
Wstępna dokumentacja projektu semestralnego *Endless Runner*

Autorzy projektu: Dominika Kawucha, Paweł Kosmala

1. Wprowadzenie

1.1. Cel

Celem niniejszego dokumentu jest przedstawienie i opis architektury oprogramowania dla projektu: *Endless Runner*, tworzonego w ramach zajęć z przedmiotu Programowanie Komponentowe.

1.2. Zakres

Dokumentacja zawiera opis przyjętych rozwiązań w projekcie oraz załączone diagramy UML, opisujące komunikację pomiędzy poszczególnymi elementami i komponentami programu.

2. Prezentacja projektu

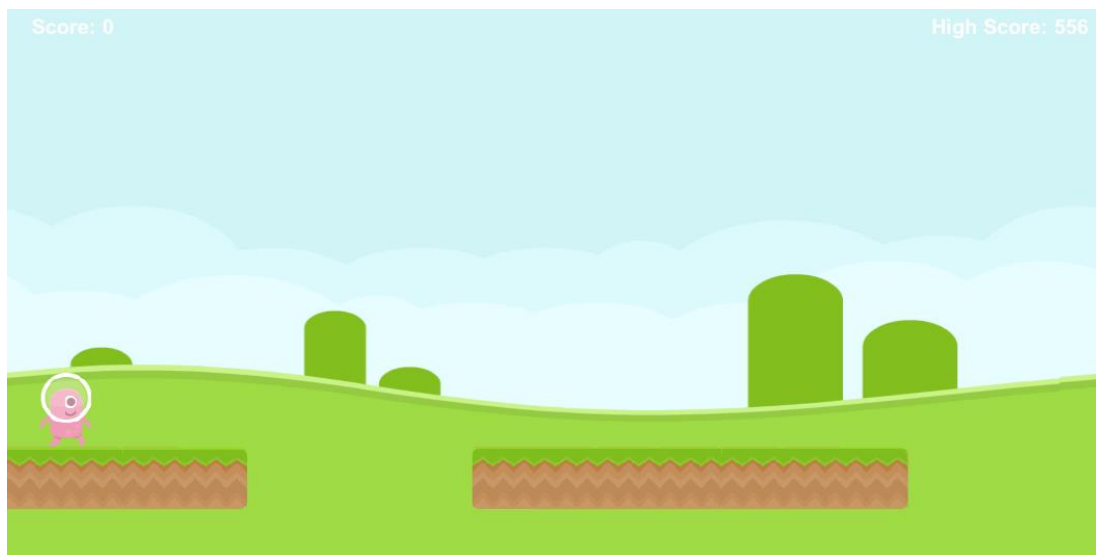
2.1. O Projekcie

Tematem projektu jest zaimplementowanie gry w 2D, w stylu *endless runner*, za pomocą narzędzia Unity. Zadaniem gracza jest przebiegnięcie jak największego dystansu oraz zebranie jak największej ilości punktów. W grze będzie istniała możliwość zbierania dodatkowych przedmiotów, które będą zwiększały końcowy wynik gracza. Docelowo, gra została napisana w konwencji programowania komponentowego.

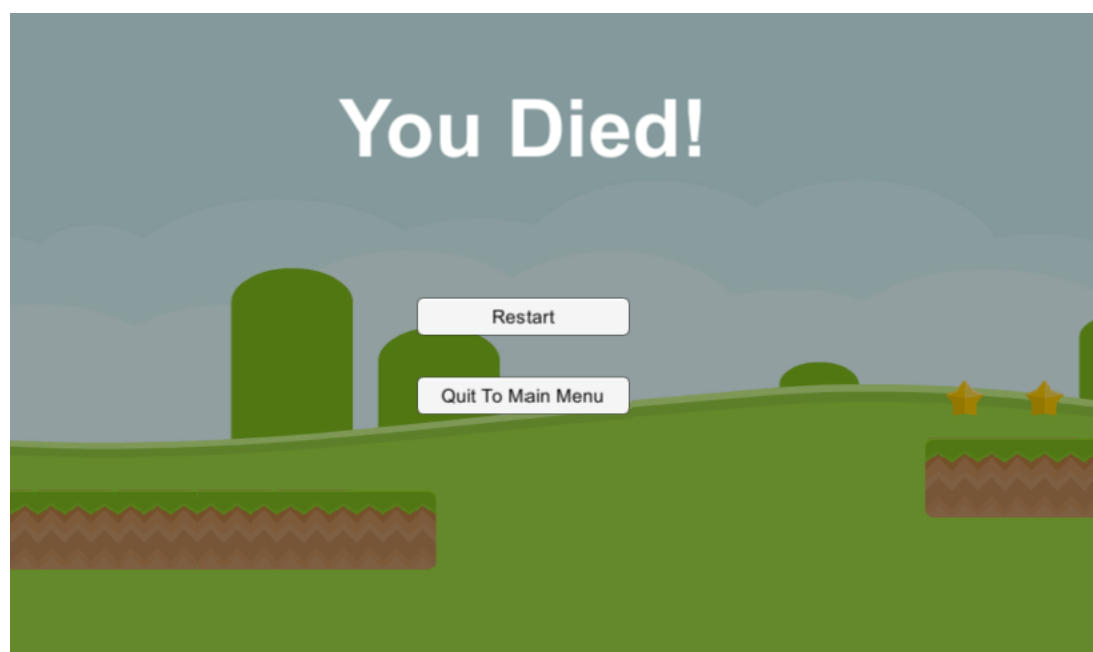
Gracz będzie miał możliwość wykonania skoku (za pomocą klawisza SPACJA), w celu zmiany platformy, po której obecnie biegnie postać. Długość przytrzymania klawisza skoku determinuje, z jaką siłą skok zostanie wykonany. Rozgrywka kończy się w momencie spadnięcia gracza z platformy. Ponieważ gra jest typu *Endless Runner* – nigdy się nie kończy, chyba, że gracz wcześniej spadnie z platformy.

2.2. Wygląd aplikacji

W momencie uruchomienia aplikacji użytkownikowi wyświetla się strona tytułowa z menu, w którym ma do wyboru uruchomienie gry, bądź wyłączenie programu. Po wybraniu opcji uruchomienia gry, wyświetla się przygotowana scena z platformami i rozpoczyna się rozgrywka. W momencie śmierci gracza, wyświetla się menu z możliwością restartu gry, bądź powrotu do głównego menu.



Rys. 1: Główna scena, po rozpoczęciu gry.



Rys. 2: Wybór możliwości po śmierci gracza.

2.3. Wygląd postaci i przedmiotów:



Rys. 3: Gwiazdka – przedmiot do zbierania, postać gracza oraz platforma.

2.4. Wykorzystane techniki

W celu zaimplementowania gry został wykorzystany silnik Unity. Do obsługi poszczególnych komponentów zostały użyte skrypty, napisane w języku programowania C#. Dodatkowo, wykorzystano darmowe grafiki i assety, dostępne w ramach darmowej licencji w internecie.

2.5. Licencja

Projekt jest oparty o licencję open-source, w naszym przypadku jest to GNU General Public Licence.

3. Platformy

Z związku z wykorzystaniem silnika Unity, możliwe jest łatwe przeniesienie programu na dowolną platformę. W naszym projekcie docelową platformą są komputery osobiste, z wbudowanym systemem Windows. Możliwe jest jednak przeniesienie gry na system Android (wymagana wersja Androida 4.4 Kit Kat). Gra po przeniesieniu jest stabilna i działa na tej samej zasadzie, co wersja komputerowa. Możliwe jest jej zainstalowanie na urządzeniu za pomocą pliku instalacyjnego .apk.

4. Inne

4.1. Niezawodność i dostępność

Niezawodność i dostępność programu jest zagwarantowana, ponieważ program nie wymaga dodatkowego połączenia internetowego.

4.2. Bezpieczeństwo

W programie nie są przechowywane dane wrażliwe.

5. Harmonogram

2 XII 2016 - Utworzenie wstępnej dokumentacji oraz diagramów UML: Use Case, Activity, Component. Rozpoczęcie prac nad główną sceną gry.

31 XII 2016 - Dodanie menu początkowego oraz po śmierci gracza, dodanie gwiazdek, jako przedmiotów do zbierania, wprowadzenie systemu punktowego i zapamiętywanie najwyższego wyniku osiągniętego przez gracza.

do 20 I 2017 - Utworzenie pełnej implementacji gry, dodanie do sceny tła, wędrującego wraz z graczem. Oprócz klawisza spacji, skok można również wykonać wciskając lewy przycisk myszy.

25 I 2017 - Oddanie końcowej wersji gry oraz pełnej dokumentacji projektu.

6. Opis techniczny tworzenia i implementacji projektu

6.1. Użyte techniki i oprogramowanie

W celu stworzenia poszczególnych diagramów UML został wykorzystany program Visual Paradigm 13.2 w wersji Community.

Następnie, implementacja programu odbyła się z wykorzystaniem silnika Unity 3D Engine 5.5. Kod źródłowy programu (skrypty obsługujące poszczególne komponenty) jest oparty na języku programowania C# - do ich edycji zostało wykorzystane środowisko Visual Studio 2015 Community.

Wszystkie grafiki, assety i tekstury, udostępnione w ramach darmowej licencji zostały pobrane z internetu.
kenney.nl/assets/platformer-pack-redux

Tutoriale wykorzystane do implementacji gry znajdują się na:
unity3d.com/learn/tutorials.

6.2. Uzasadnienie

Do zaimplementowania gry został wykorzystany silnik Unity 3D, ze względu na bardzo łatwą jego dostępność (jest to program darmowy), a dodatkowo, jest to najbardziej rekomendowane narzędzia do tworzenia tego typu gier. Jest to program bardzo intuicyjny, który umożliwia stworzenie pełnej gry (zarówno komputerowej, jak i mobilnej, ze względu na możliwość przeniesienia aplikacji na inną platformę w przeciągu kilku minut), nawet bez wcześniejszego zapoznania się z interfejsem programu.

Język C# został przez nas wykorzystany jako język, który znamy najlepiej i którym najwygodniej jest nam się posługiwać.

Ze względu na ograniczony czas przygotowania projektu zdecydowaliśmy się skorzystać z darmowych grafik i assetów, udostępnionych przez innych twórców gier. Dzięki temu rozwiązaniu udało nam się zyskać na czasie, który przeznaczaliśmy na dopracowanie silnika gry.

7. Załączenie poszczególnych diagramów UML