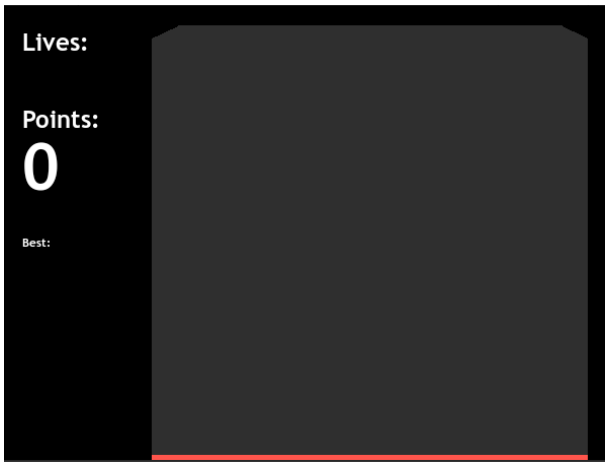
Mobilne Aplikacje Multimedialne – Laboratorium

Unity: Arkanoid

Instrukcja opracowana przez: Krzysztof Drypczewski Dawid Grajczyk Łukasz Markiewicz

2. Ćwiczenie laboratoryjne.

Zadaniem studenta jest rozbudowanie projektu gry typu „Arkanoid” utworzonego w środowisku Unity3D. W tym celu należy uruchomić Unity3D i wybrać folder z projektem. Następnie w oknie ***Project*** należy rozwinąć zawartość katalogu ***Scenes*** i otworzyć znajdującą się w nim utworzoną scenę poprzez podwójne kliknięcie. Projekt zawiera prostą scenę, której elementom można się przyjrzeć bardziej wnikliwie w oknie ***Scene*** w edytorze. W oknie ***Game*** widzimy widok części sceny, jaki zostanie wyrenderowany na ekranie i będzie widoczny dla użytkownika.



Rys 2.1: *Wygląd okna “Game” po otwarciu przygotowanej sceny.*

Zadanie 1:

Pierwszym zadaniem studentów jest dodanie do gry najważniejszych elementów gry: sterowanej za pomocą klawiatury paletki, oraz odbijającej się piłeczki.

W tym celu należy:
a) dodać do sceny 2 nowe ***GameObject-y*** – wykorzystać predefiniowane kształty: piłeczka – sfera, paletka – do wyboru: sześcian lub cylinder;
b) dobrać odpowiednią skalę i proporcje obiektów, umieścić je w odpowiednim miejscu na scenie;
c) utworzyć w projekcie 2 nowe materiały dla 2 nowych obiektów – zielony dla piłeczki oraz szary dla paletki;
d) upewnić się, że obiekty posiadają komponent typu ***Collider*** w celu umożliwienia detekcji kolizji obiektów;
e) utworzyć i dodać do obiektu piłki odpowiednio skonfigurowany materiał fizyczny (***Physic Material***);
f) dodać do piłki komponent ***Rigidbody*** i ustawić w nim odpowiednie parametry;
g) dodać do obiektu piłeczki komponent ***Audio Source*** i przeciągnąć odpowiedni klip na niego;
h) dodać do każdego z nowych obiektów odpowiedni skrypt znajdujący się już w projekcie w katalogu ***Scripts*** - dla piłeczki ***BallScript.cs***, dla paletki ***PaddleScript.cs***.

Aby dodać wspomniane powyżej ***GameObject-y*** należy skorzystać z menu ***GameObject → Create Other***. Znajduje się tam lista kilku podstawowych predefiniowanych rodzajów ***GameObject-ów*** jakie możemy w prosty sposób oddać do sceny. Z dostępnych elementów wybieramy ***Sphere***, która będzie naszą piłeczką, oraz ***Cylinder*** lub ***Cube*** będące sterowaną przez gracza paletką.

Po utworzeniu 2 nowych obiektów zmieńmy ich domyślne nazwy na: ***Ball*** i ***Paddle***. Zaznaczając odpowiedni obiekt w oknie ***Hierarchy*** wszystkie niezbędne informacje wyświetlają się w oknie ***Inspector***. Jest w nim obszar nazwany ***Transform*** - są to informacje opisujące położenie i rozmiary danego obiektu. Jako współrzędne piłeczki podajemy wektor (2, -6, -1), natomiast współrzędne paletki to (2, -6.5, -1). Wektor skali piłeczki należy ustawić na około (0.5, 0.5, 0.5), a paletki na około (3, 0.5, 1).

Aby utworzyć nowy materiał przechodzimy do pustego folderu ***Resources/Materials*** (okno ***Project***) i klikamy na niego prawym klawiszem myszy. Z menu kontekstowego wybieramy ***Create → Material***. Zmieniamy nazwę utworzonego materiału, oraz nadajemy mu odpowiedni kolor (***Main Color***). Piłeczka ma mieć materiał koloru zielonego, a paletka kolor szary.
Aby dodać tak utworzone materiały do odpowiedniego ***GameObject-u*** należy zaznaczyć dany obiekt w oknie ***Hierarchy*** (w oknie ***Inspector*** pojawią się informacje o obiekcie), a następnie przeciągnąć materiał z folceru ***Resources/Materials*** do okna ***Inspector*** zaznaczonego obiektu.

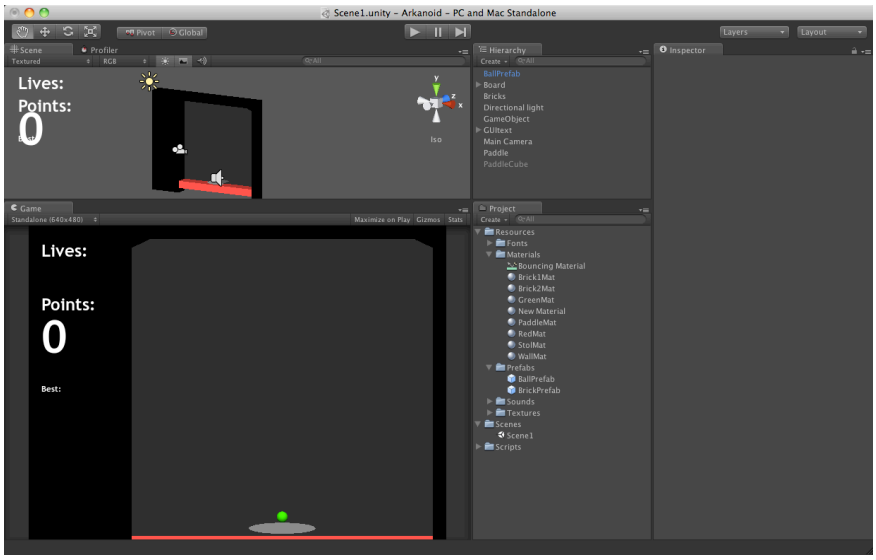
Dodane przez nas obiekty posiadają tzw. kolidery, dzięki czemu będzie można zaprogramować odpowiednie akcje wywoływane przy kolizjach tych obiektów.

Aby wykorzystać engine Unity3D do symulowania fizyki odbijającej się piłeczki należy dodać jej ***Physic Material***. W tym celu wybieramy z menu kontekstowego ***Create → Physic Material***. Zmieniamy jego nazwę na ***Bouncing Material***. Aby piłeczka była idealnie sprężysta ustawiamy współczynnik ***Bounciness*** na wartość ***1***, ***Friction Combine*** na ***Minimum***, ***Bounce Combine*** na ***Maximum***, a pozostałym parametrom przypisujemy ***0***. Tak przygotowany materiał fizyczny dodajemy do obiektu piłeczki. Dodany materiał fizyczny pojawi się w sekcji ***Sphere Collider → Material***.

Kolejnym krokiem będzie dodanie do obiektu piłeczki komponentu ***Rigidbody***. W tym celu zaznaczamy odpowiedni ***GameObject***: w oknie ***Hierarchy*** i wybieramy z menu: ***Components → Physics → Rigidbody***. Dokonujemy następującej konfiguracji komponentu: ustawiamy niewielką masę ***0.05***, zerujemy wartość pola ***Angular Drag***, zaznaczamy ***Freeze rotation***, oraz odznaczamy ***Use Gravity***, ponieważ piłeczka nie powinna w grze typu „Arkanoid” podlegać grawitacji.

Aby dodać komponent ***AudioSource*** wybieramy ***Components → Audio → Audio Source***. Aby dodać do ***AudioSource'a*** odpowiedni klip zaznaczamy obiekt (w oknie ***Hierarchy***) i przeciągamy klip ***bleep.wav*** z katalogu ***Resources/Audio*** na pole ***Audio Clip*** w obszarze o nazwie ***Audio Source***.

Następnie do obiektów należy dodać gotowe skrypty. Znajdują się one w folderze ***Scripts***. Dodaje się je w sposób analogiczny do dodania materiałów. Do piłeczki należy dodać skrypt ***BallScript.cs***, a do paletki ***PaddleScript.cs***. Należy przeanalizować kod zawarty w skryptach wraz z komentarzami. W tak przygotowanej scenie, jeśli wszystkie ktoki wykonane zostały poprawnie, powinniśmy uzyskać paletkę poruszającą się za pomocą klawiszy prawej/lewej strzałki, oraz piłeczkę odbijającej się od tej paletki i od ścian. Piłeczka początkowo znajduje się na paletce, aby rozpocząć grę należy wcisnąć klawisz spacji.



Rys 2.2: *Widok okna Unity3d po zakończeniu Zadania 1.*

Zadanie 2:

Kolejne zadanie polegać będzie na rozbudowie gry w taki sposób, aby po utracie piłki (nieudane odbicie przez gracza) na scenie utworzył się nowy obiekt piłki. Ponadto zadaniem studenta jest również utworzenie na scenie “muru” cegieł, które będą znikały po uderzeniu przez piłeczkę. Po wykonaniu zadania powinna powstać gra w którą będzie można zagrać.

Zadanie składać się będzie z następujących kroków:
a) utworzenie na scenie obiektu cegiełki;
b) utworzenie i dodanie do obiektu cegły skryptu, dzięki któremu trafiona cegła zniknie ze sceny;
c) utworzenie prefabów z obiektów cegły oraz piłeczki;
d) modyfikacja skryptu ***GameScript.cs*** tak, aby na scenie po uruchomieniu gry został utworzony “mur” z cegieł.

Cegła ma być obiektm typu ***Cube*** ze skalą: (1, 0.5, 2). Należy utworzyć dla cegły nowy materiał z teksturą ***brick1.jpg*** (która znajduje się w katalogu ***Resources/Textures*** w projekcie).

Nowy skrypt, a także nowy (pusty) ***Prefab*** dodaje się do projektu w sposób analogiczny do dodania nowego materiału (opisany w poprzednim zadaniu). Aby utworzyć z danego obiektu na scenie ***Prefab*** należy przeciągnąć ten obiekt z okna ***Hierarchy*** na utworzony wcześniej pusty ***Prefab*** (w oknie ***Project***). Po tej czynności można usunąć obiekt ze sceny.

Nazwa skryptu musi być taka sama jak nazwa klasy w nim zawartej dziedziczącej po ***MonoBehaviour***, dlatego przy zmianie nazwy skryptu należy pamiętać o zmianie nazwy klasy.

Przy implementacji skryptu cegły należy skorzystać z jednej z poniższych predefiniowanych funkcji:

```
void OnCollisionEnter(Collision collisionInfo)
void OnCollisionStay(Collision collisionInfo)
void OnCollisionExit(Collision collisionInfo)
```

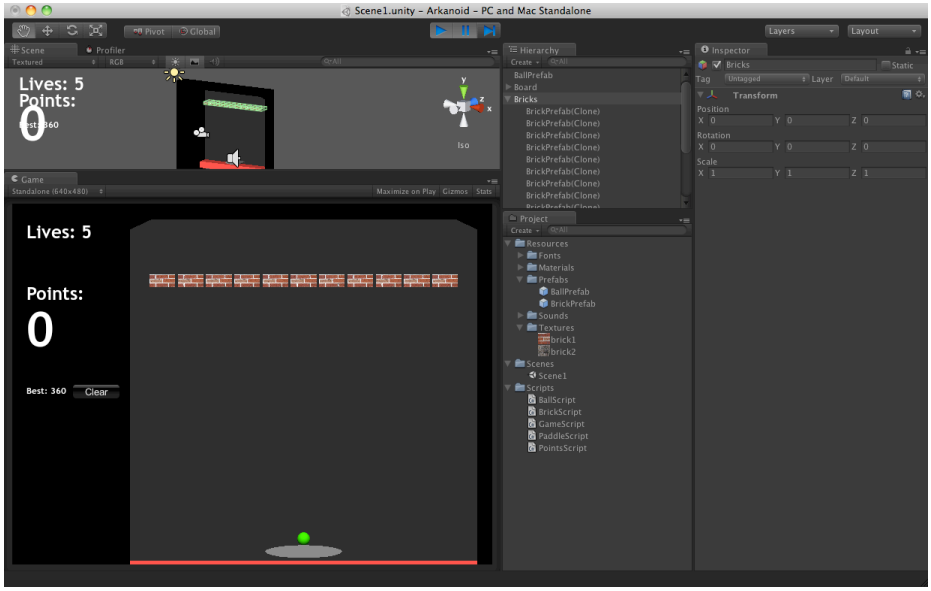
Gdy już będziemy mieli gotowe prefaby naszych obiektów do utworzenia ich instancji na scenie służy funkcja:

```
Object Instantiate(Object original)
```

Funkcja ta zwraca również referencję do utworzonego obiektu, co można wykorzystać do uzyskania dostępu do instancji skryptu podłączonego do danego obiektu.

Modyfikacji w skrypcie ***GameScript.cs*** dokonujemy w funkcji ***Start*** (jest ona wykonywana 1 raz, od razu po załadowaniu sceny). Utworzone na scenie cegły mają się znaleźć w pustym obiekcie o nazwie ***Bricks*** (znajduje się już na scenie). Aby przypisać obiektowi ***Transform*** inny obiekt ***Transform*** jako rodzica należy wpisać referencję rodzica do pola ***parent*** obiektu-dziecka.

Aby utworzyć na scenie instancję piłki wystarczy ustawić referencję ***ballPrefab*** w skrypcie ***GameScript.cs*** na nasz prefab piłki, a następnie w funkcji ***Start*** tego skryptu wywołać metodę ***InstantiateBall***. Metodę tę należy wywołać również za każdym razem gdy piłka wypadnie ze sceny (skrypt ***BallScript.cs***, metoda ***BallFail***).



Rys 2.3: *Widok okna Unity3d po zakończeniu Zadania 2.*

Zadanie 3:

Zadanie trzecie będzie polegać na dodaniu do projektu funkcjonalności zdefiniowanej przez prowadzącego na zajęciach.

Wykonując wszystkie zadania należy korzystać z dokumentacji Unity3D znajdującej się pod adresem: http://docs.unity3d.com/Documentation/ScriptReference/index.html