Paweł Kolak 218310

Dawid Mikowski

Wojciech Urbańczyk

Denis Lewandowski 226404

Grafika 3D i systemy multimedialne

Laboratorium

Temat projektu: **"Gra w kręgle"**

Prowadzący: dr inż. Jan Nikodem

Spis Treści

[1. Cel i zakres projektu 3](#_Toc30364910)

[2. Technologia 3](#_Toc30364911)

[3. Scena i opis przedmiotów 3](#_Toc30364912)

[4. Implementacja 4](#_Toc30364913)

[4.1. Modele 3D 4](#_Toc30364914)

[4.2. Fizyka 6](#_Toc30364915)

[4.2.1. Bryła sztywna 6](#_Toc30364916)

[4.2.2. Zderzaki 7](#_Toc30364917)

[4.2.3. Materiał fizyczny 7](#_Toc30364918)

[4.2.4. Ciała elastyczne 8](#_Toc30364919)

[4.3. Teksturowanie 9](#_Toc30364920)

[4.4. Oświetlenie 10](#_Toc30364921)

[4.5. Kamery 10](#_Toc30364922)

[4.6. Sterowanie 12](#_Toc30364923)

[4.7. Udźwiękowienie 13](#_Toc30364924)

[4.8. Interfejs użytkownika 13](#_Toc30364925)

[5. Logika gry 13](#_Toc30364926)

[6. Podsumowanie 13](#_Toc30364927)

# Cel i zakres projektu

Celem projektu jest zamodelowanie oraz oprogramowanie fizyki sceny 3D toru do kręgli. W ramach projektu należy zamodelować obiekty takie jak:

* Kule
* Kręgle
* Tor
* Pomieszczenie
* Sofa
* Stół
* Maszyna do ustawianie kręgli
* Maszyna do zwracania kul
* Lampy
* Okna
* Puchary

# Technologia

Zadanie projektowe zostanie zrealizowane w następujących środowiskach:

* Microsoft Visual Studio,
* Unity 3D,
* Blender.

Jako środowisko 3D posłuży oprogramowanie Unity. Zostanie także użyty dostarczony wraz z Unity silnik fizyki. Wszelkie skrypty zostaną napisane w języku C# z pomocą środowiska Microsoft Visual Studio. Obiekty 3D zostaną stworzone w programie Blender 3D. Użyte narzędzia wspomagające pracę to repozytorium kodu Github, oraz portal trello.com, w którym w łatwy i intuicyjny sposób można odwzorować „Kanban board”.

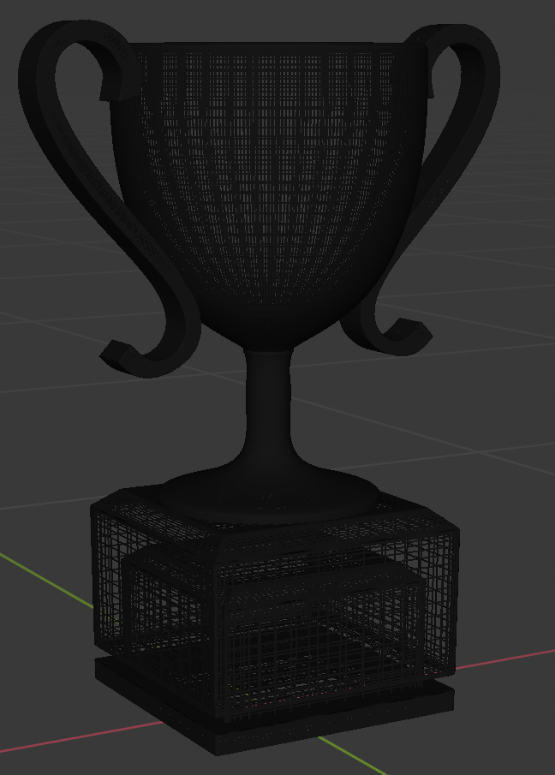
# Scena i opis przedmiotów

Scena 3D znajdzie się w kręgielni, zadaniem gracza będzie przy pomocy kuli rozbić kręgle znajdujące się na końcu parkietu. Kule są dostarczane przez maszynę do zwrotu kul. Po końcu rundy kręglem są ustawiane przez maszynę na swoje odpowiednie miejsca. W pomieszczeniu znajduje się wyświetlacz z wynikami. Za plecami gracza znajduje się przestrzeń w której znajdzie się sofa oraz stół. Pol lewej stronie toru znajdują się okna.

# Implementacja

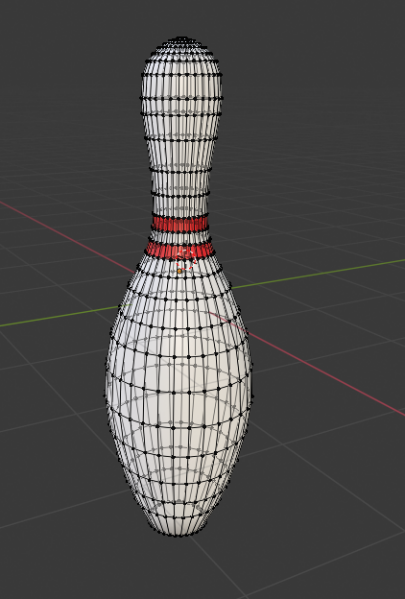
## Modele 3D

Większość modeli została wykonana przy użyciu oprogramowania Blender. Modele te były wykonywane ze starannością, aby każdy element po późniejszym teksturowaniu i renderowaniu wyglądał maksymalnie realistycznie.



Rysunek 1 Model pucharu wykonany w programie Blender.

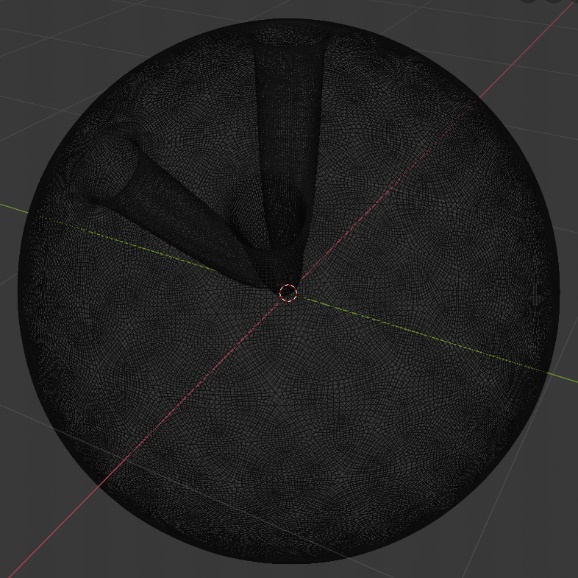
Wszystkie modele starano się wykonać w odpowiednich wymiarach, które mają przełożenie w rzeczywistości. Kręgle zostały zaprojektowane zgodnie z normami dotyczącymi gry w kręgle. Każdy kręgiel ma w najszerszym punkcie 12cm średnicy. Jego wysokość to 38cm natomiast waga 1,5kg.



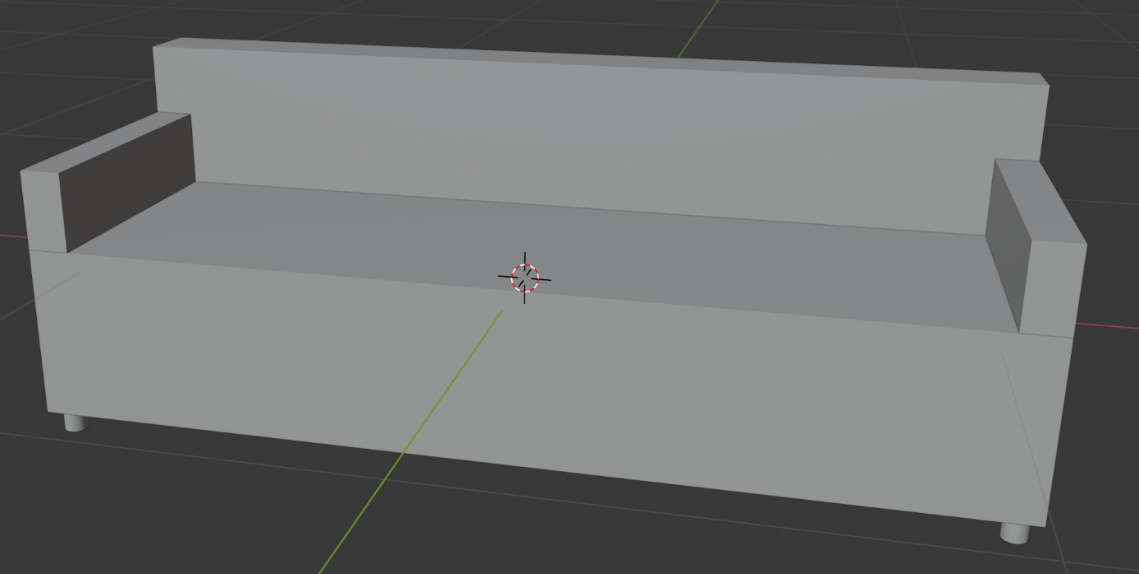
Rysunek 2 Model kręgla



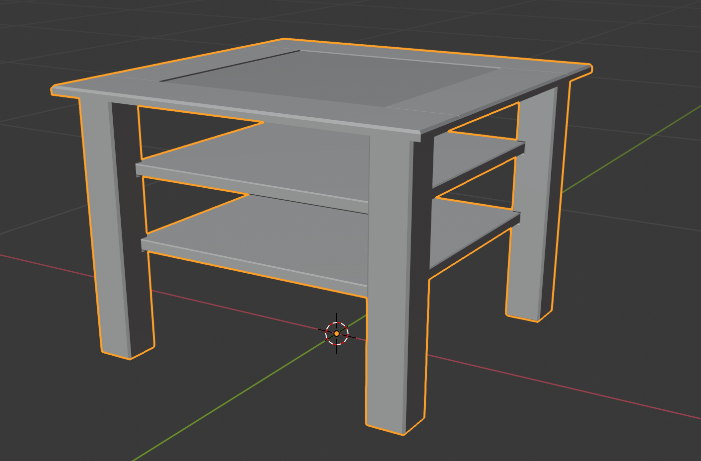
Rysunek 3 - model kuli do kręgi z białą matową powierzchnią



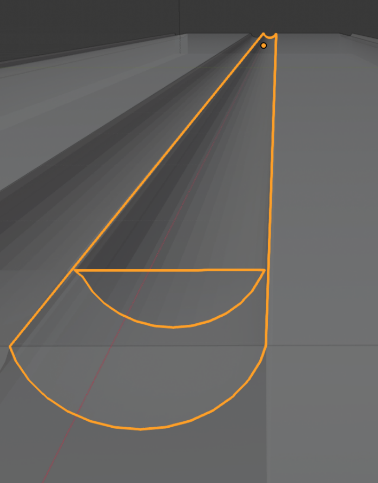
Rysunek 4 - Model kuli do kręgi siatka



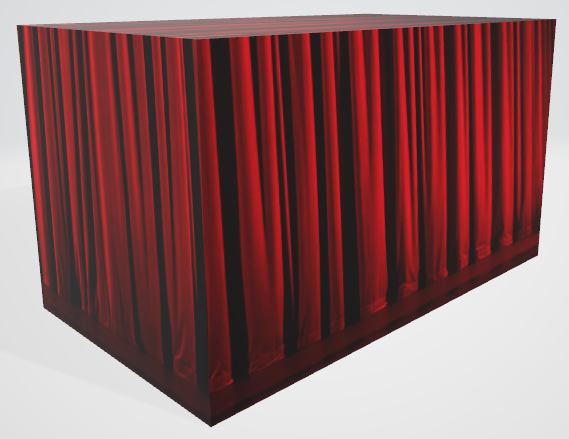
Rysunek 5 - Model sofy stworzony dla części „wypoczynkowej” kręgielni.



Rysunek 6 - Model stołu z przeszklonym blatem dla części „wypoczynkowej” kręgielni.



Rysunek 7 - Model toru i rynny na kule, które wypadły z tor



Rysunek 8 - Modle kurtyny (mechanizmu do sprzątania i stawiania kręgli)

## Fizyka

Interakcja obiektów fizycznych w projekcie jest zrealizowana z wykorzystaniem silnika fizyki

dostępnego w unity. Unity dostarcza kilka komponentów, które są odpowiedzialne za zachowanie i

wzajemną interakcję obiektów. Poniżej opisano wykorzystane w projekcie komponenty.

## Bryła sztywna

Podstawowym elementem realizującym oddziaływania fizyczne w unity jest bryła sztywna

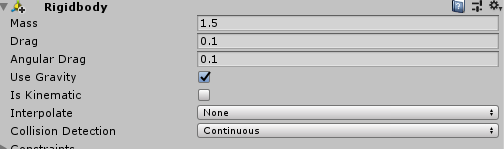
(Rigidbody). Komponent ten pozwala ustawić masę obiektu oraz opory dla ruchu obrotowego i

przesuwnego. Pozwala też włączyć grawitację dla obiektu i wybrać czy obiekt ma być statyczny czy

dynamiczny. Obiekt statyczny pozostaje w bezruchu mimo działającej na niego siły grawitacji. W

przypadku realizowanego w ramach projektu programu takim obiektem będzie na przykład pokój w

którym znajduje się tor.



Rysunek 9 - Ustawienia bryły sztywnej

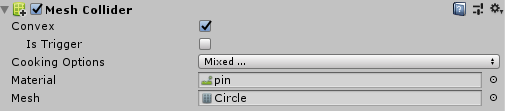
### Zderzaki

Komponent odpowiedzialny za wzajemną interakcję obiektów to zderzak (Collider). W unity

jest dostępnych kilka domyślnych modeli zderzaka takich jak sfera, pudełko i siatka i inne. Komponent ten pozwala na ustawienie wymiarów oraz materiału fizycznego. Kule zostały wyposażone w zderzaki oparte o sferę. Zderzak kręgli natomiast składają się z siatki naniesionej na kręgle, jest to również jeden z domyślnych zderzaków dostępnych w unity. Zderzaki posiadają również tor, rynny. Na obrazie poniżej przedstawiony jest zderzak siatkowy kręgla.



Rysunek 10 - Zderzak kręgla



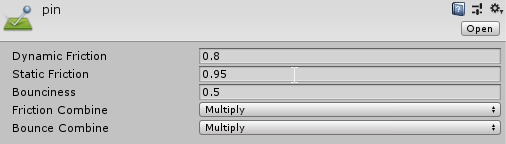
Rysunek 11 - Zderzak siatkowy

### Materiał fizyczny

Kolejnym wykorzystanym elementem jest materiał fizyczny (Physic Material), który umożliwia

określenie tarcia statycznego i dynamicznego, a także sprężystości. Na potrzeby projektu stworzono

kilka różnych materiałów, między innymi materiał kuli , czy materiał toru kręglowego.



Rysunek 12 - Materiał kręgla

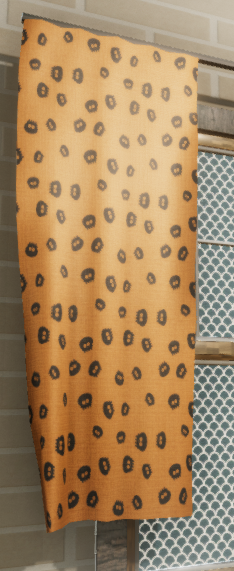
Miejscem w którym zachodzi najwięcej reakcji fizycznych jest punkt spotkania kuli z kręglami. Zachodzą w tym miejscu oddziaływania pomiędzy kulą, kręglami, torem oraz rynnami.



Rysunek 13 - Zderzenie kuli z kręglami

### Ciała elastyczne

W projekcie zastosowane zostały również ciała elastyczne. W naszym przypadku jest to zasłona na której został pokazany efekt powiewania.



Rysunek 14 - Firanka z efektem powiewania

## Teksturowanie

Wykonane wcześniej modele odpowiednio oteksturowano. Dzięki róznym teksturom można było wykorzystać te same modele tak, aby scena gry nie wydawała się monotonna.

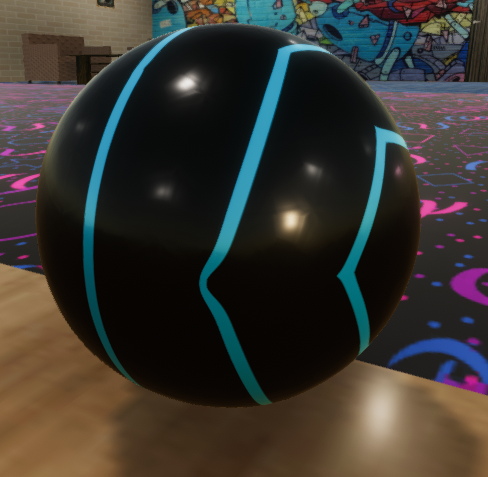


Rysunek 15 - Oteksturowane złoty puchar

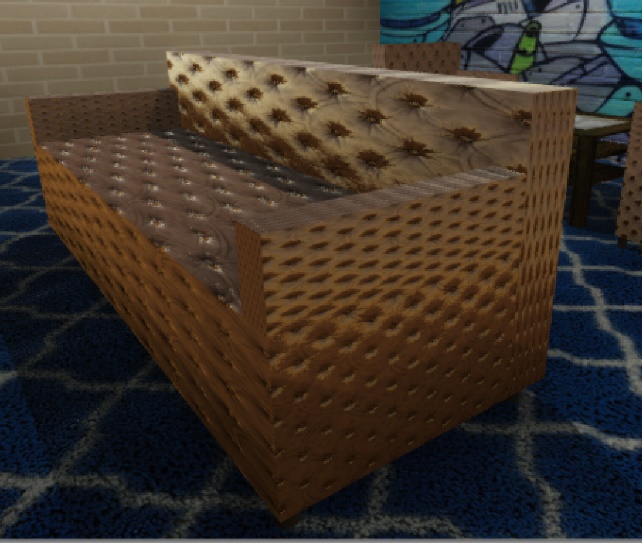


Rysunek 16 - Oteksturowane srebrny puchar

Kule do kręgli oteksturowane w taki sposób aby widoczna była rotacja jaka jest nadana kuli. Na kuli dodatkowo widoczne są odbicia światła i efekt lustra.



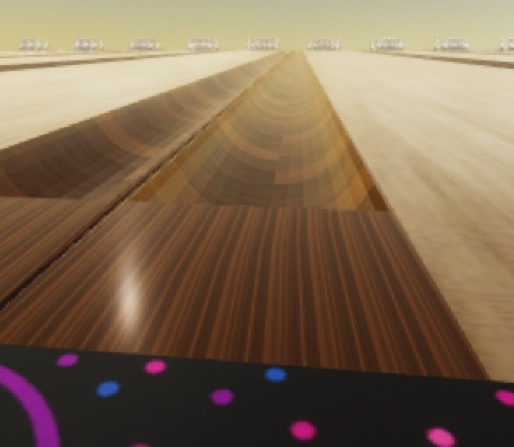
Rysunek 17 - Oteksturowana kula do kręgi.



Rysunek 18 - Dodanie tesktury dającej efekt pikowania.



Rysunek 19 - Teksturowanie drewnianego przeszklonego stolika.

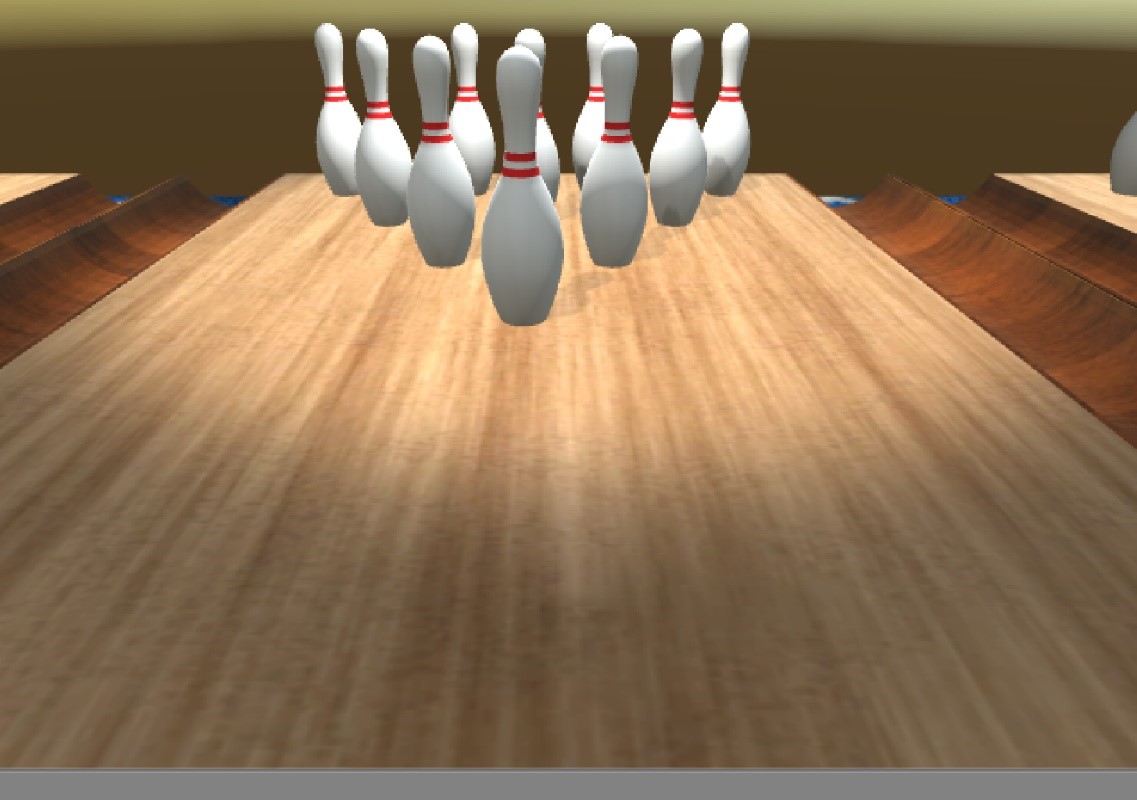


Rysunek 20 - Tesktura drewna dla rynny na kule do kręgi.

## Oświetlenie

## Kamery

W projekcie dostępny jest widok z trzech różnych kamer. Pierwsza jest tzn. wolną kamerą, dzięki której możemy zobaczyć cały obszar gry. Druga kamera jest kamerą statyczną z widokem na ustawione kręgle, które będą „rozbijane” w dalszym etepie gry. Trzecia kamera to widok gracza, który rzuca kulą do kręgi.



Rysunek 21 - Kamera na kręgle

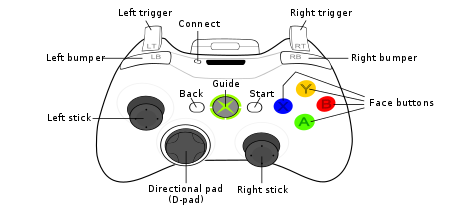
## Sterowanie

Sterowanie dla „wolnej kamery” służącej do podglądu kręgielni.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Myszka** | **Klawiatura** | **Pad** |
| **Ruch po osi y+** | - | W lub strzałka w górę | Right trigger\* |
| **Ruch po osi y-** | - | S lub strzałka w dół | Left trigger\* |
| **Ruch po osi x+** | - | D lub strzałka w prawo | Left stick\* w prawo |
| **Ruch po osi x-** | - | A lub strzałka w lewo | Left stick\* w lewo |
| **Ruch po osi z+** |  | Q | Left stick\* w górę |
| **Ruch po osi z-** |  | E | Left stick\* w dół |
| **Obracanie po sferze** | PPM | - | Right stick\* |
| **Przyspieszenie** | - | Lewy Shift | Y |
| **Zmiana Kamery** | - | Lewy Ctrl | Right/Left bumper |
| **Nowa kula** | - | X |  |
| **Ustaw kręgle** | - | Z |  |
| **Rzut kulą** | - | Spacja |  |
| **Kula w lewo** | - | 1 + ‘+’\* |  |
| **Kula w prawo** | - | 1 + ‘-’\* |  |
| **Kula wyżej** | - | 2 + ‘+’\* |  |
| **Kula niżej** | - | 2 + ‘-’\* |  |
| **Większa siła wyrzutu** | - | 3 + ‘+’\* |  |
| **Mniejsza siła wyrzutu** | - | 3 + ‘-’\* |  |
| **Kąt rzutu w lewo** | - | 4 + ‘+’\* |  |
| **Kąt rzutu w prawo** | - | 4 + ‘-’\* |  |
| **Rotacja wokół osi X** | - | 5 + ‘+’\* lub 5 + ‘-’\* |  |
| **Rotacja wokół osi Y** | - | 6 + ‘+’\* lub 6 + ‘-’\* |  |
| **Rotacja wokół osi Z** | - | 7 + ‘+’\* lub 7 + ‘-’\* |  |

Tabela 1 - Opis sterowania dla wolnej kamery.

\*przycisk liczby z klawiatury standardowej + i - klawiatury numerycznej



Rysunek 22 - Nazwy przycisków kontrolera Xbox 360.

## Udźwiękowienie

W ramach udźwiękowienia zostały dodane do projektu dźwięki w formacie .wav. Do sceny gry

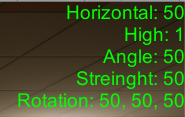
zostały dodane obiekty typu AudioSource, których zadaniem jest odtwarzanie plików .wav. Pierwszym obiektem jest BackgroundMusic, który odtwarza muzykę w tle. Na początku rozgrywki obiekt jest inicjalizowany i odgrywa muzykę w tle do momentu zakończenia gry. Do obiektu Kuli oraz kręgli dodane zostały komponenty Audio Source, które mają za zadanie odgrywać dźwięk toczenia się oraz zbijania kręgli.

## Interfejs użytkownika

W ramach interfejsu użytkownika zostało dodanych kilka informacji ułatwiających rozgrywkę.

Podczas gry w prawym górnym rogu ekranu znajdują się ustawienia rzutu kulą. Są to odpowiednio: pozycja na torze, wysokość, kąt, siła oraz wektor rotacji. W prawym górnym

ekranie wyświetlona jest informacja dotycząca ramek na sekundę.



Rysunek 23 - Parametry rzutu

# Logika gry

Użytkownik rozpoczyna rozgrywkę z domyślnymi ustawieniami które spowodują iż kula bez rotacji poleci prosto w kręgle znajdujące się na końcu toru. Po każdym zbiciu kręgli uruchamiana jest procedura gdzie opuszczana jest maszyna która w zbiera wszystkie wywrócone kręgle a w przypadku zbicia wszystkich podaje ich nowy zestaw.

# Podsumowanie

Mimo małej ilości czasu oraz nieznajomości zarówno technologii projektowania modeli 3D jak i programowania gier, udało się zrealizować wszystkie najważniejsze założenia projektu. Sprawny postęp prac jest wynikiem dobrego podziału obowiązków oraz dobrej komunikacji.

Do realizacji zadań określonych w projekcie członkowie zespołu musieli również zapoznać się z nowymi technologiami. Środowisko Unity okazało się być dość przyjemnym w użyciu, zarówno jeśli chodzi o pracę z UI, jak i próg wejścia okazał się niezbyt wysoki. Pozytywnie zaskoczyła również integracja ze środowiskiem Visual Studio, które było już dobrze znane większości zespołu.

Większe trudności były natomiast napotykane podczas pracy z programem do tworzenia obiektów 3D – Blender. Program ten charakteryzuje się dość wysokim poziomem wejścia oraz koniecznością wypracowania pewnych wzorców myślowych. Po początkowych trudnościach udało się jednak zaimplementować wszystkie założone modele. Dzięki silnikowi dostarczanemu przez Unity możliwe jest zbudowanie zaawansowanej aplikacji bez konieczności pisania większych ilości kodu. Największą część autorskiej logiki pochłonął rzut kulą. Mechanizm rzutu w związku z ilością parametrów które są mu przypisywane okazał się o wiele bardziej skomplikowany niż pierwotnie zakładano. Skomplikowanym też okazało się zaprojektowanie maszyny stanów która kontroluje sekwencję czynności: rzut, zbicie kręgli, ustawienie kręgli, ustawienie kuli. Projekt można zaliczyć do udanych, większości xxxxzałożonych funkcjonalności została uruchomiona.