

UNIwersytet Zielonogórski
Wydział Informatyki, Elektrotechniki i Automatyki
Instytut Sterowania i Systemów Informatycznych

Programowanie Gier 3D – Projekt
Prowadzący: Dr inż. Marek Sawerwain

Sprawozdanie z projektu

Wykonał: Przemysław Dragańczuk, Grupa dziekańska: 33-INF-SSI-SA
Data oddania projektu: 23.01.2019

Ocena:

Spis treści

1 Wprowadzenie	1	2.1.2 Zmiana poziomów	2
1.1 Planowane funkcjonalności	1	2.2 Skrypt kontrolujący ruch gracza	2
2 Opis wkładu własnego w realizację projektu	2	3 Spis zastosowanych zasobów	3
2.1 Skrypt zarządzający logiką gry	2	3.1 Tekstury	3
2.1.1 Śledzenie punktów	2	3.2 Skrypt kontrolujący ruch gracza	3
		4 Podsumowanie	3

1 Wprowadzenie

Celem projektu jest stworzenie gry typu “Platformer” w przestrzeni 3D i z widokiem z pierwszej osoby. Sterowanie odbywa się za pomocą myszki i klawiatury. Kamera sterowana jest używając myszki, a postacią sterują klawisze WASD oraz spacja.

1.1 Planowane funkcjonalności

- Poruszanie postacią
- Poruszanie kamerą z perspektywy pierwszej osoby
- Skakanie
- Przechodzenie między poziomami
- Punkty w postaci monet zbieranych przez gracza
- Mierzenie czasu, który upłynął od początku poziomu

2 Opis wkładu własnego w realizację projektu

2.1 Skrypt zarządzający logiką gry

Logiką i stanem gry zarządza skrypt `GameLogic`. Zajmuje się on przechowywaniem takich informacji jak ilość zebranych punktów, obecny poziom czy czas spędzony w obecnym poziomie. Steruje on również wyświetlaniem informacji w interfejsie użytkownika.

2.1.1 Śledzenie punktów

Śledzenie zebranych punktów odbywa się za pomocą listy obiektów reprezentujących punkty. Kolizja gracza z danym obiektem oznaczonym jako "Pickup" dodaje ten obiekt do listy, a następnie wyłącza go. W przypadku śmierci gracza, wszystkie zebrane obiekty są uruchamiane, a lista zostaje opróżniona. W przypadku, gdy gracz dojdzie do końca poziomu, ilość punktów zostaje dodana do zmiennej `oldPoints`, a lista zostaje opróżniona. Jako część interfejsu użytkownika zostaje wyświetlona suma punktów zebranych w poprzednich poziomach oraz ilości obiektów w liście.

2.1.2 Zmiana poziomów

Kolizja gracza z obiektem oznaczonym jako "Finish" powoduje przejście do nowego poziomu. Oznacza to wykonanie kilku czynności. Po pierwsze, następuje sumowanie punktów opisane wyżej. Następnie usuwany jest obiekt reprezentujący obecny poziom, a na jego miejsce instancjonowany i aktywowany jest obiekt reprezentujący następny poziom, który wybierany jest z listy Prefabów. Na końcu resetowany jest czas spędzony w poziomie.

Prefab reprezentujący poziom musi zawierać:

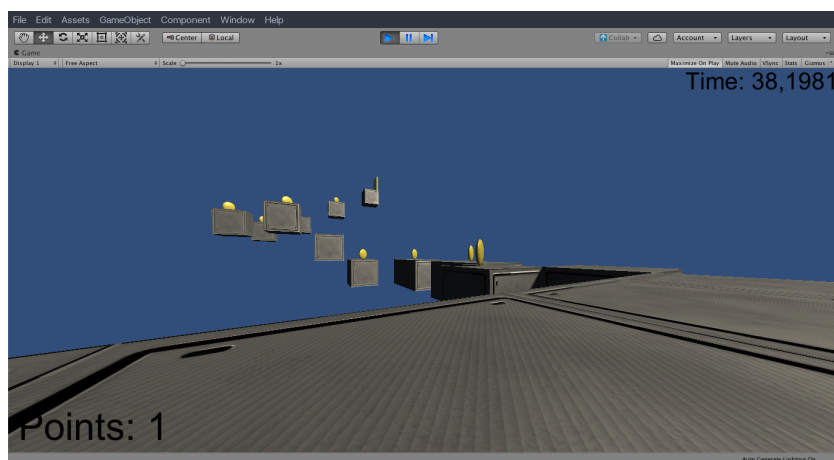
- Znacznik pozycji początkowej, czyli pusty obiekt oznaczony jako "Start"
- Platformy, po których porusza się gracz, oznaczone jako "Terrain"
- Obiekt reprezentujący koniec poziomu, oznaczony jako "Finish"

Dodatkowo powinien zawierać również obiekty reprezentujące punkty, oznaczone jako "Pickup".

Dla wszystkich tych elementów dostępny jest Prefab, zawierający odpowiednie oznaczenia.

2.2 Skrypt kontrolujący ruch gracza

Poruszanie się gracza odbywa się za pomocą zmodyfikowanego skryptu, będącego portem kodu z gry (3.2) Modyfikacje miały na celu usunięcie funkcjonalności nie używanej w tym projekcie, dodanie funkcjonalności takich jak prawidłowe ustawianie pozycji startowej, a także poprawę czytelności kodu.



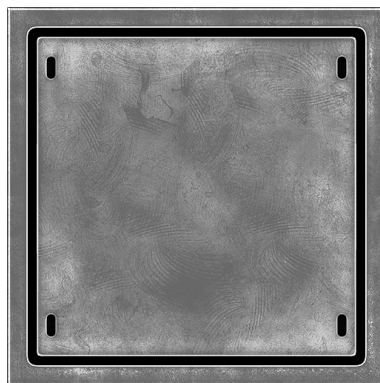
Rysunek 1: Widok z perspektywy postaci

Na początku gry, a także przy przejściu do następnego poziomu, pozycja gracza ustawiana jest w zależności od początkowej pozycji ustalonej dla każdego poziomu jako pusty obiekt. W przypadku, gdy gracz spadnie z mapy, jego pozycja jest również resetowana.

3 Spis zastosowanych zasobów

3.1 Tekstury

W projekcie została użyta tekstura autorstwa użytkownika “Spiney” ze strony <https://opengameart.org/content/metalstone-textures>, a konkretnie tekstura i mapa normalnych “mtl_o” z tej paczki¹. Użyta została jako tekstura platform na mapie.



Rysunek 2: Użyta tekstura

3.2 Skrypt kontrolujący ruch gracza

Skrypt kontrolujący ruch gracza został pobrany z tego repozytorium ². Skrypt ten jest bezpośrednim portem systemu poruszania w grze Quake 3. Oryginalny kod z Quake 3 używa licencji GPL-2.0, zatem jego port również jej używa. Co za tym idzie, zmodyfikowany skrypt używany w tym projekcie także używa tej licencji.

4 Podsumowanie

W trakcie trwania projektu udało się zaimplementować prostą, lecz grywalną grę typu “Platformer”. Dzięki zaimplementowanemu systemowi zmiany poziomu, dodawanie nowych map jest proste, a wbudowany pomiar czasu w połączeniu ze zdobywaniem punktów ma szansę na zainteresowanie społeczności speedrunerskiej.

W projekcie udało się zaimplementować wszystkie planowane funkcjonalności.

¹<https://opengameart.org/content/metalstone-textures>

²<https://github.com/WiggleWizard/quake3-movement-unity3d>