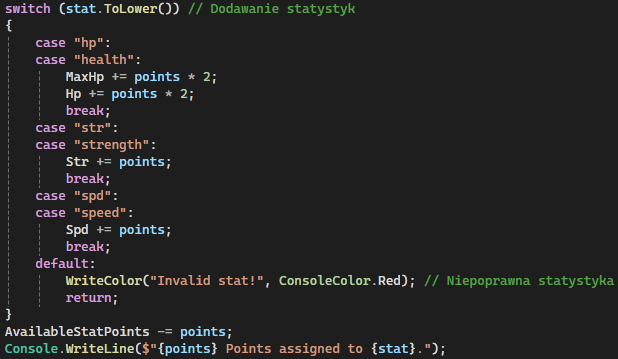
Fragment kodu losującego rzadkość przedmiotów:



Fragment funkcji tworzącej broń i załączającej ją do ekwipunku:



Fragment kodu przypisujący punkty do statystyk po osiągnięciu nowego poziomu:



Przykładowe scenariusze testowe:

Walka z przeciwnikiem:

* Wejście: Rycerz (HP=60, STR=16, SPD=7) vs Scorpion (HP=40, STR=10, SPD=5).
* Wynik: Gracz wygrywa, pozostaje mu 20 HP. Udane.

Level up:

* Wejście: Gracz zdobywa 50 EXP na poziomie 1.
* Wynik: Awans na poziom 2, przyznano 5 punktów statystyk. Udane.

Losowanie ekwipunku:

* Wejście: Broń rzadkości "Legendary" (STR=25, SPD=15).
* Wynik: Statystyki poprawnie przypisane, opcja późniejszej wymiany przedmiotu. Udane.

Wybór biomu:

* Wejście: Wybrano biom "Mountains".
* Wynik: Generowanie przeciwników przypisanych do biomu (np. Yeti). Udane.