

1. Temat projektu

Celem projektu było odtworzenie pierwszej części z popularnej serii gier automatowych - "Mortal Kombat", wraz z wykorzystaniem nowoczesnych technik programowania w języku C++ oraz metodyki obiektowej. Projekt przyjął żartobliwą nazwę "Patykovy Mordulec".

2. Analiza tematu

Zadanie wymaga sprawnej realizacji wielu aspektów tworzenia gry komputerowej. Dodatkowo praca musi być w miarę możliwości równo podzielona pomiędzy 2 osoby. Najważniejsze zagadnienia:

2.1 Podstawowy silnik gry

Konieczne jest utworzenie silnika pozwalającego na wyświetlanie dwuwymiarowych grafik. Powinien on również wspierać obsługę prostej fizyki, detekcję kolizji oraz dać możliwość implementacji wymaganych nowoczesnych bibliotek języka C++.

2.2 Interfejs i nawigacja

W celu oddania stylu realizacji menu oraz interfejsu oryginalnego wzorca, należy pozostawić sterowanie nawigacją klawiaturze. Menu wyboru postaci oraz ekrany kończące pozostać mają proste i przejrzyste.

2.3 Kluczowe aspekty rozgrywki

Odwzorowane mają zostać najważniejsze aspekty gry typu bijatyka, takie jak poruszanie, skakanie, wyprowadzanie różnorodnych ciosów, blokowanie czy uniki. Dodatkowo gracze mają mieć możliwość wyboru postaci, które różnić będą się nie tylko wyglądem ale również zestawem ruchów oraz powiązaniem z nim "stylem walki".

2.4 Cykl trwania meczu

Każdy mecz podzielony zostanie na niezależne rundy. Po rozpoczęciu rozgrywki czas do zakończenia rundy pozostanie ograniczony. Runda trwać będzie do momentu, aż któremuś z graczy skończą się punkty zdrowia, lub do upływu czasu. Mecz kończy się w momencie wygrania przez jednego z graczy 2 rund.

2.5 Elementy GUI

Na ekranie gry poza samymi postaciami znajdować będą się ulokowane w stałych miejscach elementy Graficznego Interfejsu Użytkownika. Będą to między innymi paski, reprezentujące ilość posiadanych punktów zdrowia każdego z graczy oraz licznik czasu rundy. Ich wygląd również wzorowany będzie na oryginalnej inspiracji.



Rys. 1: Finalna wersja głównych elementów GUI

2.6 Treści i zawartość

Poza typowo technicznymi aspektami, projekt zakłada dodanie do napisanej gry podstawowych zawartości. W celu oddania oryginalnej oprawy graficznej rysowane elementy, takie jak grafiki postaci lub areny wyglądem będą przypominały grę retro. Dodatkowo utworzone będą oryginalne postacie, wraz z pasującymi do ich wyglądu zestawami ruchów. Same postacie będą musiały przejść przez proces optymalizacji oraz balansowania, w celu zapewnienia sprawiedliwej i zajmującej rozgrywki.



Rys. 2: Klatki animacji ataku jednej z postaci

2.7 Optymalizacja

Gry gatunku bijatyka mocno opierają się na szybkich reakcjach graczy. Z tego powodu płynna i responsywna rozgrywka jest wyjątkowo istotna. Wymagane jest, aby silnik posostawiał wiele miejsca na optymalizację. Ważne jest unikanie czasochłonnych operacji oraz sprawna identyfikacja punktów krytycznych.

2.8 Biblioteki zewnętrzne

W celu realizacji podstawowych elementów silnika oraz rozgrywki zdecydowano się wykorzystać bibliotekę Simple and Fast Multimedia Library. Pozwala ona na proste wprowadzanie wielu istotnych mechanik, takich jak wczytywanie i wyświetlanie grafik, operacje związane z liczeniem czasu lub detekcję kolizji.

3. Specyfikacja zewnętrzna

3.1 Uruchamianie

Program można uruchomić z linii poleceń, lub klikając dwukrotnie na plik wykonywalny programu.

3.2 Ekran wyboru postaci

Po uruchomieniu, wyświetla się ekran powitalny, który po 5 sekundach przechodzi do ekranu wyboru postaci - menu.



Rys. 3: Ekran wyboru postaci

Gracze mogą wybierać spośród 4 unikalnych postaci, za pomocą klawiszy *WSAD* oraz *IKJL*. Portrety aktualnie wybranych postaci posiadają odpowiednie kolory tła, dla gracza pierwszego czerwone, a dla gracza drugiego - niebieskie. Oprócz tego, po lewej i prawej stronie ekranu wyświetlana jest animacja, w której biorą udział aktualnie wybrane postaci. Aby zakończyć etap wyboru postaci i przejść do rozgrywki, gracz pierwszy powinien wcisnąć i przytrzymać klawisz *TAB* a gracz drugi, klawisz *ENTER*.

3.3 Ekran ładowania

W czasie ładowania się gry (wczytywania z pliku tekstur poszczególnych postaci i ataków) wyświetlany jest odpowiedni ekran, prezentujący postępy. Ponieważ wczytanie gry odbywa się bardzo szybko, na nowoczesnych komputerach ekran ładowania nie zostanie raczej w ogóle zauważony przez użytkownika.

3.4 Rozgrywka

4. Specyfikacja wewnętrzna

5. Testowanie

Program był wielokrotnie uruchamiany i rozgrywany różnymi dostępnymi postaciami. Ponadto, pojedyncze komponenty były testowane jednostkowo. Zwrócono szczególnie uwagę na możliwe błędy

występujące przy wcisnięciu wielu klawiszy na raz, takie jak na przykład kucanie postaci podczas skoku. Wykryte nieprawidłowości zostały skutecznie wyeliminowane, w czym bardzo przydatne było zastosowanie uproszczonej maszyny stanów dla pozycji gracza. # 6. Uwagi i wnioski

6.1 Praca zespołowa i organizacja

Elementem który pozwolił utrzymać wysoką produktywność oraz uniknąć wielu kłopotów podczas późniejszych faz realizacji projektu było przykładowe i konkretne podzielenie się obowiązkami oraz ustalenie wspólnej wizji produktu końcowego w pierwszej fazie pracy zadaniem. Praca została rozłożona w taki sposób, aby obydwie osoby pracujące nad projektem nie przeszkadzały sobie nawzajem. Tworzone komponenty pozostały mało zależne od siebie, a interfejsy kompatybilne. W ten sposób mogły być tworzone, testowane i modyfikowane jednocześnie bez wzajemnego utrudniania pracy. Wyjątkowo przydatne okazało się korzystanie z systemu kontroli wersji GIT. Umożliwił on bezproblemową współpracę nad kodem oraz pomógł lepiej kontrolować historię pisanego kodu.

6.2 Kłopotliwe aspekty produkcji gier

6.3 Nowoczesne biblioteki języka C++