Adam Borowski

Radek Pieczątkiewicz

Gdańsk, 03.06.2015r.

MSIBall

Dokumentacja API

PlayerController.cs

Główna klasa gry. Wykorzystuje Erf i komunikuje mu o zdarzeniach. Zajmuje się też obsługą zdarzeń zwrotnych. W zależności od emocji gracza dostosowuje poziom trudności w czasie rzeczywistym za pomocą funkcji:

```
setDifficultyEnum(GameDifficulty difficulty, bool play=true);
```

```
updateSpeed();
```

```
setDifficulty(float dif);
```

Uaktualnia zmienne rozgrywki za pomocą funkcji FixedUpdate(). Definiuje m.in. prędkość gracza oraz bezwładność. Głównym zdarzeniem gry jest kolizja, która obsługiwana jest za pomocą funkcji

```
OnCollisionEnter(Collision other)
```

CameraController.cs

Prosta klasa sterująca jedyną kamerą dostępną w trakcie rozgrywki. Determinuje jej podążanie za kulą - graczem.

ErfInitialization.cs

Służy załadowaniu biblioteki rozpoznającej emocje do kontekstu. Dzięki niej można w klasie PlayerController posłużyć się "using Erf".

```
ErfContext.GetInstance ().LoadComponentLibrary ("MSIBallErf.dll");
```

W drugiej kolejności wyszukuje Eksperta naszej gry za pomocą

```
ErfContext.GetInstance ().FindExpert ("MSIBallEmotionModel");
```

Dzięki temu dopasowujemy kontroler do zdarzeń występujących w MSIBall.

GameSettings.cs

Definiuje globalne ustawienia przekazywane od sceny początkowej MainMenuScene do sceny z rozgrywką BallGame. Określa tryb rozgrywki oraz poziom trudności.

GroundManager.cs

Gra zbudowana jest na idei Endless Runner. Klasa odpowiada za generowanie kolejnych elementów trasy. By uniknąć zbędnego tworzenia i usuwania obiektów minięte przez gracza fragmenty przesuwane są do przodu z wykorzystaniem kolejki

```
Queue<Transform> objectQueue;
```

Za zmiany pozycji odpowiedzialna jest funkcja

```
Recycle();
```

MainMenu.cs

Definiuje zawartość początkowego menu. Pozwala wybrać poziom trudności i tryb przed rozpoczęciem rozgrywki. Przekazuje parametry do kolejnej sceny. Korzysta z definicji GameSettings.

ObstacleGenerator.cs

Generuje koleje przeszkody, czyli prostokątne obiekty na planszy. Definiuje odstępy między kolejnymi przeszkodami oraz ich rozmiary w zależności od położenia i rozmiarów poprzednich wygenerowanych przeszkód. Umieszcza je na kolejnych fragmentach planszy generowanych przez GroundManager.

ObstacleManager.cs

Zarządza powstałymi przeszkodami - głównie pamięcią, czyli usuwa niepotrzebne obiekty.

ScoreController.cs

Odpowiada za wyświetlanie aktualnego wyniku w prawym górnym rogu ekranu gry. Dodatkowo wyświetla informacje o czasie, Booscie oraz ostatnich 10s rozgrywki (ilość kolizji).