

Wydział Informatyki Politechniki Białostockiej Komunikacja Człowiek - Komputer	Data: 02.11.2023r
Projekt 1: Tryb tekstowy Grupa: PS 6 Mariusz Ostaszewski 110913	Prowadzący: dr inż. Eugenia Busłowska

1. Opis projektu

Projektem jest aplikacja okienkowa "Labirynt". Zadaniem użytkownika jest wyjście z labiryntu poprzez rozwiązywanie mini gier oraz ułożenie głównego hasła całej gry. Za każdą poprawnie rozwiązana mini gierkę użytkownik dostanie jedną literkę odpowiadającą. Gracz posiada pewną pulę szans do odgadnięcia głównego hasła. Standardowo ilość ta wynosi 5. Za każdą złą próbę odgadnięcia hasła traci się jedną szansę. Gdy ilość szans wyniesie zero – gracz przegrywa grę.

2. Opis funkcjonalności

Oto opis głównych elementów programu:

1. Inicjalizacja i Zmienne:

Program inicjalizuje obiekt klasy Maze jako generatora labiryntu oraz definiuje stałe, takie jak szerokość i wysokość ekranu (ScreenWidth i ScreenHeight), kąt widzenia (Fov), głębokość (Depth) oraz zmienną isGameRunning, która śledzi stan gry.

2. Menu Główne:

Klasa GameMenu ma metodę RunMainMenu, która wyświetla menu główne gry.

Menu zawiera opcje takie jak "Play", "About" (informacje o grze) i "Exit" (wyjście z gry).

Wybór opcji jest obsługiwany przy użyciu klawiszy strzałek i Enter.

3. Informacje o Grze:

Klasa GameMenu ma metodę DisplayAboutInfo, która wyświetla informacje o grze, takie jak powitanie, a następnie czeka na naciśnięcie dowolnego klawisza, aby wrócić do menu głównego.

4. Rozpoczęcie Gry:

Po wybraniu opcji "Play" w menu głównym, program uruchamia grę przy użyciu metody StartGame.

W grze gracz przemieszcza się w trójwymiarowym labiryncie.

Wykorzystywana jest technika raycastingu do renderowania świata gry na ekranie konsoli.

5. Pauza i Menu Pauzy:

Klasa GameMenu ma metodę PauseGame, która pozwala na pauzowanie gry i wyświetla menu pauzy.

Gracz ma możliwość kontynuowania gry lub wyjścia.

6. Menu Game Over i Wygrana:

Klasa GameMenu posiada metody GameOver i GameWin, które wyświetlają menu informujące o końcu gry lub wygranej.

W przypadku przegranej, gracz może rozpocząć nową grę lub wyjść.

Po wygranej, gracz ma te same opcje.

7. Raycasting i Renderowanie Świata:

Główny silnik gry obsługuje raycasting do renderowania trójwymiarowego świata gry.

Świat jest renderowany na podstawie pozycji gracza i jego kierunku widzenia.

Gracz widzi trójwymiarowy widok świata w konsoli z efektem perspektywy.

8. Mapa i Pozycja Gracza:

Na ekranie wyświetlana jest mapa labiryntu oraz pozycja gracza (oznaczona jako "P").

Na górnym pasku ekranu wyświetlane są informacje o pozycji gracza, kierunku, a także licznik klatek na sekundę (FPS).

9. Restart i Wyjście z Gry:

Po przegranej lub wygranej, gracz ma możliwość rozpoczęcia nowej gry lub opuszczenia programu.

3. Szczególnie interesujące zagadnienia projektowe

Projekt został napisany w języku C#. Wykorzystałem do tego celu środowisko Visual Studio 2022. Szczególną uwagę zwraca wygląd aplikacji. Stawia wrażenie 3-wymiarowej choć jest 2D. Użyta została do tego metoda Raycastingu czyli techniki do renderowania trójwymiarowych scen na dwuwymiarowym ekranie poprzez symulację rzucania promieni (lub "rayów") z kamery gracza i określanie, co te promienie trafiają na swojej drodze. W tym kodzie używany jest raycasting do renderowania perspektywicznego widoku trójwymiarowego labiryntu w konsoli.

Główna pętla gry znajduje się w metodzie StartGame, gdzie generowany jest widok gry na podstawie raycastingu. Poniżej jest omówienie głównych kroków wykonywanych w tej metodzie:

Inicjalizacja zmiennych: Wczytywane są pozycje gracza, ustawienia gry, takie jak szerokość i wysokość ekranu (ScreenWidth i ScreenHeight), kąt widzenia (Fov), i inne zmienne pomocnicze.

Pętla gry: Główna pętla gry to nieskończona pętla, w której raycasting jest używany do renderowania trójwymiarowego widoku. W każdej iteracji pętli wykonywane są następujące kroki:

- a) Obliczanie kąta promienia: Dla każdego piksela na ekranie obliczany jest kąt promienia w zależności od pozycji gracza i kierunku patrzenia.
- b) Rzucanie promienia: Rozpoczyna się rzucanie promienia z pozycji gracza i kierunku patrzenia. Promień ten jest poruszany krok po kroku, aż do trafienia na ścianę lub osiągnięcia maksymalnej odległości (Depth).
- c) Sprawdzanie kolizji: Dla każdego kroku promienia sprawdzane jest, czy promień trafia na ścianę labiryntu. Jeśli tak, to obliczana jest odległość do tej ściany i określane jest, czy jest to granica między dwiema ścianami (czy promień jest "związany").
- d) Renderowanie klatki: Na podstawie odległości do ściany i ewentualnej granicy obliczane jest, jaki znak (np. ściana, granica, puste pole) zostanie wyświetlony na ekranie w danym pikselu. W ten sposób renderowany jest widok 3D na ekranie konsoli.
- e) Aktualizacja pozostałych elementów gry: W pętli aktualizowane są także inne elementy gry, takie jak pozycja gracza, interakcje z labiryntem, statystyki itp.

Wyświetlanie wyników: Na końcu każdej iteracji pętli, wynikowa klatka gry jest wyświetlana na ekranie konsoli za pomocą metody `SetCursorPosition` i `Write`.

4. Instrukcja instalacji

Aby zainstalować program trzeba wykonać następujące kroki:

Przygotuj środowisko pracy: Upewnij się, że masz zainstalowane środowisko programistyczne i narzędzia niezbędne do kompilacji i uruchomienia programu. W przypadku programu, który dostarczyłeś, wydaje się, że jest to aplikacja konsolowa napisana w języku C# i zależy od narzędzi związanych z platformą .NET Framework lub .NET Core. Upewnij się, że masz środowisko .NET zainstalowane na swoim systemie.

Pobierz kod Źródłowy: Upewnij się, że masz dostęp do kodu Źródłowego programu. Możesz go przekopiować i zapisać na swoim komputerze w odpowiednim folderze.

Kompilacja kodu Źródłowego: Otwórz wiersz poleceń (Command Prompt) lub terminal i przejdź do katalogu, w którym zapisano kod Źródłowy programu.

Jeśli korzystasz z .NET Framework (starsze wersje), możesz skorzystać z narzędzia `csc` (C# Compiler) do kompilacji. Wpisz następującą komendę:

```
bashCopy code
```

```
csc /out:NazwaTwojejAplikacji.exe TwójKodZrodlowy.cs
```

W miejsce `NazwaTwojejAplikacji.exe` podaj nazwę, jaką chcesz nadać plikowi wynikowemu, a w miejsce `TwójKodZrodlowy.cs` podaj nazwę pliku z kodem Źródłowym.

Jeśli korzystasz z .NET Core (nowsze wersje), użyj narzędzia dotnet do kompilacji. Wykonaj następującą komendę:

Copy code

dotnet build

Upewnij się, że jesteś w katalogu zawierającym plik csproj.

Uruchom program: Po kompilacji, uruchom program z wiersza poleceń lub terminala, wpisując:

Copy code

./NazwaTwojejAplikacji.exe

Gdzie NazwaTwojejAplikacji.exe to nazwa pliku wynikowego po kompilacji.

To powinno wystarczyć do zainstalowania i uruchomienia programu na twoim komputerze. Pamiętaj, że jeśli kod źródłowy zawiera zależności lub inne instrukcje instalacyjne, to konieczne może być ich spełnienie przed uruchomieniem programu.

5. Instrukcja użytkownika

W grze poruszasz się za pomocą AWSO:

- Wciśnij klawisz "W", aby poruszać się do przodu.
- Wciśnij klawisz "S", aby poruszać się do tyłu.
- Wciśnij klawisz "A", aby obrócić się w lewo.
- Wciśnij klawisz "D", aby obrócić się w prawo.

Zbieranie liter: Na swojej drodze możesz natknąć się na litery oznaczone jako "?" na mapie. Jeśli jesteś obok takiej litery, możesz spróbować ją zebrać, wciśnięciem klawisza "E". Wtedy gra przekieruje Cię do mini gry, której przejście doda literkę do hasła głównego.

Rozgrywka słowna: W grze słownej, używając klawisza "R", możesz rozpocząć grę, w której musisz spróbować ułożyć słowo lub zdanie z zebranych liter. Ułożenie słowa poprawnie zapewnia zwycięstwo.

Pauza w grze: Możesz w dowolnym momencie w trakcie rozgrywki wciśnąć klawisz "Escape", aby wstrzymać grę. Będziesz miał możliwość wznowienia gry lub wyjścia do menu głównego.

Menu główne: Po uruchomieniu gry, pojawi się menu główne z opcjami "Play", "About" i "Exit". Użyj klawiszy strzałek, aby wybrać opcję, a następnie wciśnij "Enter" w celu wyboru.

- Opcja "About": Opcja "About" pozwala na wyświetlenie informacji o grze. Możesz wrócić do menu głównego, wciskając dowolny klawisz.
- Opcja "Exit": Opcja "Exit" pozwala na wyjście z gry. Jeśli jesteś pewien swojej decyzji, wciśnij "Enter", aby zakończyć grę.

7. Wnioski

Podsumowując, gra zawiera elementy tradycyjnej rozgrywki w labiryncie, ale dodaje do niej ciekawy element gry słownej. Pozwala to na różnorodność w rozgrywce i zachęca gracza do eksploracji i kreatywnego myślenia. Dodatkowo, możliwość wstrzymania gry i powrotu do menu głównego czyni ją bardziej przyjazną dla gracza. Na dodatek proste interfejsy tekstowe, mogą uczynić aplikację bardziej przystępnymi dla użytkowników.

8. Samoocena

Myślę, że gra posiada wygodny interfejs użytkownika, który pozwala graczowi w prosty sposób poruszać się i kontrolować swoją postać w trójwymiarowym środowisku. Projektowanie interfejsu gry uwzględnia różne akcje i zachowania gracza, takie jak poruszanie się, obracanie kamerą czy wejście do menu pauzy. To proste, intuicyjne interfejsy użytkownika czynią rozgrywkę bardziej przystępną, nawet dla graczy bez wcześniejszego doświadczenia w grach komputerowych. Dzięki interfejsowi gracze mogą kontrolować swoją postać bez potrzeby znajomości skomplikowanych poleceń, co sprawia, że gra jest łatwiejsza do zrozumienia i bardziej przyjazna dla nowych graczy.