

**Projekt pt.**

# **“Newtons of Fun”**

**Specyfikacja**

**Janosz Zieliński**

## **Spis Treści:**

### **1. Wstęp:**

- Cel projektu..... 2
- Krótki opis realizacji..... 2
- Zakres projektu..... 2
- Terminy..... 2

### **2. Opis projektu:**

- Używane komponenty..... 2
- Omówienie działania..... 2
- Charakterystyka użytkowników..... 3
- Założenia projektu..... 4
- Ograniczenia..... 4

### **3. Wymagania projektu:**

- Wymagania Funkcjonalne.....4
- Przykłady stosowanych rozwiązań.....4
- Harmonogram tworzenia projektu..... 5
- Status dokumentu.....6

### **4. Dodatkowe informacje .....6**

# 1. Wstęp

**Celem** utworzenia tego projektu jest zaliczenie przedmiotu, wykazanie się kreatywnością, poszerzenie horyzontów w dziedzinie programowania gier oraz pomoc naukowa.

Projekt jest **grą komputerową** typu przygodówka/zręcznościówka. Gra dotyczy podstawowych praw fizyki, tj. zasady dynamiki, grawitacja, itp. Użytkownik ma do dyspozycji kilka poziomów zręcznościowych przy zastosowaniu odpowiednich dla poziomu praw fizyki, na których porusza się postacią na wzór człowieka. Plansze często odwzorowują przykładowe zadania z fizyki, np. równia pochyła, postać, którą gra użytkownik również jest poddawana różnym, czasem brutalnym testom mającym na celu udowodnienie praw fizyki. Stosowany model tworzenia będzie typu ewolucyjnego.

**Zakres projektu:** Uczelnie wyższe, szkoły na poziomach: liceum, gimnazjum, podstawówki oraz użytek domowy.

## **Terminy:**

*Gracz:* osoba korzystająca z systemu.

*Poziom:* plansza, na której rozgrywana jest główna akcja gry.

*Postać:* główny bohater rozgrywki, osoba którą steruje *gracz*.

*Baza:* "kryjówka" *postaci*, miejsce, z którego wchodzi się do innych *poziomów*.

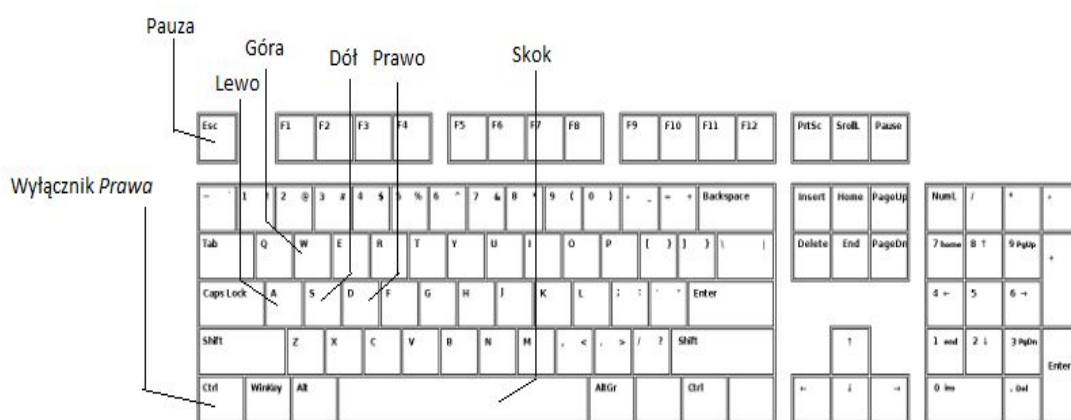
*Prawa:* prawa fizyki, jako umiejętności wykorzystywane w grze.

# 2. Opis projektu

Projekt będzie realizowany z użyciem Blueprints w silniku graficznym Unreal Engine. Tekstury będą tworzone w programie do tworzenia modeli 3D. Progres użytkownika będzie zapisywany w pliku.txt.

Program ten będzie tworzony obiektowo, wszystkie klasy będą w osobnych plikach, w których będą stosowane odpowiednie algorytmy, np. poruszanie się *postaci*, kolizja z innymi obiektami, liczenie wzorów fizycznych, itd. *Gracz* ma

możliwość swobodnego poruszania się po *bazie*, która będzie zaprojektowana w ww. silniku graficznym, wbudowana będzie opcja dowolnego sterowania kamerą za pomocą myszki. *Gracz* ma do dyspozycji następujące przyciski na klawiaturze:



*Postać* standardowo znajduje się w *bazie*, która jest głównym pomieszczeniem w grze, z którego się przechodzi do *poziomów* oraz do którego się powraca jeśli się wyjdzie albo ukończy *poziom*. W tym pomieszczeniu można również przejrzeć progres *gracza*, czyli ilość ukończonych *poziomów*, poznane *prawa*, czy też czas gry. Samo pomieszczenie ma przypominać coś w rodzaju podziemnego laboratorium naukowego, gdzie znajduje się tablica ze wzorami fizycznymi i drzwi prowadzące do *poziomów*. Każde drzwi są zablokowane i żeby je odblokować trzeba rozwiązać zagadnienie z fizyki, które dotyczy poprzedniego *poziomu*. *Gracz* ma możliwość odblokowania danego *poziomu*, jeśli to zagadnienie umie rozwiązać, w przeciwnym razie musi przejść poprzedni *poziom*, żeby je poznać. *Poziomy* będą oddzielnymi planszami, dotyczącymi konkretnego prawa fizyki, np. grawitacji. W tym przypadku *poziom* może polegać na unikaniu spadających obiektów o niebezpiecznie dużej sile ciężkości. Na niektórych rodzajach *poziomów* *postać* może zyskać *prawo*, co daje jej też możliwość wyłączenia go, np. czasowe wyłączenie grawitacji. *Postać* ma wyglądać jak szalony naukowiec. Zagadnienia na poszczególnych *poziomach* mają poniższy zakres tematyki (może on ulec zmianie w czasie ewentualnego ulepszania projektu):

- Zasady dynamiki Newtona
- Kinematyka
- Mechanika
- Zasady termodynamiki Newtona

Każdy kto posiada sprawny komputer i umiejętność obsługi myszki i klawiatury może korzystać z tej gry, ale jest skierowana przede wszystkim do studentów i uczniów liceum/gimnazjum, którzy chcą w łatwy sposób zrozumieć podstawy fizyki.

Projekt powinien uczyć od podstaw w łatwy i przyjemny sposób praw fizyki oraz dostarczać rozrywki. Powinien oferować również pomoc naukową w formie trójwymiarowego odwzorowania podstawowych zadań z fizyki, np. obliczenie przyspieszenia na równi pochyłej. Ma to na celu uzmysłowienie *graczowi* działania poszczególnych praw fizyki, w celu łatwiejszego ich zrozumienia. Program jest tylko dla jednego gracza, posiada interfejs 3D, nie wymaga połączenia z internetem.

Zalecany wymogi jest znajomość podstawowych działań matematycznych, takich jak: dodawanie, odejmowanie, dzielenie, mnożenie, czy operacje na ułamkach. Jak również znajomość języka angielskiego.

### 3. Wymagania Projektu

#### Wymagania funkcjonalne:

**Interfejs menu** posiada następujące elementy:

- Nowa gra
- Kontynuacja
- opcje
- wyjście z gry

Wybranie nowej gry spowoduje rozpoczęcie gry od nowa, w przypadku kiedy *gracz* ma jakiś progres, jest on całkowicie kasowany.

Wybranie kontynuacji spowoduje przeniesienie *gracza* do *bazy*, z całym zrobionym wcześniej progresem.

element “opcje” powoduje przeniesienie się do **interfejsu opcji**, w którym są następujące opcje:

- ustawienie rozdzielczości
- zgłaszanie/przyciskanie dźwięku
- zmiana klawiszy sterowania
- przycisk powrotu do **menu**

Wybranie wyjścia skutkuje zamknięciem aplikacji.

**Interfejs pauzy** posiada następujące elementy:

- Powrót do gry
- Opcje
- Powrót do bazy

Powrót do gry powoduje wyłączenie **interfejsu pauzy**.

Wybranie opcji powoduje włączenie **interfejsu opcji**.

Wybierając powrót do bazy decydujemy się na opuszczenie *poziomu* i przeniesienie *postaci* z powrotem do *bazy*.

**Interfejs ekranu rozgrywki** pokazuje obliczenia dotyczące aktualnego *poziomu* i *prawa*, wzory fizyczne oraz zmianę jakiejś wielkości fizycznej w czasie, np. zmieniająca się w czasie rzeczywistym prędkość. W lewym górnym rogu ekranu pokazane są wszystkie stałe, takie jak przyspieszenie grawitacyjne, czy współczynnik tarcia.

Wszystkie obiekty i *poziomy* są projektowane w **Unreal Engine**, w 3D. Gracz ma widok tylko z trzeciej osoby, czyli widzi *postać* zza jej pleców. **Algorytmy** dotyczące działania “fizyki” w grze są oparte na wzorach fizycznych, sama “fizyka” w grze musi niemalże identycznie odzwierciedlać prawdziwe prawa fizyki.

**Przykłady stosowanych rozwiązań (algorytm na siłę ciężkości opisany pseudokodem):**

*funkcja siła ciężkości*

*zwróć masa obiektu pomnożona przez przyspieszenie ziemskie*

*koniec funkcji*

*funkcja spadanie*

*jeżeli siła ciężkości obiektu większa od zera i przyspieszenie ziemskie większe od zera, to dopóki położenie obiektu nie jest równe położeniu podłoża wykonuj położenie jest równe różnicy położenia i prędkości zwiększanej co kwadrat sekundy o przyspieszenie ziemskie koniec dopóki, koniec jeżeli, koniec funkcji*

Powyższy pseudokod liczy siłę ciężkości obiektu, po czym wywołuje funkcję spadania, czyli zwiększa prędkość danego obiektu, który spada w dół.

**Harmonogram tworzenia projektu:**

- Przygotowanie używanych komponentów, stworzenie repozytorium i połączenie go z visual studio, instalacja i implementacja silnika graficznego. (1 tydzień)

- Tworzenie interfejsu menu i systemu zapisywania gry, testowanie działania programu w międzyczasie. Robienie testowego *poziomu*, implementacja silnika graficznego, tworzenie struktur danych, klas i obiektów (2 tygodnie)
- Wstępne Tworzenie modeli *postaci*, obiektów na mapie i tekstur (1 tydzień)
- Naprawa błędów i silnika graficznego (1 tydzień)
- Stworzenie *bazy* (1 tydzień)
- Dodawanie interfejsu opcji i pauzy, robienie interfejsu rozgrywki, dopracowywanie detali, Dodanie do gry dźwięków i muzyki (1 tydzień)
- Zrobienie ostatecznej wersji aplikacji i poddawanie jej testom (1 tydzień)

#### **Status dokumentu:**

autor: Janosz Zieliński

tytuł: Newtons of Fun

data opracowania dokumentu: 19.11.2017 r.

ostatnia modyfikacja: 05.01.2017 r.

czas na realizację projektu: 8 tygodni (do 19.01.2018 r.).

aktualny stan: Stworzenie *bazy*.

## **4. Dodatkowe informacje**

- Gra jest w języku angielskim
- Gra posiada autozapis
-