

Koszęcin, dnia 31.03.2020

Laboratorium ZTPGK  
**Fizyka w grach  
komputerowych**

Karol Kozuch  
AEI Informatyka, IGT SII  
Semestr: 1  
E-mail: [karokoz247@student.polsl.pl](mailto:karokoz247@student.polsl.pl)

## 1. Cel laboratorium

Celem laboratorium jest stworzenie sceny, która wykorzystuje elementy silnika fizycznego Unity. Postanowiono stworzyć kilka celów składających się z segmentów, do których potencjalny użytkownik będzie mógł strzelać.

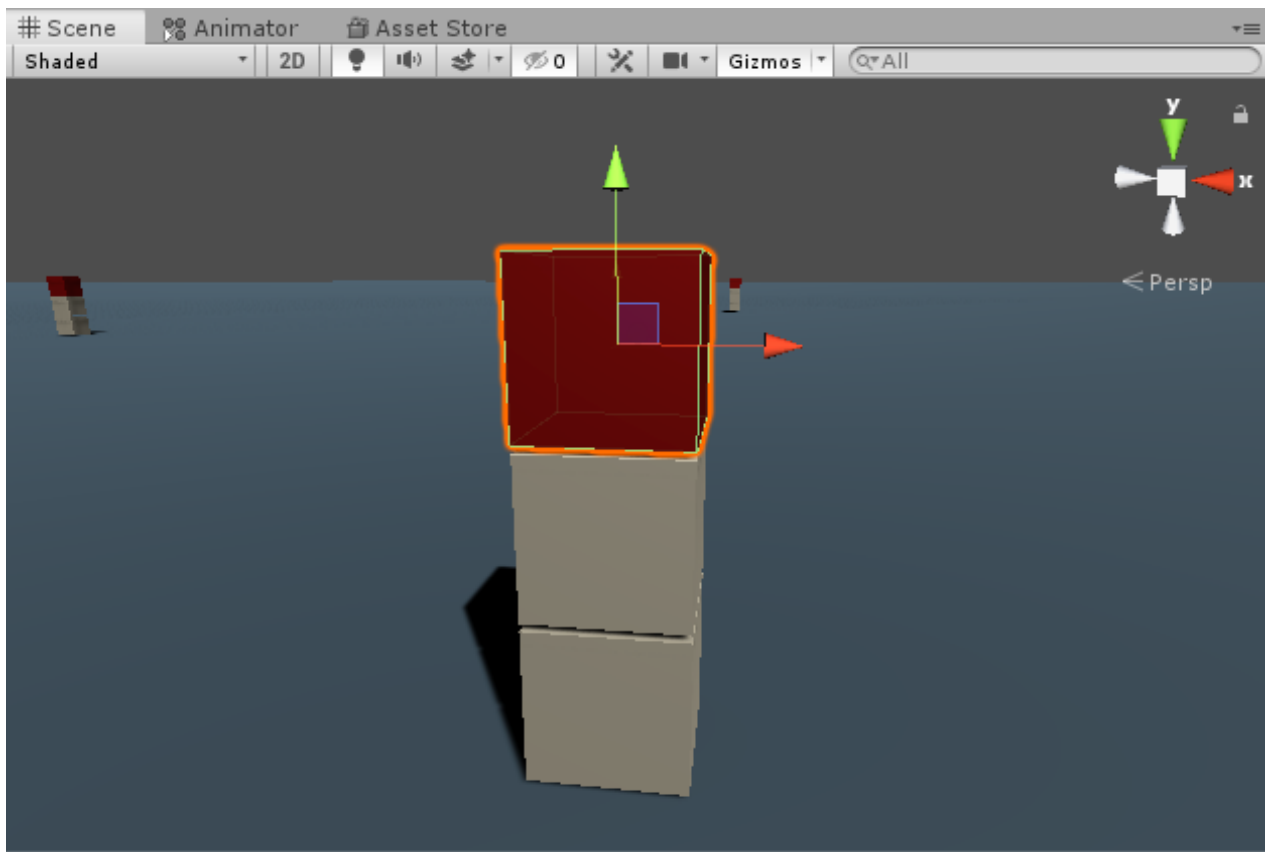
## 2. Tworzenie obiektów

Pierwszą rzeczą, którą stworzono, były cele. W zamierzeniu każdy cel składać się ma z segmentów połączonych odpowiednimi zawiasami lub sprężynami, gdzie każde złączenie może wytrzymać pewną maksymalną siłę chwilową, która na to złączenie oddziałuje. Dzięki temu przy trafieniu bezpośrednim dany cel może rozpaść się na części.

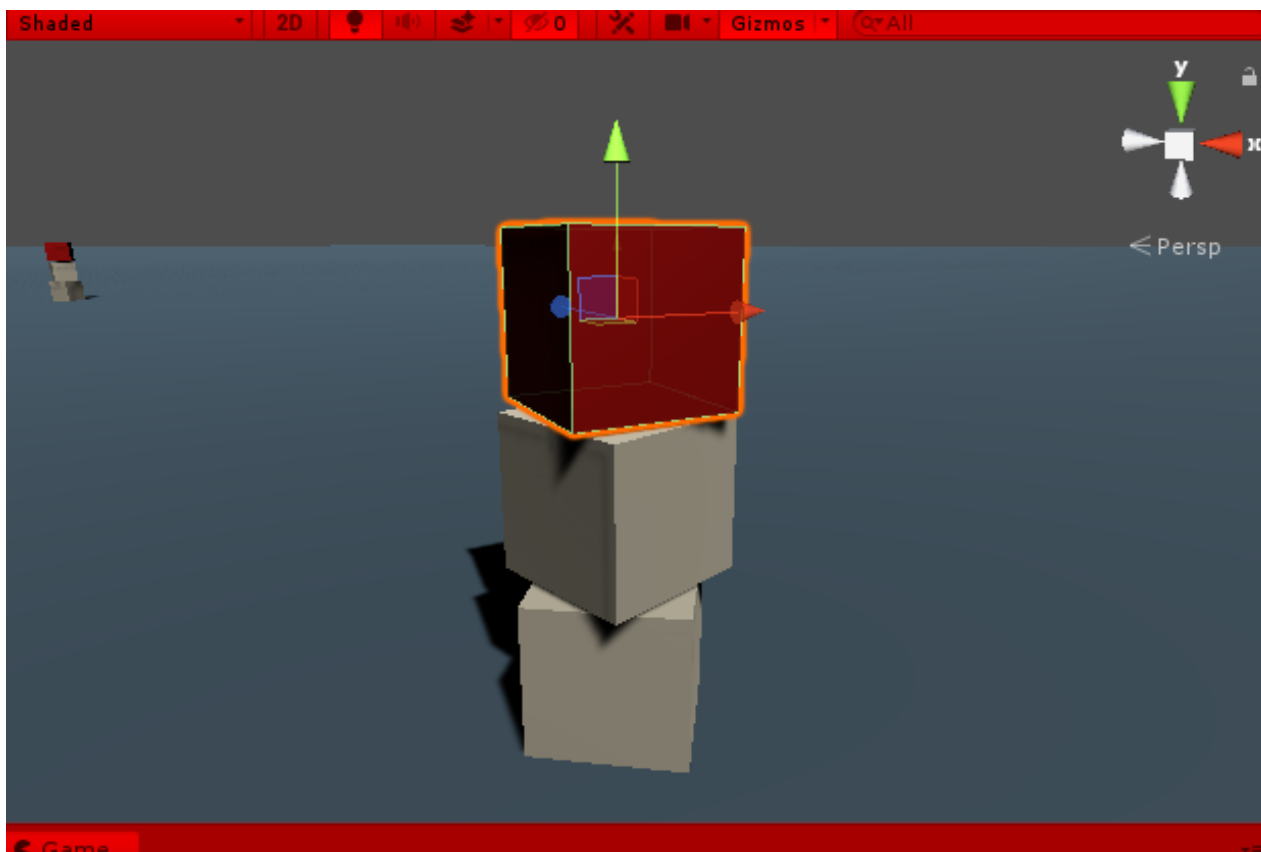
### 2.1. Cel stojący

Pierwszym stworzonym typem celu jest cel stojący. Składa się on z 3 sześciątów – podstawy, środka i głowy. Głowa pokolorowana jest na czerwono (dla widoczności) i połączona ze środkiem komponentem *Hinge Joint*. Środek połączony jest natomiast z podstawą przy pomocy tego samego komponentu. Każdy z zawiasów może znieść maksymalną siłę chwilową o wartości 1000 jednostek. Dla większej widoczności, głowa stale obraca się w jedną, a środek w drugą stronę wkoło lokalnej osi Y danego komponentu. Rysunek 1 przedstawia stojący cel, natomiast rysunek 2 prezentuje cel w trakcie działania symulacji. Skrypt obracający ma 3 parametry kontrolne:

- Wektor siły (znormalizowany) – definiuje w którym kierunku ma działać siła,
- Wartość siły – długość wektora siły,



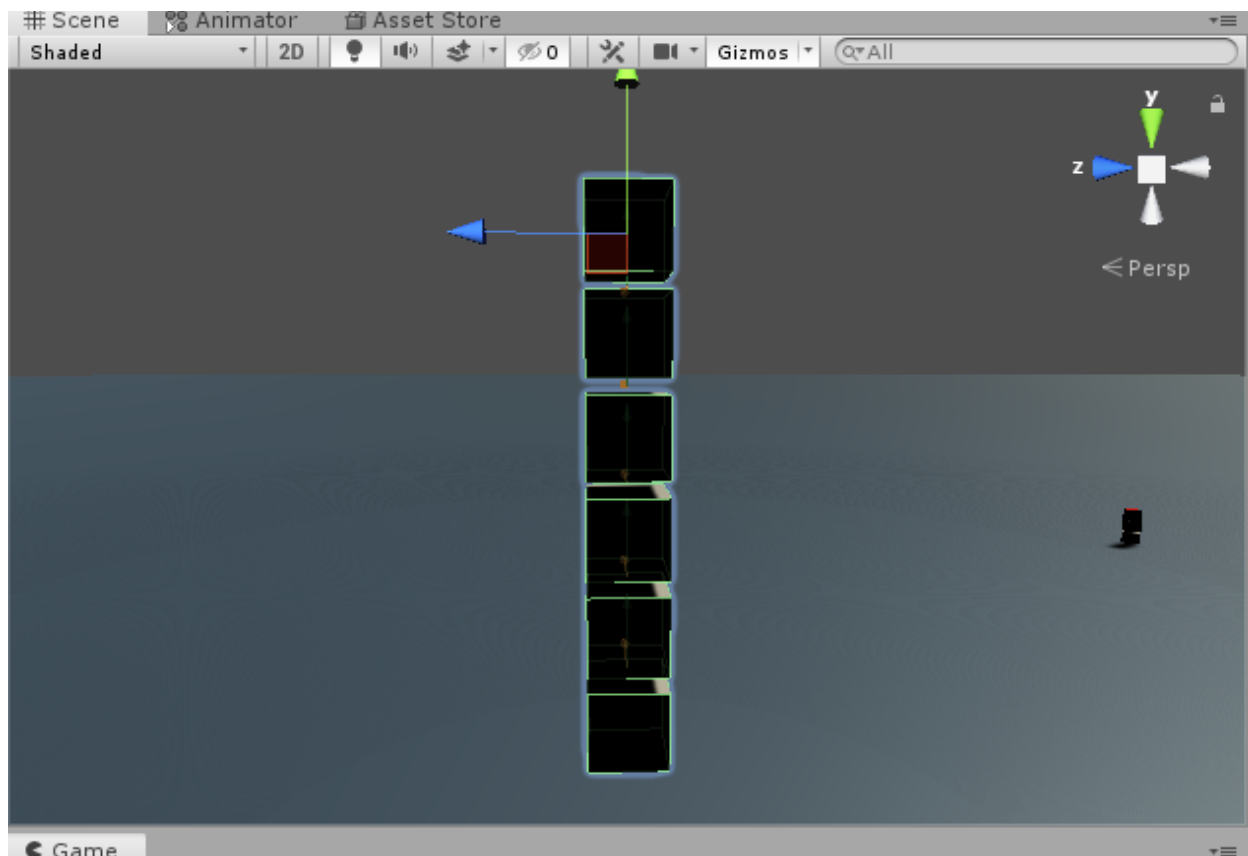
Rys. 1 – Cel stojący. Poszczególne segmenty połączone są komponentami typu Hinge Joint.



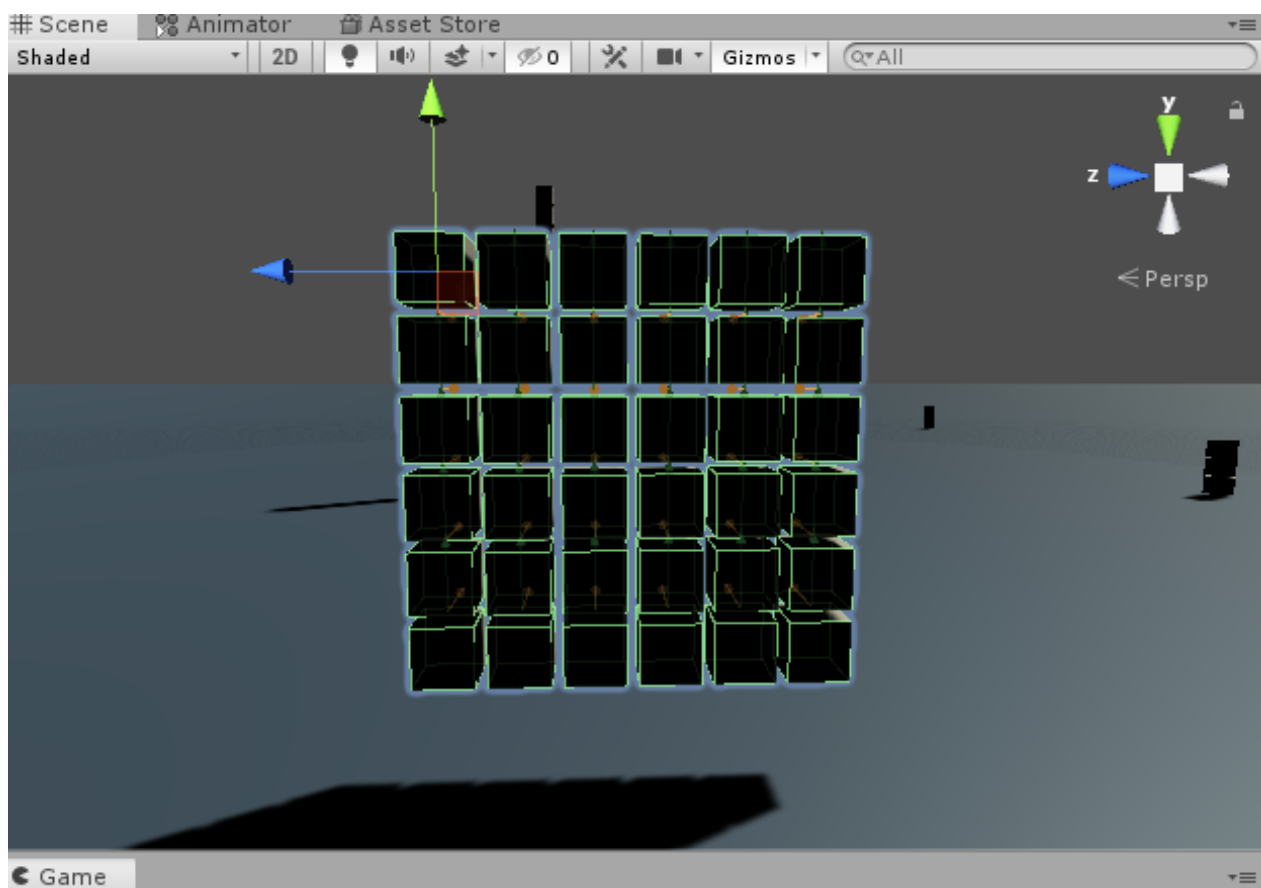
Rys. 2 – Cel stojący w trakcie symulacji. Górna część obraca się w jedną stronę, środkowa – w drugą, względem osi Y (zielona strzałka).

## 2.2. Łańcuch, Kurtyna

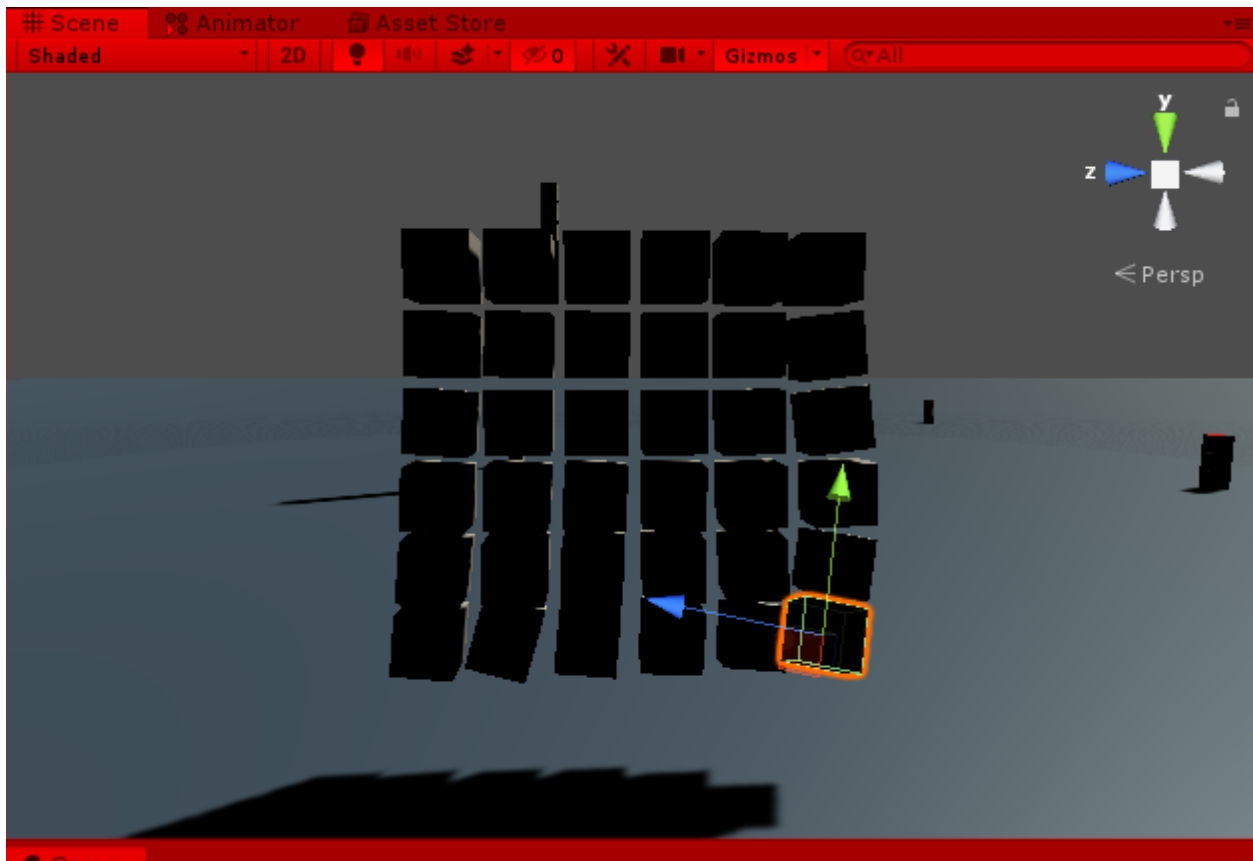
W następnej kolejności stworzono cel bazujący na łańcuchu. Postanowiono, że pojedynczy łańcuch składać się będzie z 6 sześciąt. Elementy łańcucha połączone są komponentem *Configurable Joint*, biorąc pod uwagę że elementy ruszać się mogą we dwóch kierunkach jednocześnie (stwierdzono, że elementy łańcucha nie będą się obracać wokół własnej osi, a przynajmniej będzie to ograniczone). Użyty komponent pozwala na dokładną konfigurację zawiasu. Ograniczono ruch liniowy na wszystkich osiach układu współrzędnych, natomiast obrót został ograniczony dla osi Y. Każdy zawias wytrzymuje maksymalną wartość siły w danej chwili wynoszącą 5000 jednostek. Przekroczenie tej wartości spowoduje zerwanie łańcucha w przeciążonym zawiasie. Pierwszy segment łańcucha (umieszczony najwyżej w łańcuchu) może być ustawiony na *IsKinematic*. Umożliwi to utrzymywanie się łańcucha w powietrzu. W przypadku nastąpienia kolizji tego segmentu z obiektem przypisanym do warstwy *Bullets* nastąpi przełączenie flagi *isKinematic* na *false*, co umożliwi upadek łańcucha powiązanego z segmentem. Następnie posłużono się utworzonym prefabem łańcucha i utworzono przy jego pomocy prefab kurtyny. Kurtyna ma rozmiar 6x6, stworzona jest z sześciu osobnych łańcuchów. Każdy najwyższy segment łańcucha ma ustawioną flagę *IsKinematic*, co powoduje, że kurtyna domyślnie wisi w powietrzu. Łańcuch przedstawiono na rysunku 3, natomiast kurtynę – na rysunku 4. Rysunek 5 przedstawia kurtynę wprawioną w ruch przez poruszenie skrajnego prawego dolnego sześciątka.



Rys. 3 – Łańcuch. Połączenia stanowią komponenty *Configurable Joint*.



Rys. 4 – Kurtyna 6x6. Tworzy ją 6 łańcuchów widocznych na rysunku 3.



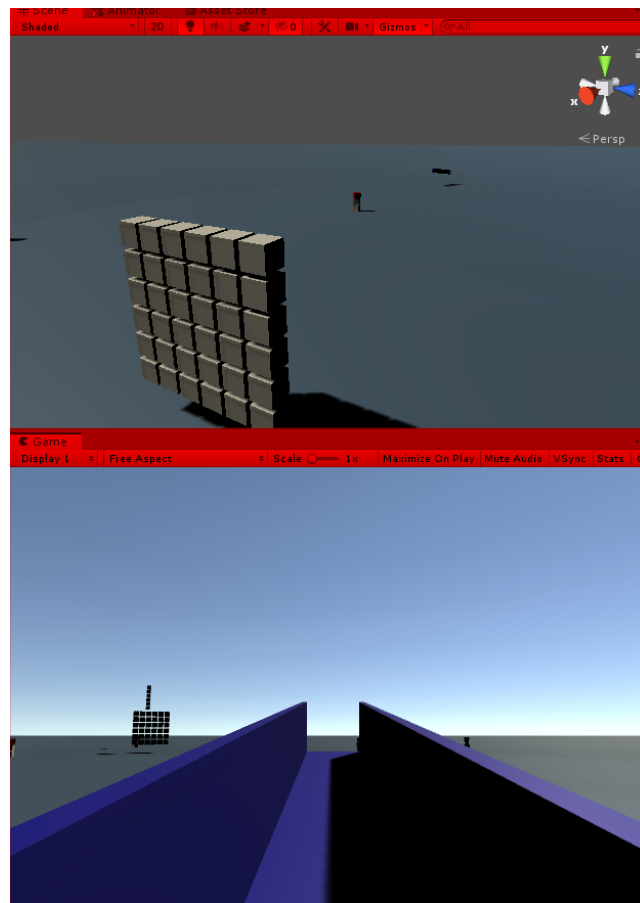
Rys. 5 – Kurtyna w trakcie symulacji.

### 3. Gracz

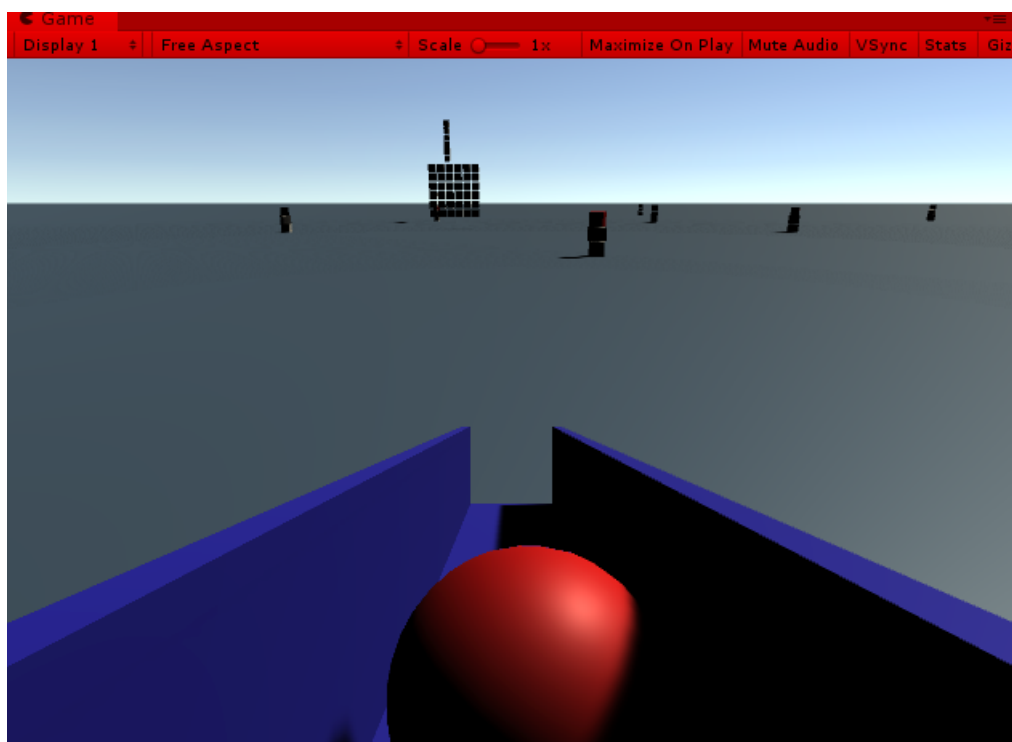
Gracz steruje obiektem działa. Działo jest w stanie wystrzelić naraz pojedynczy pocisk w postaci kuli. Obsługę działa umożliwiają następujące przyciski:

- A – obrót w lewo
- D – obrót w prawo
- W – obrót w górę
- S – obrót w dół
- R – załadowanie nowego pocisku
- Spacja – wystrzał

Działo zawieszone jest w powietrzu i może się tylko obracać. Przy załadowaniu działa (klawisz R), następuje podpięcie komponentu *SpringJoint* do obiektu tworzącego pociski. Komponent ten następnie ustawiany jest tak, by odnosił się do komponentu *Rigidbody* pocisku – zapobiega to staczaniu się pocisku w przypadku, gdy gracz celuje w dół. W momencie strzału na pocisk działa chwilowa siła o wartości 100 000 jednostek, a sprężyna jest odczepiana od pocisku. Wektor siły obracany jest przy użyciu kwaternionów obrotu działa w poziomie oraz obrotu w pionie (wartości te są odczytywane z różnych obiektów by nie ingerowały ze sobą bezpośrednio – w takim wypadku rotacja następowała także na osi X, gdy wina tylko na osiach Y i Z). Działo ukazane jest na rysunkach 6 oraz 7 (niezaładowane oraz załadowane, w tej samej kolejności). Na rysunku 8 widoczny jest pocisk tuż po zderzeniu z kurtyną. Na rysunku 9 ukazano cel stojący tuż przed trafieniem, a na rysunku 10 – tuż po trafieniu.



Rys. 6 – Niezaładowane działo.



Rys. 7 – Działo gotowe do strzału. Sprężyna utrzymuje pocisk i nie pozwala mu zsunąć się z działła.



Rys. 8 – Wystrzelony pocisk przelatujący przez kurtynę. Chwilowa siła, którą pocisk wygenerował na łączących segmenty zawiasach, spowodowała ich zerwanie i przełamanie łańcucha w kilku miejscach.



Rys. 9 – Cel stojący tuż przed trafieniem. Jeżeli chwilowa siła działająca na zawiasy między segmentami będzie większa niż 1000 jednostek – cel rozpadnie się.



Rys. 10 – Cel stojący tuż po trafieniu pociskiem. Siła chwilowa wywarta na zawiasy była na tyle duża by rozerwać cały obiekt na obiekty składowe.

#### **4. Użyte dodatkowe oprogramowanie**

Ze względu na szybkie poruszanie się pocisków użyto programu bandicam do nagrania krótkich filmów. Filmy te zostały użyte w celu uchwycenia odpowiedniej klatki prezentującej określoną funkcjonalność. Na rysunkach uzyskanych w ten sposób widnieje napis „[www.Bandicam.com](http://www.Bandicam.com)”, pod górną krawędzią.

#### **5. Link do repozytorium**

Repozytorium Github z projektem znajduje się pod wskazanym adresem:  
<https://github.com/Hornster/ZTPGK-Fiz>