



Lekcja 3. Extreme Track

Cel lekcji

Celem lekcji będzie stworzenie toru przeszkód. Stworzycie dzisiaj śmiertelne przeszkody oraz niebezpieczne taśmociągi. Wykorzystamy do tego specjalnie przygotowany system checkpointów z efektowną metą! Na koniec każdy z was będzie mógł udostępnić swoją grę do świata Robloxa.

Ważne informacje dotyczące przebiegu lekcji

Tempo lekcji

Pamiętaj, lekcja to nie są wyścigi, jeśli podczas zajęć odносimy wrażenie że może zabraknąć nam czasu na realizację całości materiału to nie przyspieszamy tempa prowadzenia zajęć żeby za wszelką cenę zrealizować każdy punkt lekcji. Najważniejsze jest aby uczestnik zrozumiał materiał, nowe zagadnienia, które pojawiają się na danej lekcji. W zależności od prowadzonego kursu: pominięcie zadania, uproszczenie projektu, zrezygnowanie z jakiejś funkcjonalności będzie lepszym rozwiązaniem niż narzucanie szybkiego tempa i chaotycznego sposobu prowadzenia lekcji.

Osoby spóźnione

Lekcje zaczynamy punktualnie, nie czekamy 10 minut na spóźnialskich. Osoby, które dotrą na zajęcia spóźnione staramy się sprawnie wprowadzić w temat i cel danej lekcji. W zależności od czasu spóźnienia, osobę taką wprowadzamy poprzez krótką rozmowę lub udostępnienie projektu z aktualnym stanem.

Początek i podsumowanie zajęć

Na początku lekcji powinno odbyć się przypomnienie materiału z poprzednich zajęć, należy zadać pytania, przypomnieć uczestnikom zagadnienia, popytać o projekt, jego działanie. Na koniec lekcji podsumowujemy ją. Bazujemy na

pytaniach podanych w konspekcie, ale oczywiście możemy też zadać dodatkowe, własne dostosowane do danej grupy.

Agenda

1. Pytania początkowe i prezentacja projektu
2. Pobranie pliku z przygotowanym systemem checkpointów i uruchomienie
3. Stworzenie podstawowych przeszkód z wykorzystaniem dostępnych bloków
4. Stworzenie śmiercionośnych przeszkód
5. Stworzenie niebezpiecznych taśmociągów przy użyciu platform
6. Zadania dodatkowe – kratownica, 180, skok wiary
7. Udostępnienie gry
8. Pytania końcowe i podsumowanie lekcji

Konieczne do wykonania przed zajęciami

Przed rozpoczęciem zajęć pamiętaj o tym aby zapoznać się z instrukcją instalacji i obsługi programów. Sprawdź również jakie są najczęstsze problemy w działaniu środowiska. Dzięki temu, gdy uczestnik ma problem możesz zaproponować szybkie rozwiązanie. Link do wszelkich potrzebnych elementów znajdziesz poniżej:

<https://drive.google.com/drive/folders/1vaEMCR5IGI8NCKJgiErSM7KQrYO7IDb7>

W konspekcie znajdziesz zdania zapisane *kursywą* są to wskazówki dla Ciebie, jakie zadać pytanie, co dopowiedzieć, korzystaj z tego, parafrazuj, lekcja na pewno będzie ciekawsza.

NIE sprowadzaj się do tego że napiszesz linijkę kodu i przejdziesz dalej bez wytłumaczenia za co odpowiada, zawsze można coś dodatkowo powiedzieć. Konspekt jest, rozpisany szczegółowo, natomiast jesteś zobowiązany/a przerobić go w całości wcześniej **przed** zajęciami. Przerabiając sobie projekt z zajęć krok po kroku zgodnie z konspektem będziesz mieć pełną kontrolę, swobodę w tłumaczeniu i nic Cię nie zaskoczy podczas lekcji. Nasze konspekty nie są idealne (choć cały czas staramy się żeby takie były ☺), przerabiając je wcześniej możesz zlokalizować ewentualne błędy i je zgłosić. Czasami błędy będą wynikać z aktualizacji danej platformy więc na bieżąco musimy reagować.

1. Pytanie początkowe i prezentacja projektu

1. Jak nazywa się inaczej tor przeszkód w Robloxie? (Obby)
2. Czy graliście w Obby w Robloxie? (Jeżeli tak to, z jakimi przeszkodami się spotkaliście?)
3. Do czego służą checkpointy?

Prezentacja gotowego projektu:

[Extreme Track - gotowy.rbxl](#)

*Uruchamiamy gotowy projekt, chwilę gramy tłumacząc z jakich elementów składa się nasz projekt warto na tym etapie pokazać że na tej lekcji testując etapy będziemy często korzystać z opcji **Play Here** wtedy to gracz pojawi się od razu w miejscu gdzie będzie ustawiona kamera, i dzięki temu możemy testować dany konkretny etap a nie przechodzić cały tor.*

Gra zaczyna się na elemencie startowym.



Po wejściu na checkpointa, obrazek mignie kilka razy, to oznacza że została zapisana pozycja gracza, gdy gracz zginie, odrodzi się na ostatnim checkpointie.



Po wejściu na metę zostanie zagrana krótka muzyka z wystrzeleniem fajerwerków!



2. Pobranie pliku z przygotowanym systemem checkpointów i uruchomienie

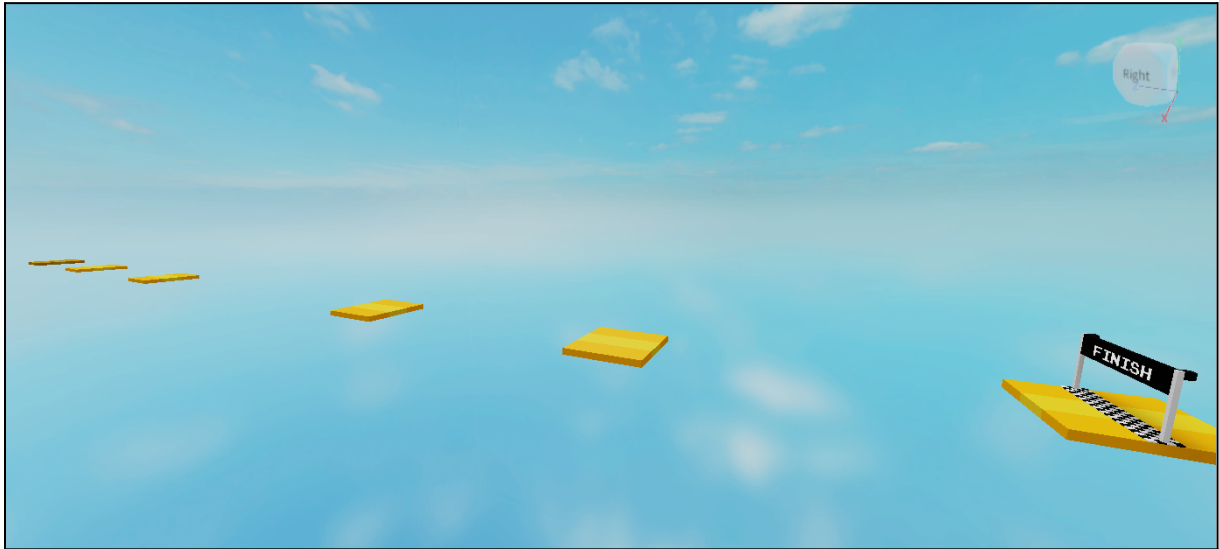
Na początku pobieramy plik z przygotowanymi checkpointami oraz meta.

Link: <https://drive.google.com/file/d/1zZt269u5Sp7LgQSm9tgNh1GqsfSv3616>

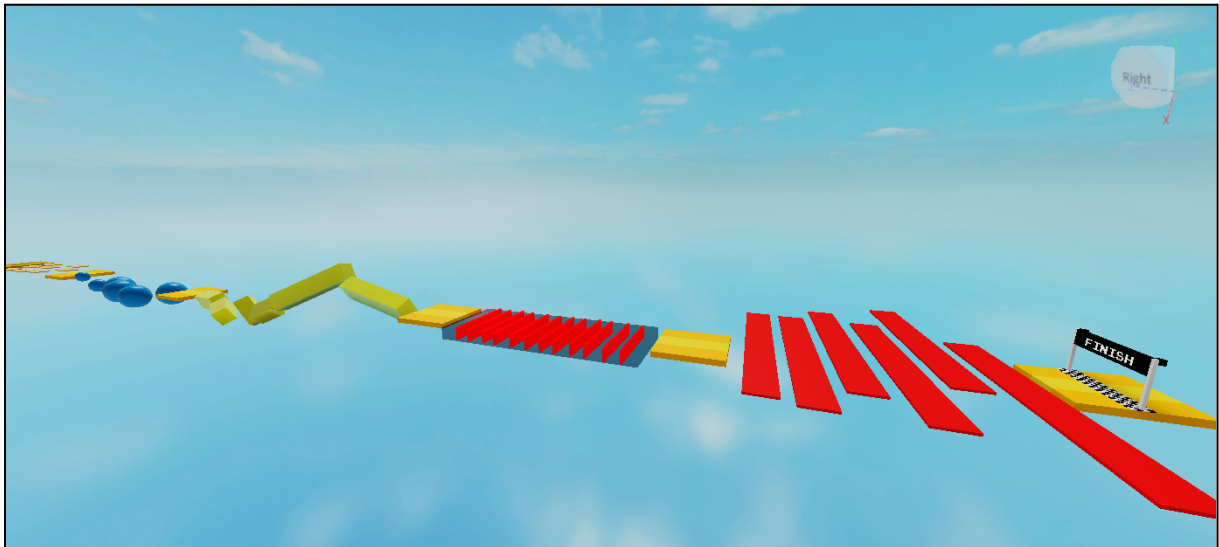
Skrócony link: <https://cutt.ly/TAhPBma>

Naszym zadaniem jest zbudowanie poszczególnych etapów do toru przeszkód.

Mapka bez etapów.



Mapka ze zbudowanymi etapami.

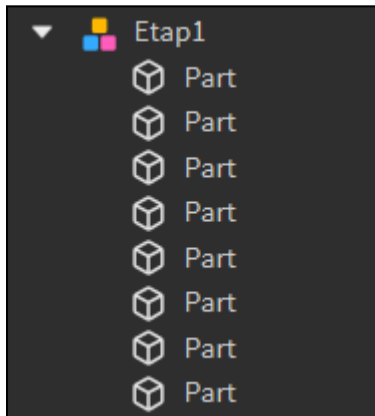


3. Stworzenie podstawowych przeszkód z wykorzystaniem dostępnych bloków

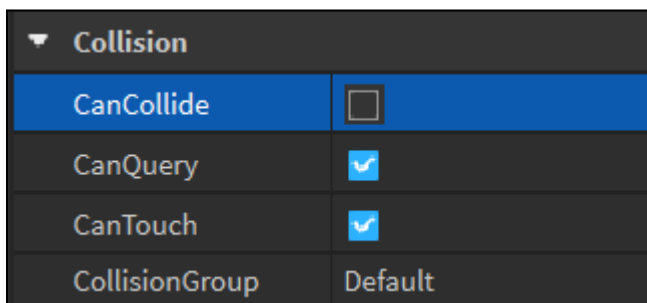
Pierwsze trzy etapy mogą składać się z podstawowych bloków, takich jak:

- Blok (Block)
- Kula (Sphere)
- Klin (Wedge)
- Cylinder (Cylinder)

Przy budowaniu poszczególnych etapów należy zwrócić uwagę na porządek, aby każdy etap był zgrupowany w jeden model.



Niektóre bloki mogą być pułapką, przez którą spadniemy, możemy to zrobić za pomocą zmiany właściwości **CanCollide** na odznaczoną. **Należy również przypomnieć uczestnikom o zaznaczeniu **Anchor**, aby bloki nie spadały.**



Czas respawnu

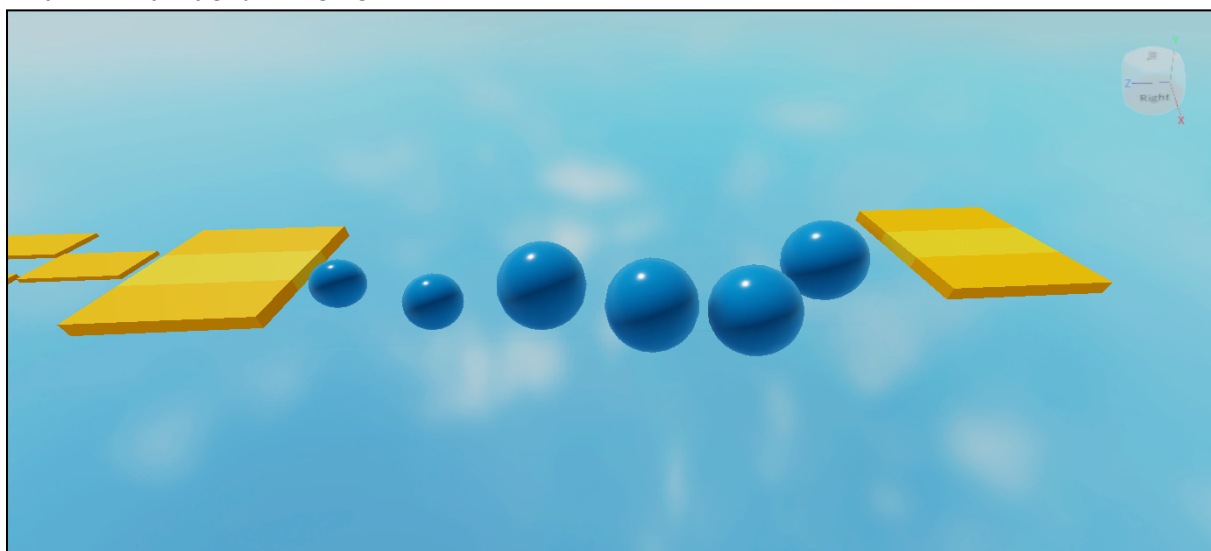
Jako, że nasz tor zawieszony jest w przestrzeni, gracz spadając jest odradzany kiedy przekroczy pewien pułap wysokości. Możemy wyregulować sobie tą wysokość (domyślna wartość ustawiona to -500) z pomocą właściwości **FallenPartsDestroyHeight**, którą znajdziemy we właściwościach **Workspace**. Proponowana wartość **-100**.

Pierwszy etap należy wykonać wspólnie z uczestnikami pokazując wszystkie kluczowe elementy: takie jak kotwica, grupowanie. Na 2 i 3 etap warto dać uczestnikom czas, żeby samodzielnie je wykonywali ponieważ zazwyczaj uczestnicy mają własne super pomysły na przeszkody. Gdyby ktoś jednak nie miał to inspirujemy go naszymi przykładowymi etapami. Etapy 4 i 5 wykonujemy wspólnie z uczestnikami

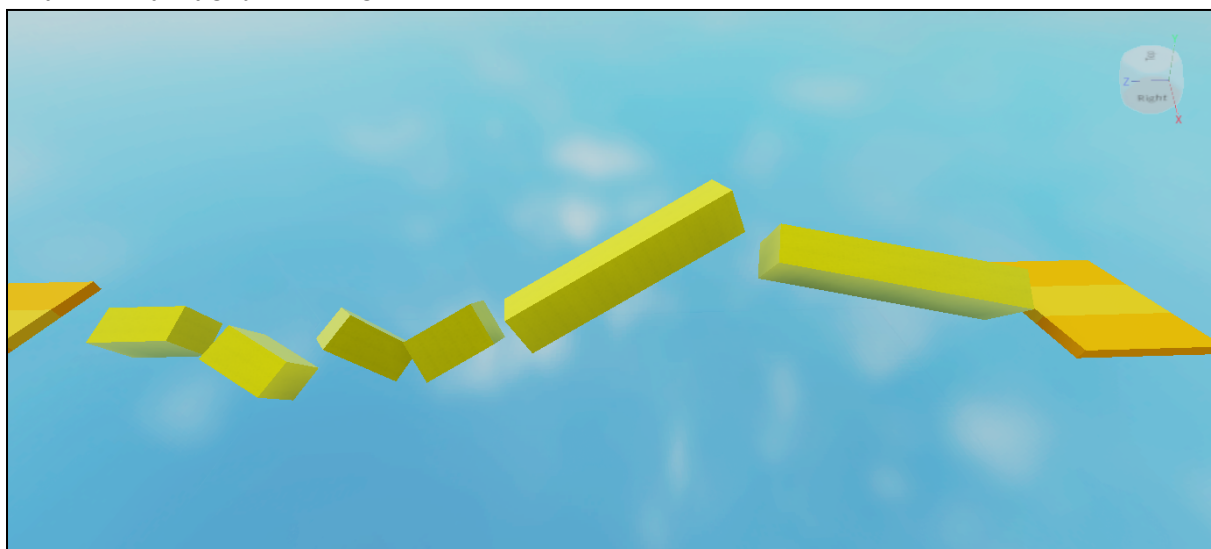
Przykładowy wygląd pierwszego etapu:



Przykładowy wygląd drugiego etapu:



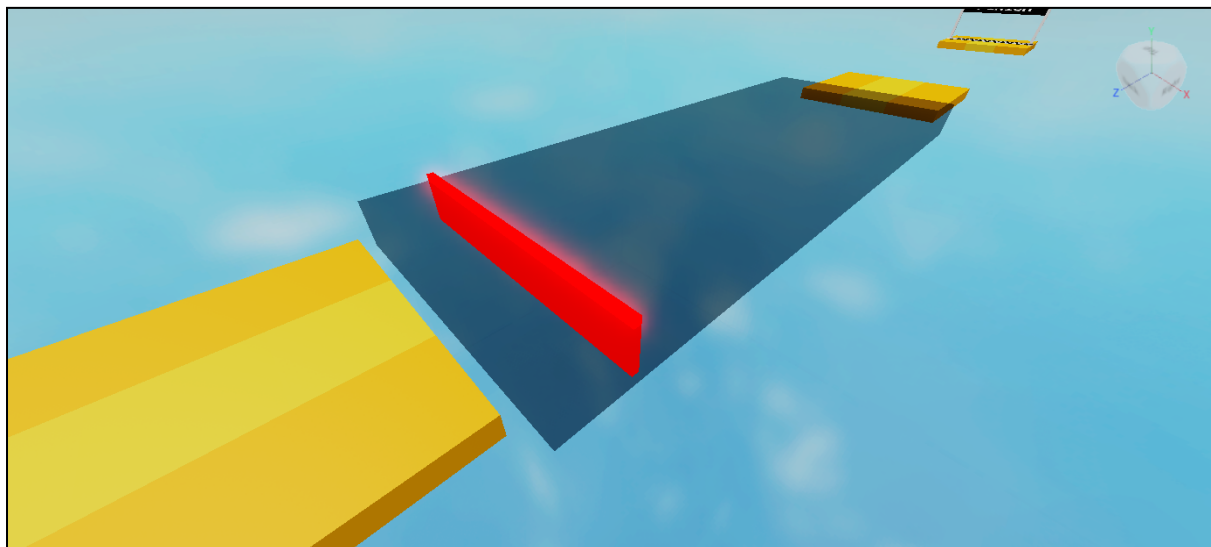
Przykładowy wygląd trzeciego etapu:



4. Stworzenie śmiertelnych przeszkód

Przyszła pora na stworzenie śmiertelnych przeszkód, nie będziemy mogli ich dotknąć, ponieważ od razu spowodują, że nasza postać się rozpadnie i będziemy musieli zaczynać od ostatniego checkpointu.

Budujemy na początku platformę, po której będzie chodził gracz, następnie tworzymy przeszkodę w postaci czerwonego bloczka, możemy użyć materiału neon.



Przed przystąpieniem do pisania kodu sprawdzimy w praktyce co należy zmienić, aby nasza postać straciła zdrowie, co w konsekwencji sprawi, że się rozpadnie.

Na początku włączamy grę i przechodzimy do okna **Explorer**.

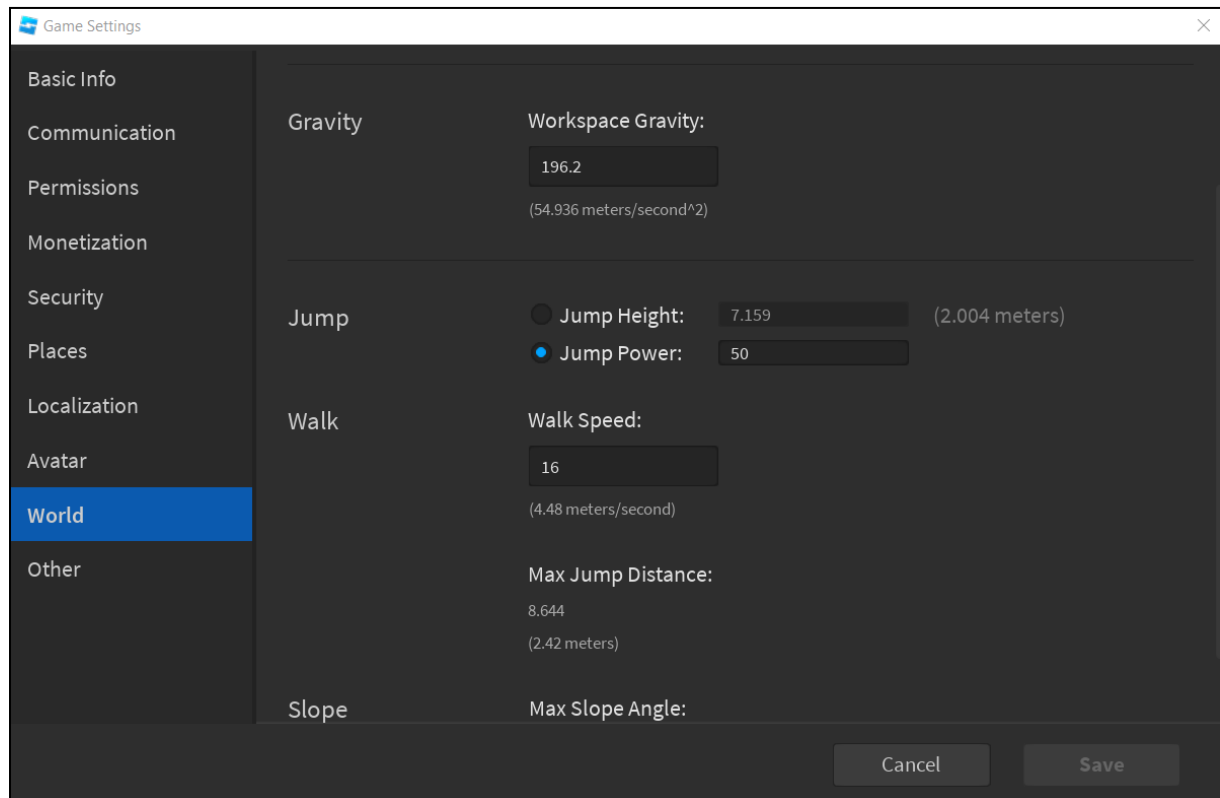
Rozwijamy **Workspace**, a następnie szukamy modelu z nazwą naszej postaci, kolejno w nim zaznaczamy element o nazwie **Humanoid** i przechodzimy do okna **Properties** do właściwości **Health**.

*Kilka słów o elemencie **Humanoid**. Humanoid to tak naprawdę specjalny obiekt, który nadaje modelom funkcjonalność postaci. Daje możliwość modelowi chodzenia oraz interakcji z różnymi elementami.*



Ćwiczenie do wykonania:

Zacznij od zmiany właściwości **Health** na 0, postać powinna się rozpaść, to właśnie będziemy wykorzystywać w programie, który będzie znajdował się w śmiertelnych przeszkodach. Dodatkowo możecie wspólnie również zmienić wartości właściwości **WalkSpeed** oraz **JumpPower**. Aby zmienić je na stałe należy przejść do **File > Game Settings...** (jeśli projekt nie jest zapisany do Robloxa to najpierw trzeba potwierdzić zapis) a następnie do zakładki **World**.



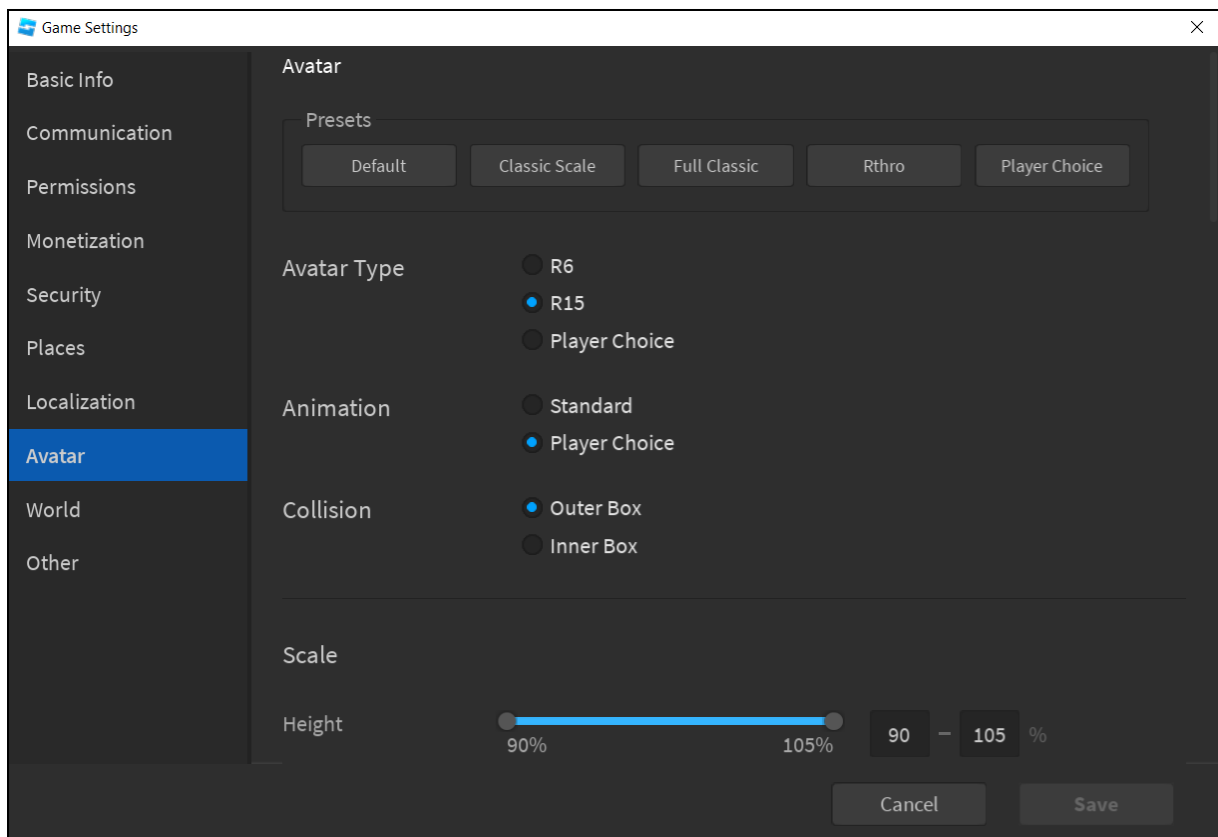
Ciekawostka:

Wiedziecie, że w Robloxie występują dwa rodzaje typów postaci R6 i R15? Czy ktoś wie co te skróty oznaczają?

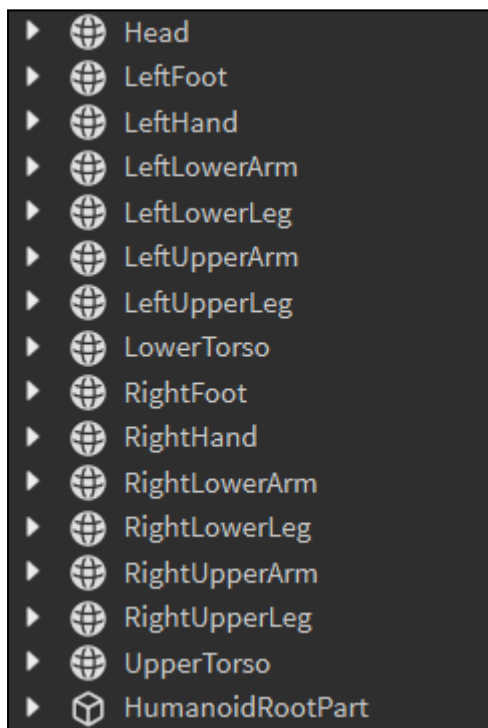
Odpowiedź:

R6 oraz **R15** to tak naprawdę określenie ilości części kończyn postaci, R6 wykorzystuje 6 części, jest to prostszy model, R15 wykorzystuje 15 części jest on bardziej złożony.

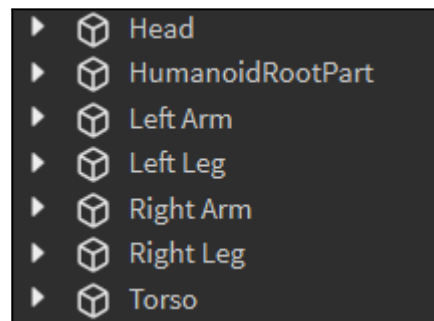
Domyślnie w **Game Settings...** ustawiona jest opcja **Avatar Type** na R15, aby sprawdzić prostszy model należy zaznaczyć opcję R6, sprawdź jak zmieniła się budowa postaci oraz jak teraz wyglądają jego podstawowe animacje ruchu oraz skoku.



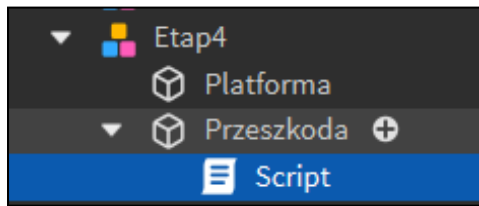
R15



R6



Aby bloczek stał się śmiertelny, należy dodać do niego **Script**, a następnie napisać specjalny program.



Dzisiaj w programie użyjemy **instrukcji warunkowej**.

Czy ktoś z was wie co to jest instrukcja warunkowa i do czego służy?

Instrukcja warunkowa sprawdza czy dany warunek jest prawdziwy, jeżeli tak to wykonuje się wybrany program. Przykładowo jeżeli zbierzemy wystarczającą ilość gotówki w grze, będziemy mogli kupić swoje wymarzone auto. W dzisiejszym przykładzie programu wykorzystamy instrukcję warunkową do sprawdzenia czy przeszkoda została dotknięta przez naszego ludzika, ale zanim przejdziemy do pisania programu poznamy podstawową budowę instrukcji warunkowej.

Budowa podstawowej instrukcji warunkowej wygląda następująco (możemy zaprezentować na zwykłym skrypcie w Workspace):

Wersja podstawowa:

```
warunek = true

if warunek then
    print("Warunek jest prawdziwy - wykonaj pierwszy program")
end
```

Wersja z możliwością wykonania drugiej opcji w momencie fałszywego warunku:

```
warunek = false

if warunek then
    print("Warunek jest prawdziwy - wykonaj pierwszy program")
else
    print("Warunek jest fałszywy - wykonaj drugi program")
end
```

Wersja ze sprawdzeniem dwóch warunków:

```
warunek1 = false
warunek2 = true

if warunek1 then
    print("warunek1 jest prawdziwy - wykonaj pierwszy program")
elseif warunek2 then
    print("warunek2 jest prawdziwy - wykonaj drugi program")
else
    print("warunek1 oraz warunek2 są fałszywe - brak możliwości wykonania programów")
end
```

Przykładowe warunki: **Czy to prawda czy fałsz?** (pytamy uczniów)

```
print(4 > 6)
print(2 < 3)
print(4 == 4)
print(4 ~= 4)
print(5 >= 2)
print(6 <= 6)
```

Oprócz sprawdzania liczb możemy sprawdzać również nazwy:

```
local nick = "PrzemekV5" -- w tym miejscu niech każdy wpisze swój nick

if nick == "PrzemekV5" then -- w tym miejscu niech każdy wpisze swój nick
    print("Akceptacja - użytkownik znajduje się w bazie danych")
else
    print("Blokada dostępu - nieznany użytkownik")
end
```

Kod programu tłumaczmy linijka po linijce, aby uczestnicy zrozumieli jak działa.

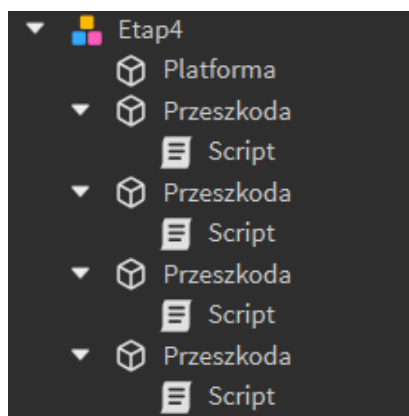
Kod programu:

```
--odwołanie do rodzica skryptu czyli dostęp do obiektu naszej przeszkody
local przeszkoda = script.Parent

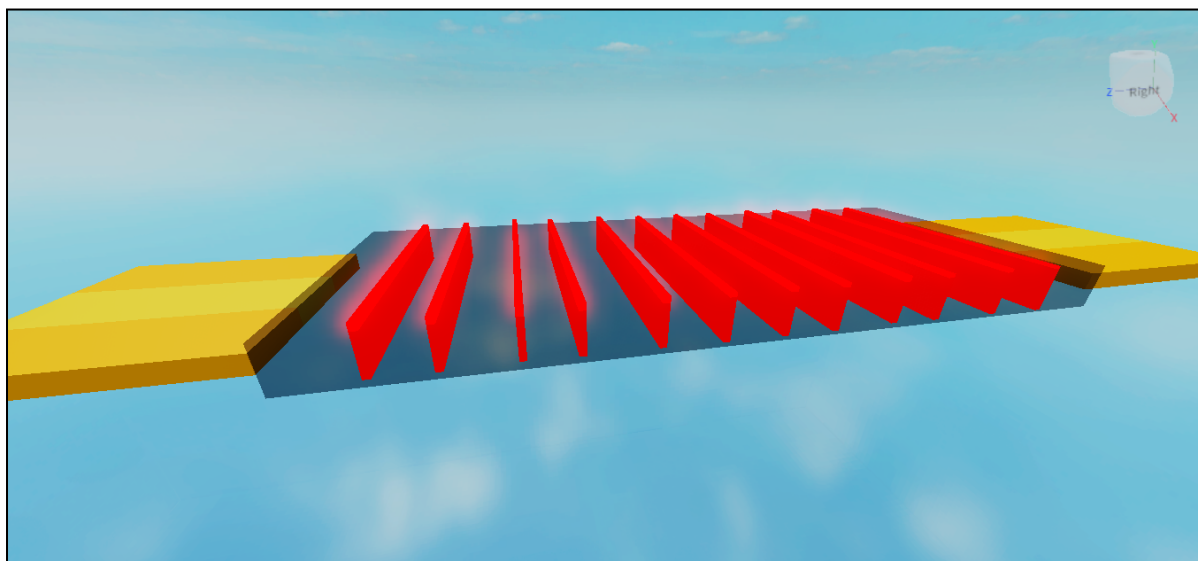
--do funkcji przekazujemy obiekt który dotknie naszej pułapki
local function Zniszcz(part)
    --gracz może dotknąć przeszkodę częścią swojego ciała,
    --zwracamy kto jest rodzicem tej części
    --poprzez part.Parent zwracamy info kto jest rodzicem tej części,
    --która dotyka przeszkodę
    --czyli uzyskamy dostęp do modelu gracza w nim szukamy obiektu Humanoid
    --Humanoid nadaje modelowi cechy ludzkie takie jak życie,
    --prędkość chodzenia, wysokość skoku, itp.
    local ludzik = part.Parent:FindFirstChild("Humanoid")
    --Jeżeli ludzik został znaleziony
    if ludzik then
        --zmień życie na 0
        ludzik.Health = 0
    else --w przeciwnym wypadku
        --poinformuj, że to nie ludzik
        print("To nie ludzik :(")
    end
end

--ustawiamy na naszej przeszkodzie zdarzenie na dotknięcie
przeszkoda.Touched:Connect(Zniszcz)
```

Po przetestowaniu jednego śmiertelniegroźnego bločka można duplikować przeszkody, kod jest uniwersalny, więc będzie działać z każdym nowym blokiem.

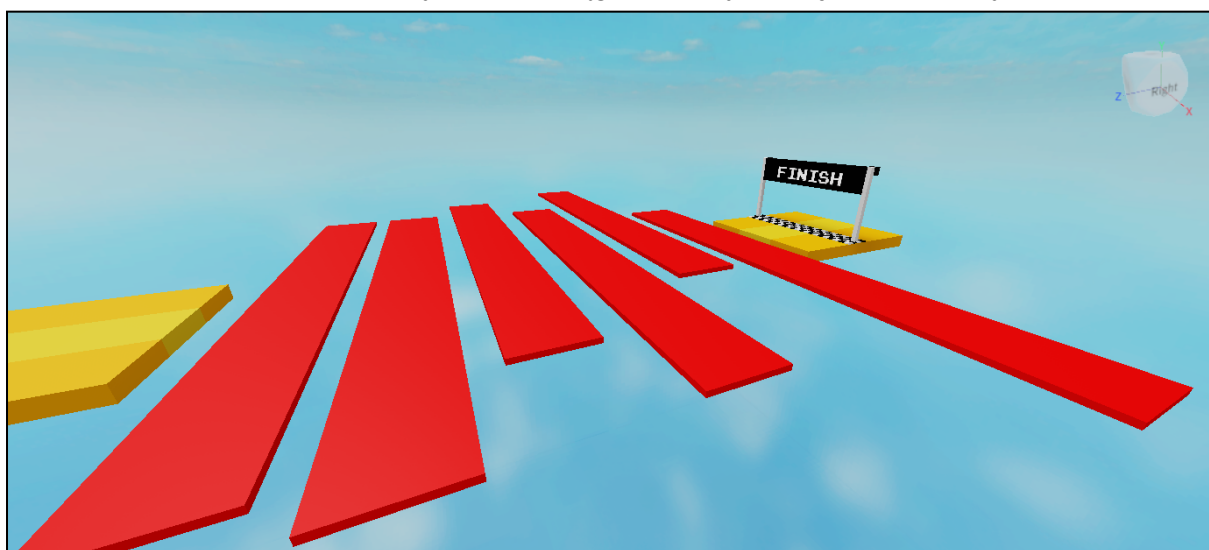


Przykładowy wygląd czwartego etapu:



5. Stworzenie niebezpiecznych taśmociągów przy użyciu platform

W celu stworzenia niebezpiecznych taśmociągów należy na najpierw stworzyć kilka platform:



Następnie, we właściwościach każdej platformy należy zmienić właściwość **AssemblyLinearVelocity**, należy pokazać uczestnikom zmianę wartości dla poszczególnych osi oraz wartości minusowe.

▼ Assembly	
▼ AssemblyLinearVelocity	15, 0, 0
X	15
Y	0
Z	0
▶ AssemblyAngularVelocity	0, 0, 0
▶ AssemblyCenterOfMass	-113.901, 3.5, -772.357
AssemblyMass	inf
AssemblyRootPart	tasmociag

Wejście na tak przygotowany blok spowoduje, że postać będzie przesuwana w danym kierunku.

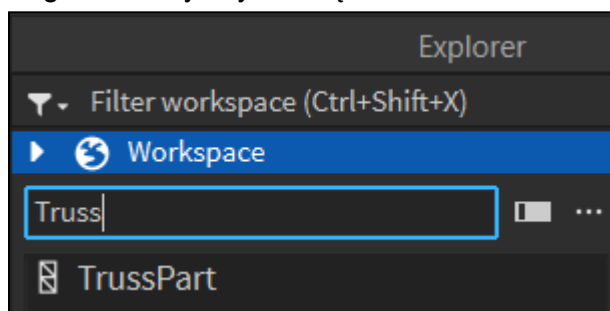
6. Zadania dodatkowe – kratownica, 180, skok wiary

Jeżeli zostało dużo czasu na zajęciach, możemy pokazać dodatkowe etapy.

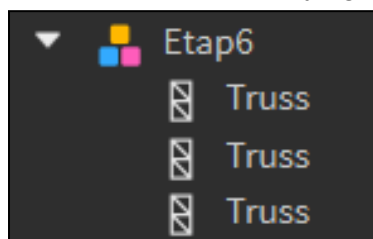
Należy pamiętać o zduplikowaniu checkpointa i zmianie jego nazwy.

Kratownica

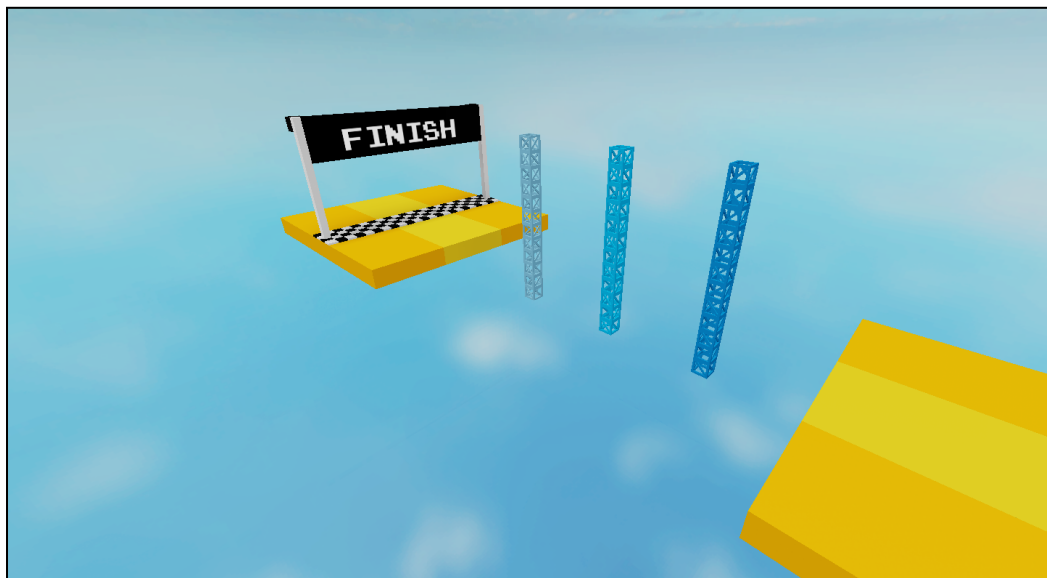
Aby stworzyć kratownicę należy dodać *TrussPart* do *Workspace*, aby zmodyfikować jej długość należy użyć narzędzia *Scale*.



Ten etap również należy zgrupować.

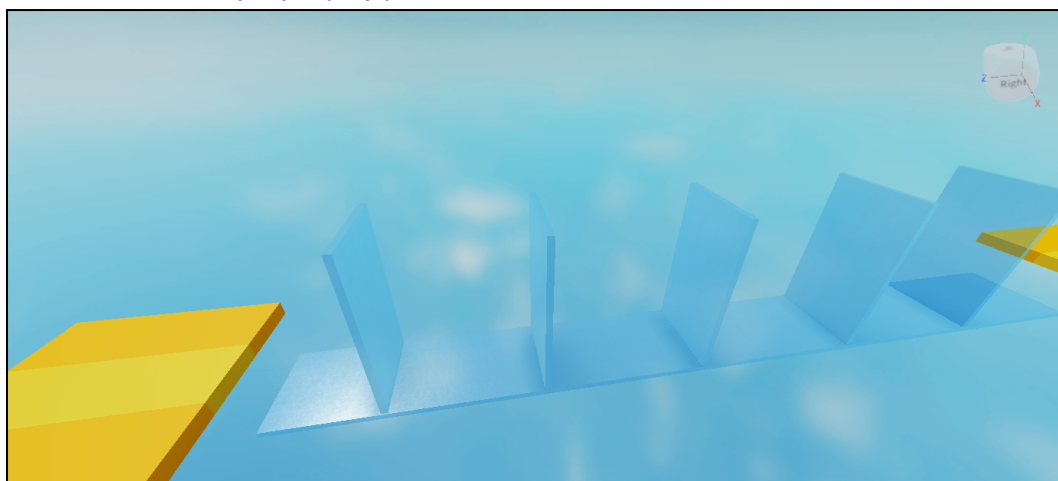


Przykładowy wygląd szóstego etapu:



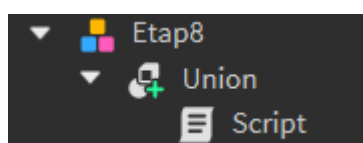
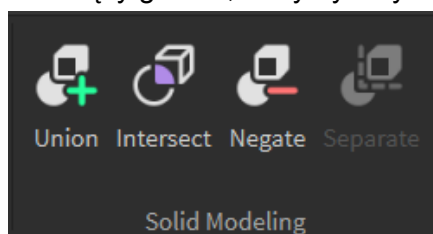
180

Etap przechodzimy wykręcając obrót w powietrzu

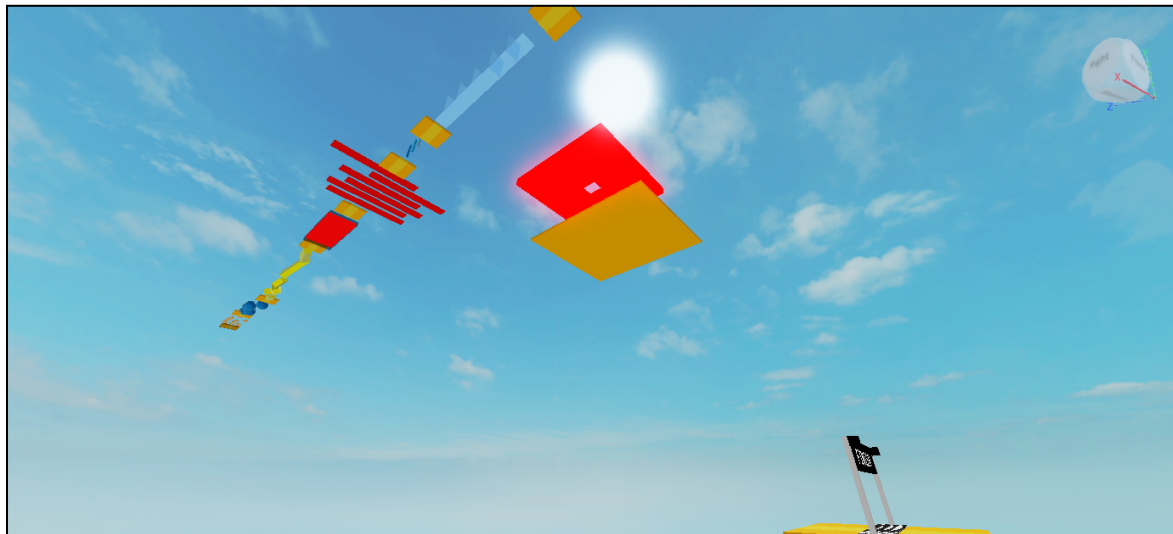


Skok wiary

Za pomocą Solid Modeling - tworzymy bryłę z małym otworem przez, który gracz będzie musiał przelecieć spadając z dużej wysokości, do tak utworzonej bryły dodajemy skrypt niszczący gracza, który był używany wcześniej.

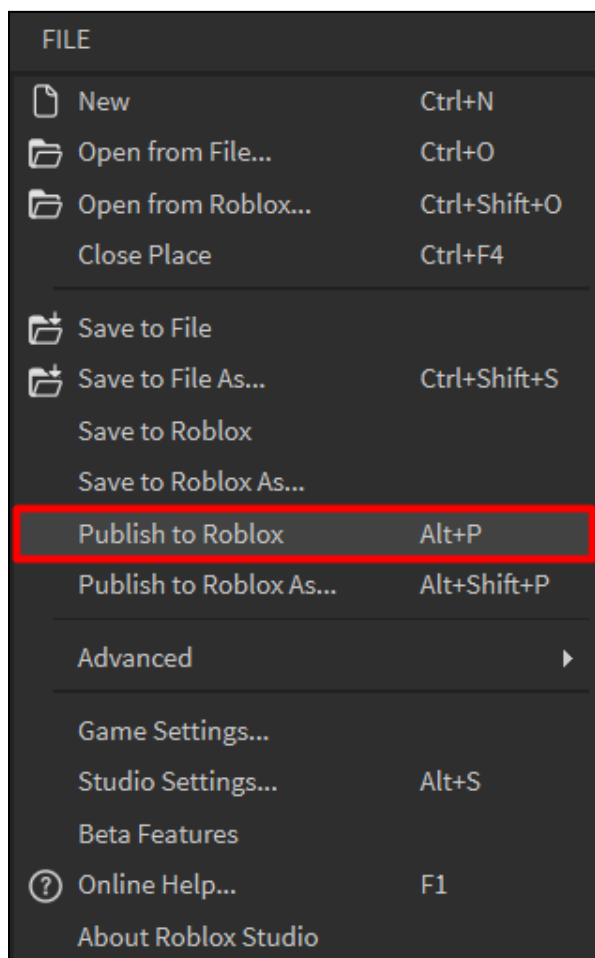


Przykładowy wygląd ósmego etapu:



7. Udostępnienie gry

Aby udostępnić grę, należy kliknąć **File** w lewym górnym rogu, a następnie kliknąć przycisk **Publish to Roblox**.



Publish Game

Basic Info

Basic Info

Name

Extreme Track

13/50

Description

0/1000

Creator

Me

Genre

All

Devices

☒ Computer

☒ Phone

☒ Tablet

☐ Console

☐ VR

Team Create

☐

Enables collaboration and autosave to cloud [Learn more](#)

Update existing game...

Cancel

Create

Po uzupełnieniu informacji na temat gry i jej nazwy klikamy **Create**, gdy gra zostanie udostępniona pomyślnie, należy przejść do **Game Settings** i zmienić dostępność gry na publiczną (**Permissions** > **Public**).

Game Settings

Basic Info

Communication

Permissions

Monetization

Security

Places

Localization

Avatar

World

Other

Permissions

Playability

Who can play this game?

☐ Friends

Friends on Roblox

☒ Public

Anyone on Roblox

☐ Private

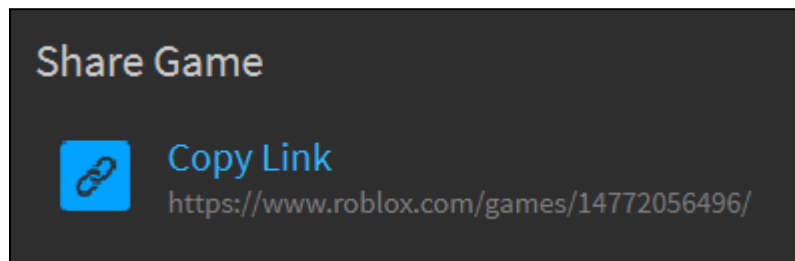
Only developers of this game

Cancel

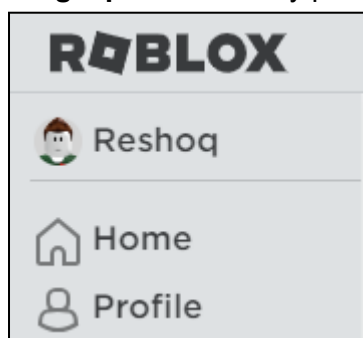
Save

Aby zagrać grę z innymi graczami należy:

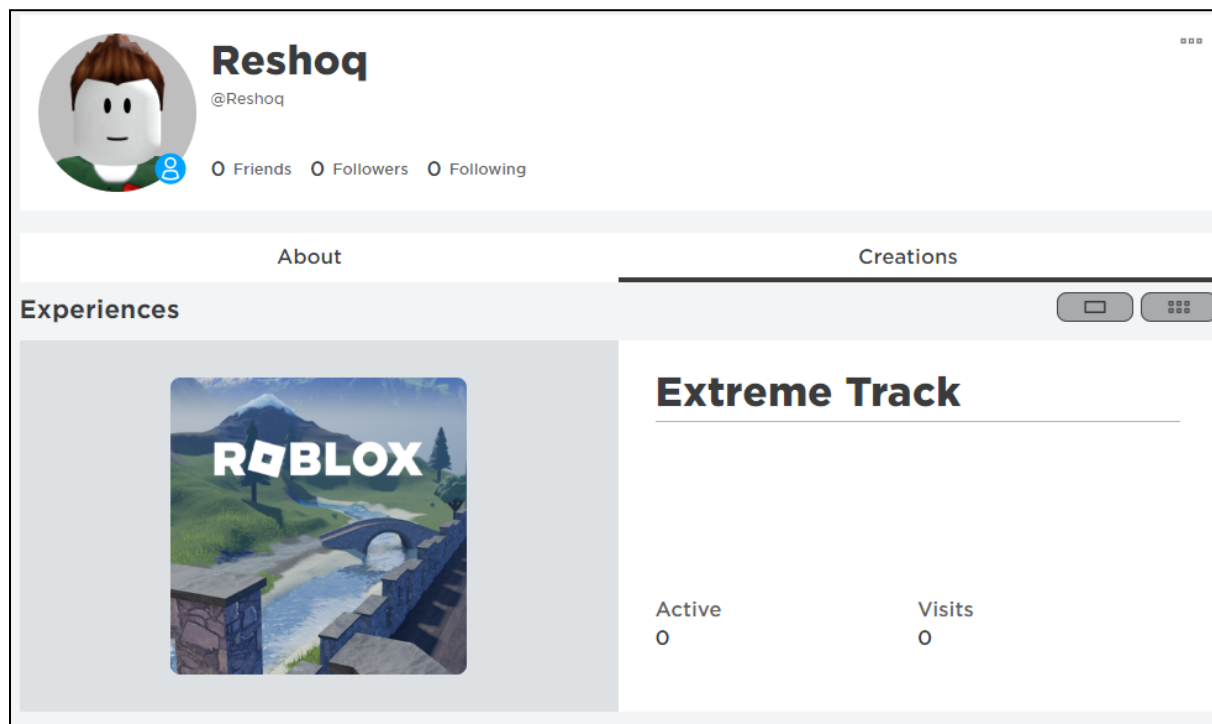
Pierwszy sposób - w prawym górnym rogu wybrać znaczek udostępniania (obok nicku) a następnie skopiować link i wysłać znajomym.



Drugi sposób - należy przejść do strony Robloxa i przejść do profilu gracza.



W sekcji **Creations** będą dostępne gry, które zostały udostępnione i są dostępne dla graczy.



8. Pytanie końcowe i podsumowanie lekcji

Pytanie końcowe:

1. Czy będziecie ulepszać stworzony przez was tor przeszkód?
2. Jak nazywa się właściwość, dzięki której możemy sprawić, żeby platforma stała się pułapką (po wejściu na nią spadamy)?
3. W jaki sposób sprawić, aby nasza gra była dostępna dla wszystkich?