

Rzeczywistość wirtualna (VR) dla każdego

Aframe i HTML 5



Krzysztof Wołk



KRZYSZTOF WOŁK

Rzeczywistość wirtualna (VR) dla każdego – Aframe i HTML 5.

VR w HTML 5 na każdym urządzeniu z Internetem

WERSJA DEMONSTRACYJNA



Wydawnictwo Psychoskok
Konin 2018

Krzysztof Wołk
„Rzeczywistość wirtualna (VR) dla każdego – Aframe i HTML 5”

Copyright © by Krzysztof Wołk, 2018
Copyright © by Wydawnictwo Psychoskok Sp. z o.o. 2018

Wszelkie prawa zastrzeżone. Żadna część niniejszej publikacji
nie
może być reprodukowana, powielana i udostępniana w
jakiegokolwiek formie bez pisemnej zgody wydawcy.

Projekt okładki: Krzysztof Wołk
Korekta: Bogusław Jusiak
Skład epub, mobi i pdf: Kamil Skitek

Krzysztof Wołk - www.wolk.pl

ISBN: 978-83-8119-295-8

Wydawnictwo Psychoskok Sp. z o.o.
ul. Spółdzielców 3, pok. 325, 62-510 Konin
tel. (63) 242 02 02, kom. 695-943-706

<http://www.psychoskok.pl/>
<http://wydawnictwo.psychoskok.pl/>
e-mail: wydawnictwo@psychoskok.pl

Spis treści

Wstęp

1. Struktura HTML

1.1. Definiowanie struktury strony

1.2. Stworzenie podstawowej struktury strony

1.3. Podstawowe elementy blokowe

1.4. Podstawowe elementy liniowe

2. Znaczniki semantyczne

2.1. Wstęp do znaczników HTML 5

Metadata kontent

Sectioning kontent

Heading kontent

Phrasing kontent

Embedded kontent

Interactive kontent

2.2. Elementy tworzące sekcje

2.3. Elementy nietworzące sekcji

2.4. HTML 5 outline

3. Cięcie techniczne – wstęp do CSS

a. Cięcie techniczne

Zadanie

b. Tworzenie dokumentu CSS

Zadanie

Zadanie

Zadanie

Zadanie

4. Elementy blokowe i opływanie

4.1. Elementy blokowe

Zadanie

4.2. Opływanie

Zadanie: Pozycjonowanie elementów w nagłówku

Zadanie: Opływanie pozostałych elementów

5. Nowe selektory i pozycjonowanie CSS

Zadanie: Stylowanie menu

Zadanie: Stylowanie aside

Zadanie: Stylowanie boksów ustawień

Zadanie: Stylowanie nowości

Zadanie: Boksy reklamowe

Zadanie: Stopka

Zadanie dodatkowe

6. Responsywny układ elementów HTML

7. Strona internetowa portfolio z formularzem kontaktowym.

8. HTML5 Pokaz Slajdów/Galeria WWW

9. Wprowadzenie do Aframe

10. Czym jest aframe

11. Rozbudowywanie HTML 5 o aframe

11.1. Panorama w wirtualnej rzeczywistości

11.2. Co udało nam się tu zrobić dzięki Aframe

12. Jak zbudować swoją pierwszą scenę w VR

- 12.1. Dodawanie tekstu
- 13. Tryb VR
- 14. Budowanie większej sceny
 - 14.1. Dodawanie jednostki
 - 14.2. Przekształcanie jednostki w 3D
 - 14.3. Umieszczanie pudełka przed kamerą
 - 14.4. Domyślne elementy sterujące
 - 14.5. Dodawanie środowiska
 - 14.6. Stosowanie tekstury obrazu
 - 14.7. Tworzenie środowiska niestandardowego (opcjonalnie)
 - 14.8. Dodawanie tła do sceny
 - 14.9. Dodawanie podłoża
 - 14.10. Poprawianie oświetlenia
 - 14.11. Dodawanie animacji
 - 14.12. Zaawansowane szczegóły
 - 14.13. Dodawanie interakcji
 - 14.14. Komponent odbiornika zdarzeń
 - 14.15. Animowanie zdarzeń
 - 14.16. Dodawanie audio
 - 14.17. Dodawanie tekstu
- 15. Tworzenie galerii obrazów 360-stopniowych
 - 15.1. Szkielet 103
 - 15.2. Używanie standardowych komponentów
 - 15.3. Używanie komponentów społeczności
 - 15.4. Komponent szablonu
 - 15.5. Komponent zestawu zdarzeń

- 15.6. Pisanie komponentu specjalnie dla aplikacji
- 16. Budowanie scen 3D z MagicaVoxel
 - 16.1. Instalacja
 - 16.2. Film instruktażowy
 - 16.3. Okno edycji
 - 16.4. Paleta kolorów
 - 16.5. Pędzle
 - 16.6. Działania
 - 16.7. Eksportowanie do A-Frame
 - 16.7.1. Format PLY (z wbudowanymi cieniami)
 - 16.7.2. Format OBJ
- 17. Budowanie demo à la Minecraft
 - 17.1. Przykładowy szkielet
 - 17.2. Dodawanie podłoża
 - 17.3. Wstępne ładowanie zasobów
 - 17.4. Dodawanie tła
 - 17.5. Dodawanie wokseli
 - 17.6. Wzór komponentowo-jednostkowy
 - 17.7. Komponent random-color
 - 17.8. Komponent snap
 - 17.9. Mixins
 - 17.10. Dodawanie teleportacji do lewej ręki
 - 17.11. Dodawanie woksela przemieszczania do prawej ręki
 - 17.12. Dodawanie wsparcia dla urządzeń mobilnych i komputerów

9. Wprowadzenie do Aframe

Aframe to szkielet do budowy aplikacji stworzony przez ludzi z Mozilli. Ułatwia rozpoczęcie pracy z wirtualną rzeczywistością. Jest zbudowany na bazie three.js. Istnieje wiele komponentów zbudowanych na aframe, wraz z wieloma stworzonymi przez społeczność.

Aframe ułatwia tworzenie dobrych doświadczeń VR nawet amatorom JS. Mając trochę więcej wiedzy na temat threejs, możesz także tworzyć własne komponenty aframe. Dzięki temu, że szkielet ten wewnętrznie korzysta z aplikacji three.js, możliwości są nieograniczone, ponieważ możesz tworzyć własne komponenty niestandardowe, gdy coś, czego chcesz, nie jest dostępne.

10. Czym jest aframe

Aframe to system programowania jednostkowo-komponentowego, oparty na three.js.

Ułatwia pracę z three.js, ma również wiele komponentów tworzonych przez społeczność, co ułatwia każdemu rozpoczęcie pracy z wirtualną rzeczywistością przy pełnej wiedzy o threejs. Najlepsze w Aframe jest to, że jeśli chcesz stworzyć własne niestandardowe komponenty, możesz to zrobić za pomocą three.js.

Najlepszym sposobem, by zrozumieć Aframe, jest zapoznanie się z przykładowym sposobem jego użycia.

Stworzymy prosty sposób wyświetlania obrazów 360-

stopniowych.

Istnieją różne sposoby budowania aplikacji VR za pomocą Aframe. Tutaj będziemy budować je przy użyciu zwykłego HTML.

11. Rozbudowywanie HTML 5 o aframe

Najpierw musimy uwzględnić Aframe JS wbudowany w HTML. Cała akcja Aframe odbywa się wewnątrz komponentu a-scene. Wewnątrz tego możesz dodać pozostałe komponenty.

Dodajmy do naszej sceny prostą kulę.



I tyle. Spójrzmy na komponenty, których użyliśmy.

a-scene

Tworzy scenę threejs. Wszystkie jednostki są zawarte w scenie. Scena może zawierać zasoby, geometrię i tak dalej. Podczas pisania kodu Aframe komponenty są budowane wewnątrz a-sceny.

a-sphere

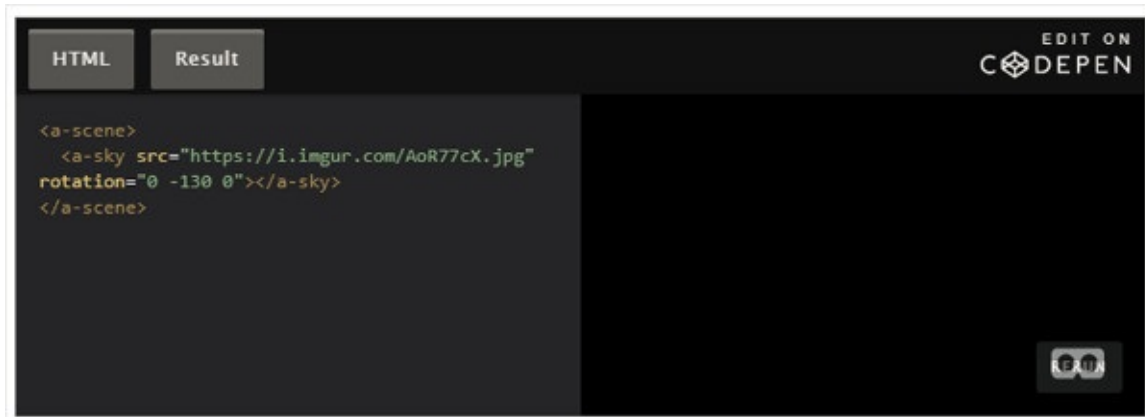
Ten komponent tworzy kształty sferyczne lub wielościenne. Wewnątrz tworzy komponent geometryczny z prymitywnym zestawem jako kulą. Jest to najbardziej podstawowy przykład w Aframe.

Teraz, gdy udało nam się stworzyć kulę w naszym środowisku, stwórzmy coś bardziej złożonego.

11.1. Panorama w wirtualnej rzeczywistości

We wprowadzeniu do Aframe będziemy wyświetlać panoramę wewnątrz naszej a-scene. Kiedy oglądamy na urządzeniu innym niż VR, takim jak komputer stacjonarny, możemy przeciągać wokół panoramy. Aby przejrzeć panoramę podczas oglądania w VR, wystarczy poruszać głową.

Ponieważ używamy Aframe, która owija funkcje threejs w niestandardowe komponenty, jesteśmy w stanie bardzo łatwo stworzyć 360-stopniową wirtualną rzeczywistość panoramiczną. Gdybyśmy robili to w zwykły sposób w threejs, potrzebowalibyśmy więcej kodów i musielibyśmy dłużej nad tym pomyśleć. Musielibyśmy napisać kod do obsługi jednocześnie w urządzeniach VR oraz innych niż VR.



To takie proste.

11.2. Co udało nam się tu zrobić dzięki Aframe

Przejdźmy przez to, co udało się zrobić dzięki Aframe. Pierwszym znacznikiem jest `a-scene`. Jak wyjaśniono wcześniej, tworzymy tu scenę `threejs`, gdzie dodajemy nasze treści.

`a-sky` tworzy sferyczną geometrię w `threejs`. Używa się go do nadania tła lub wyświetlania 360-stopniowych równoodległościowych zdjęć. Kiedy ten kod jest uruchamiany, kamera znajduje się w centrum kuli i rozglądamy się wokół.

Gdy tryb VR jest włączony w urządzeniach nieobsługujących VR, przełącza się na pełny ekran. W przeciwnym razie wyświetli efekt VR.

12. Jak zbudować swoją pierwszą scenę w VR

Pomocna okazuje się dokumentacja `a-frame`. Jest wiele przydatnych sekcji, które możesz sprawdzić, aby uzyskać

bardziej dogłębny wygląd. Mają też bloga i możesz do nich łatwo dotrzeć.

```
<script src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
<a-scene>
  <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
  <a-box position="-1 0.5 -3" rotation="0 45 0" width="1" height="1" depth="1" color="#4CC3D9"></a-box>
  <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
  <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
  <a-sky color="#ECECEC"></a-sky>
</a-scene>
```

Pierwsza scena w a-frame

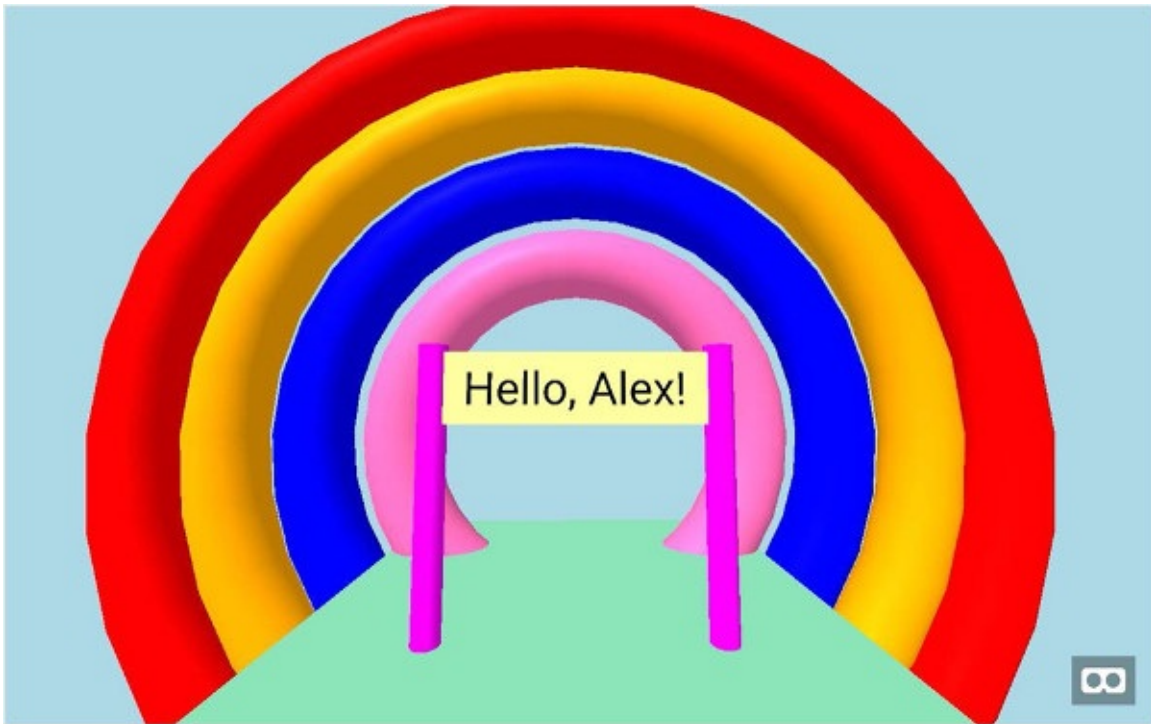
Podstawowy przykład, który widzisz na powyższym zrzucie ekranu, ma w sumie 9 linii.

Pierwsza linia, określana jako skrypt, to plik JavaScript, który zawiera kod do wykonania ramki. Linia ta musi być obecna we wszystkich scenach a-frame.

Pozostałe linie opisują to, co jest wewnątrz samej sceny, która obejmuje kulę, skrzynkę, cylinder, samolot i niebo. Możesz modyfikować te obiekty (zwane także jednostkami) zgodnie z własnymi potrzebami. Na przykład, aby przesunąć obiekty dookoła, można zmienić wartości osi x, y, z wewnątrz atrybutów położenia i obrotu.

Możesz również łatwo modyfikować wartości promienia, szerokości, wysokości i głębokości, aby obiekty były większe lub mniejsze, większe lub cieńsze itd.

Możesz także zmienić kolor, bawiąc się kodami kolorów HTML (użyтыми w powyższym przykładzie) lub po prostu wpisując nazwę wybranego koloru.



Mój pierwszy „twórczy” przykład

Kiedy zrozumiesz podstawowy przykład, możesz zaszaleć i zbudować bardziej osobistą (w moim przypadku bardziej psychodeliczną) scenę.

Zrzut ekranu powyżej pokazuje pierwszy, łatwy do zbudowania „kreatywny” przykład. Rzućmy okiem na kod.

```
1 <script src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
2
3 <a-scene>
4   <a-torus color="hotpink" position="0 1.25 -5" radius="2"></a-torus>
5   <a-torus color="orange" position="0 1.25 -5" radius="4" radius-tubular="0.2"></a-torus>
6   <a-torus color="blue" position="0 1.25 -5" radius="3" radius-tubular="0.2"></a-torus>
7   <a-torus color="red" position="0 1.25 -5" radius="5" radius-tubular="0.2"></a-torus>
8
9   <!-- banner pillars -->
10  <a-cylinder position="1 1 -3" rotation="0 45 0" radius="0.1" height="2" depth="0.2" color="magenta"></a-cylinder>
11  <a-cylinder position="-1 1 -3" rotation="0 45 0" radius="0.1" height="2" depth="0.2" color="magenta"></a-cylinder>
12
13  <!-- ground and sky -->
14  <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
15  <a-sky color="lightblue"></a-sky>
16 </a-scene>
```

Jak widzisz, dodałem do sceny cztery różne jednostki torusa. Zmieniłem ich promień i kolor, pozostawiając je w tej samej pozycji, aby uzyskać okrągły efekt podobny do tęczy.

Dla filarów mojego nagłówka zmodyfikowałem tylko dwa cylindry w kolorze magenta, dzięki czemu były one wyższe i cieńsze. I aby „wspierać” moje filary, dodałem grunt, który jest w zasadzie płaszczyzną, jaka pokrywa powierzchnię. Na końcu nadałem niebu kolor jasnoniebieski.

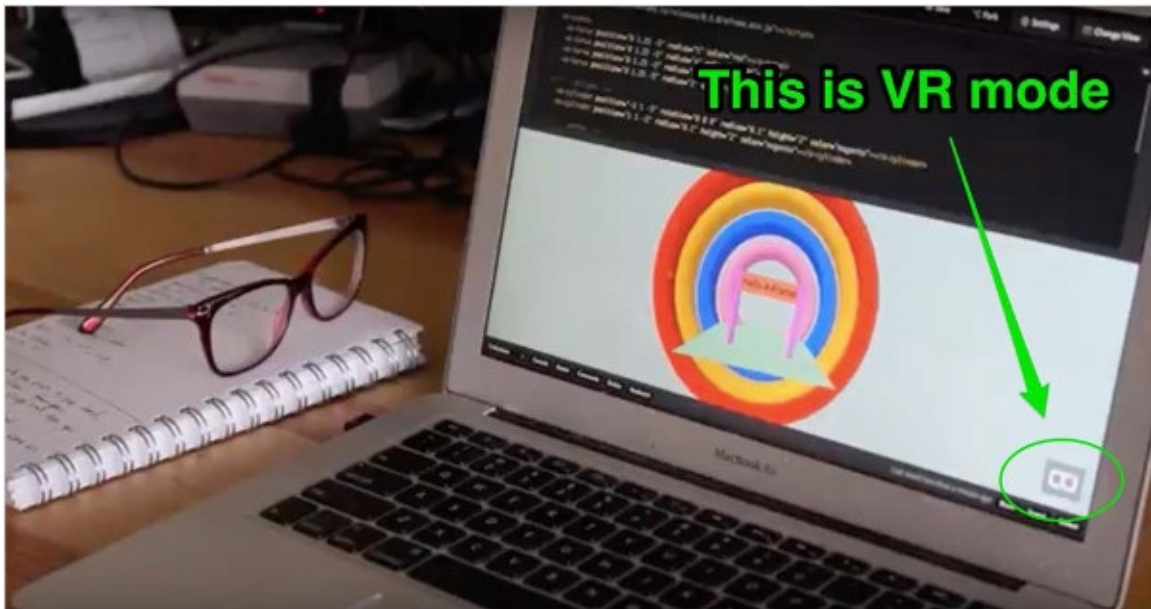
12.1. Dodawanie tekstu

```
17 <!-- text banner -->
18 <a-text value="Hello, Alex!" position="-0.8 1.7 -2.9" color="black" width="7"></a-text>
19 <a-box position="0.01 1.7 -3" rotation="0 0 0" width="2" height="0.5" depth="0.1" color="khaki"></a-box>
20
21 </a-scene>
```

Dodawanie tekstu jest bardzo proste, jak wszystko w a-frame. Pierwsza linia powyższego zrzutu ekranu to wszystko, czego potrzebujesz: komponent tekstowy klatki `<a-text>`, wartość (cokolwiek chcesz napisać), pozycja (gdzie chcesz umieścić ją na scenie), kolor (nie muszę wyjaśniać, o co chodzi) i szerokość (jak gruby ma być tekst).

Aby uczynić z niego nagłówek, właśnie umieściłem tekst na wierzchu pudełka, które zmodyfikowałem tak, by mieściło się pomiędzy dwoma cylindrycznymi filarami.

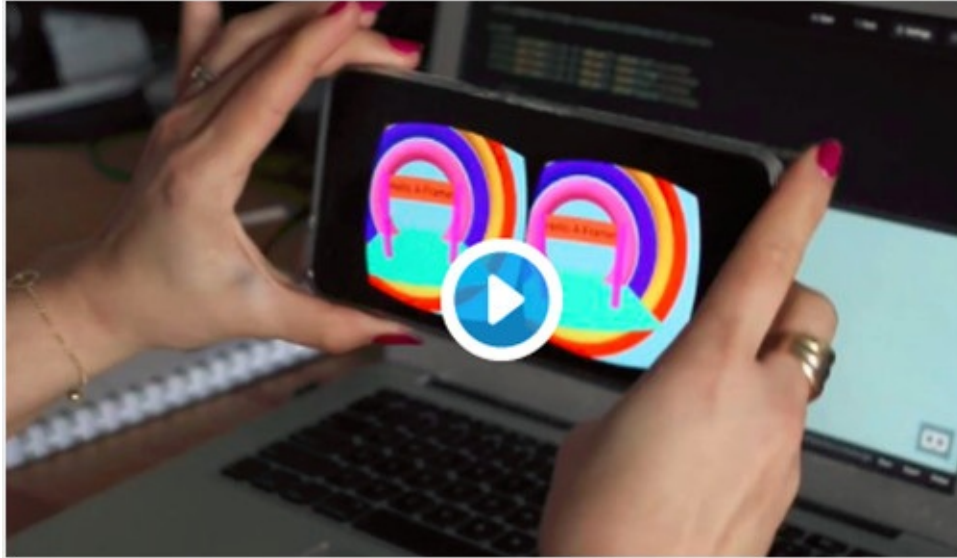
13. Tryb VR



Jak można zobaczyć swoją scenę w rzeczywistości wirtualnej za pomocą jednego kliknięcia?

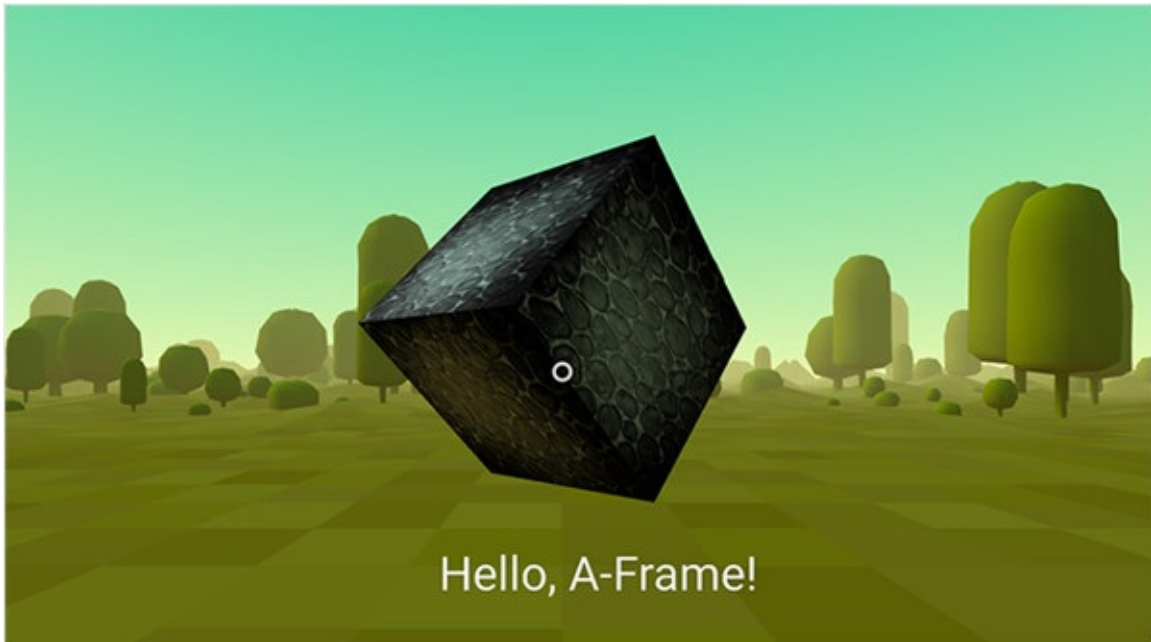
To jedna z najfajniejszych rzeczy w A-Frame. To jest szkielet, który umożliwia tworzenie wirtualnej rzeczywistości, dzięki czemu możesz jednym kliknięciem zamienić swoją scenę w VR.

Aby móc zobaczyć to na swoim telefonie, po prostu skopiuj adres URL i wklej go w przeglądarce mobilnej. Gdy scena pojawi się na ekranie, możesz wejść z nią w interakcję, a po kliknięciu w przycisk przypominający parę kartonowych okularów w prawym dolnym rogu wejdiesz w tryb VR. Jest to użyteczne (i można używać od zaraz!), jeśli posiadasz urządzenia takie jak Google Cardboard czy Gear VR albo urządzenia obsługujące platformę Daydream, itd.



I tyle, właśnie stworzyliście pierwszą scenę VR!

14. Budowanie większej sceny



Zacznijmy od stworzenia podstawowej sceny A-Frame. W tym celu potrzebujemy podstawowej wiedzy na temat HTML.

Nauczmy się, jak:

- dodawać elementy 3D (np. Obiekty) za pomocą elementów podstawowych,
- przekształcać obiekty w przestrzeni 3D z pozycją, obrotem, skalą,
- dodawać środowisko,
- dodawać tekstury,
- dodawać podstawową interaktywność za pomocą animacji i zdarzeń,
- dodawać tekst.

Zaczynamy od minimalnej struktury HTML:

```
<html>
  <head>
    <script
src="https://aframe.io/releases/0.7.0/aframe.min.js">
</script>
  </head>
  <body>
    <a-scene>
    </a-scene>
  </body>
</html>
```

Załączamy A-Frame jako znacznik skryptu w `<head>`, wskazując na kompilację A-Frame hostowaną na CDN. To musi być zawarte przed `<a-scene>`, ponieważ A-Frame rejestruje niestandardowe elementy HTML, które muszą być zdefiniowane,

zanim `<a-scene>` zostanie dopisane, w innym przypadku `<a-scene>` nic nie zmieni.

Następnie dodajemy `<a-scene>` do `<body>`. `<a-scene>` będzie zawierał każdą jednostkę w naszej scenie. `<a-scene>` obsługuje całą konfigurację, która jest wymagana dla 3D: konfigurowanie WebGL, obrazy, kamery, światła, renderer, renderowanie, a także obsługa WebVR na platformach takich jak HTC Vive, Oculus Rift, Samsung GearVR i smartfony (Google Cardboard). Sam `<a-scene>` wymaga od nas włożenia dużo wysiłku.

14.1. Dodawanie jednostki

W ramach naszej `<a-scene>` dołączamy obiekty 3D, używając jednego ze standardowych prymitywów graficznych A-Frame `<a-box>`. Możemy użyć `<a-box>` tak jak normalnego elementu HTML, definiując znacznik i używając atrybutów HTML, aby go dostosować. Inne przykłady prymitywów, które pochodzą z A-Frame, to `<a-cylinder>`, `<a-plane>` lub `<a-sphere>`. Tutaj definiujemy kolor `<a-box>`. Zobacz dokumentację `<a-box>`, aby uzyskać więcej atrybutów (np. szerokość, wysokość, głębokość).



Zdjęcie autorstwa Rubena Muellera z vrjump.de

```
<a-scene>  
  <a-box color="red"></a-box>  
</a-scene>
```

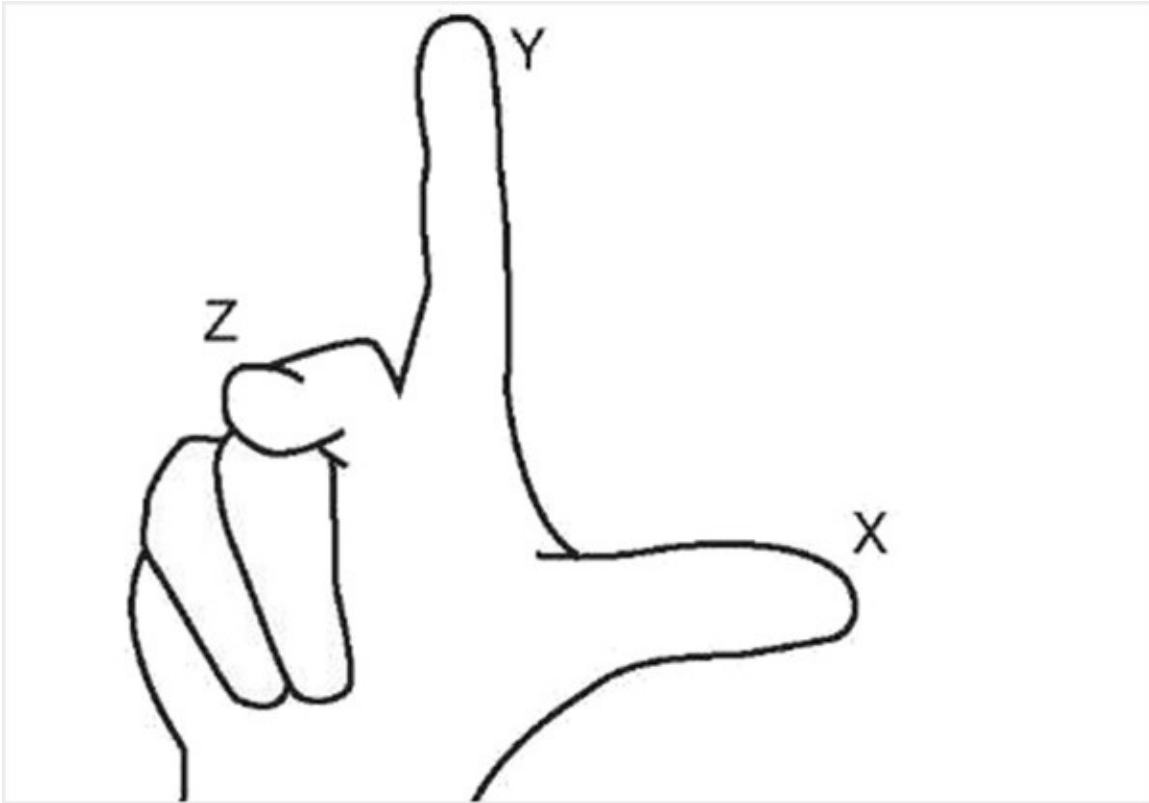
Na marginesie chcielibyśmy dodać, że prymitywy są łatwymi w użyciu elementami HTML A-Frame, które stanowią podstawowy zespół jednostek i komponentów. Mogą być wygodne w użyciu, ale pod `<a-box>` jest `<a-entity>` z geometrią i komponentami materiałowymi:

```
<a-entity id="box" geometry="primitive: box"  
material="color: red"></a-entity>
```

Niemniej jednak, ponieważ domyślna kamera i skrzynka są ustawione w domyślnej pozycji w punkcie początkowym 0 0 0, nie będziemy mogli zobaczyć tego pola, chyba że je przenieśmy. Możemy to zrobić, używając komponentu pozycji do transformacji pudełka w przestrzeni 3D.

14.2. Przekształcanie jednostki w 3D

Przejdźmy najpierw przez przestrzeń 3D. A-Frame używa praworęcznego układu współrzędnych. Przy domyślnym kierunku kamery: dodatnia oś X rozciąga się w prawo, dodatnia oś Y wystaje w górę, a dodatnia oś Z wychodzi z ekranu w naszą stronę:



Obraz z what-when-how.com

Jednostka odległości A-Frame podana jest w metrach, ponieważ interfejs WebVR API wykazuje dane pozycyjne w metrach. Projektując scenę dla VR, należy wziąć pod uwagę rzeczywistą skalę świata tworzonych przez nas obiektów. Pudło o wysokości `height="10"` może wyglądać normalnie na ekranach komputerów, ale w rzeczywistości wirtualnej skrzynka będzie wyglądać masywnie.

Obrotowa jednostka A-Frame podana jest w stopniach, chociaż zostanie przekształcona wewnętrznie na radiany, gdy przejdzie do pliku `three.js`. Aby określić dodatni kierunek obrotu, użyj reguły prawej ręki. Skieruj kciuki w dół w kierunku osi dodatniej i w kierunku, w którym nasze palce

kręcą się wokół dodatniego kierunku obrotu.

Aby przetłumaczyć, obrócić i przeskalować skrzynkę, możemy zmienić pozycję, obrót i komponenty skali. Najpierw zastosujemy komponenty rotacji i skali:

```
<a-scene>  
  <a-box color="red" rotation="0 45 45" scale="2 2 2"></a-  
box>  
</a-scene>
```

Spowoduje to obrócenie pudełka pod kątem i podwoi jego rozmiar.

Transformacje rodzica i potomka

A-Frame HTML reprezentuje wykres sceny 3D. Na wykresie sceny jednostki mogą mieć jednego rodzica i wiele potomków. Jednostki potomków dziedziczą transformacje (tj. pozycję, obrót i skalę) od swoich jednostek nadrzędnych (rodziców).

Na przykład możemy mieć kulę jako dziecko pudełka:

```
<a-scene>  
  <a-box position="0 2 0" rotation="0 45 45" scale="2 4 2">  
    <a-sphere position="1 0 3"></a-sphere>  
  </a-box>  
</a-scene>
```

Jeśli obliczysz światową pozycję kuli, będzie to 1 2 3, osiągnięte przez ułożenie pozycji macierzystej sfery z jej własną pozycją. Podobnie w przypadku obrotu i skali, sfera

odziedziczyłaby obrót i skalę pudełka. Kula również byłaby obracana i rozciągana, podobnie jak jej pudełko macierzyste. Jeśli pudełko zmieni swoją pozycję, obrót lub skalę, natychmiast zastosuje się do sfery i wpłynie na nią.

Gdybyśmy dodali do kuli cylinder jako jej dziecko, transformacja cylindra będzie zależna zarówno od transformacji kuli, jak i skrzynki. Pod maską w `three.js` robi się to przez pomnożenie macierzy transformacji. Na szczęście nie musimy o tym myśleć.

14.3. Umieszczanie pudełka przed kamerą

Spróbujmy teraz sprawić, aby pudełko było widoczne na kamerze od samego początku.

Możemy przenieść skrzynię z powrotem o 5 metrów na ujemną oś Z z komponentem pozycji. Musimy również przesunąć go o 2 metry na dodatniej osi Y , aby skrzynia nie przecinała się z ziemią, ponieważ skalowaliśmy skrzynię, a skalowanie odbywa się od środka:

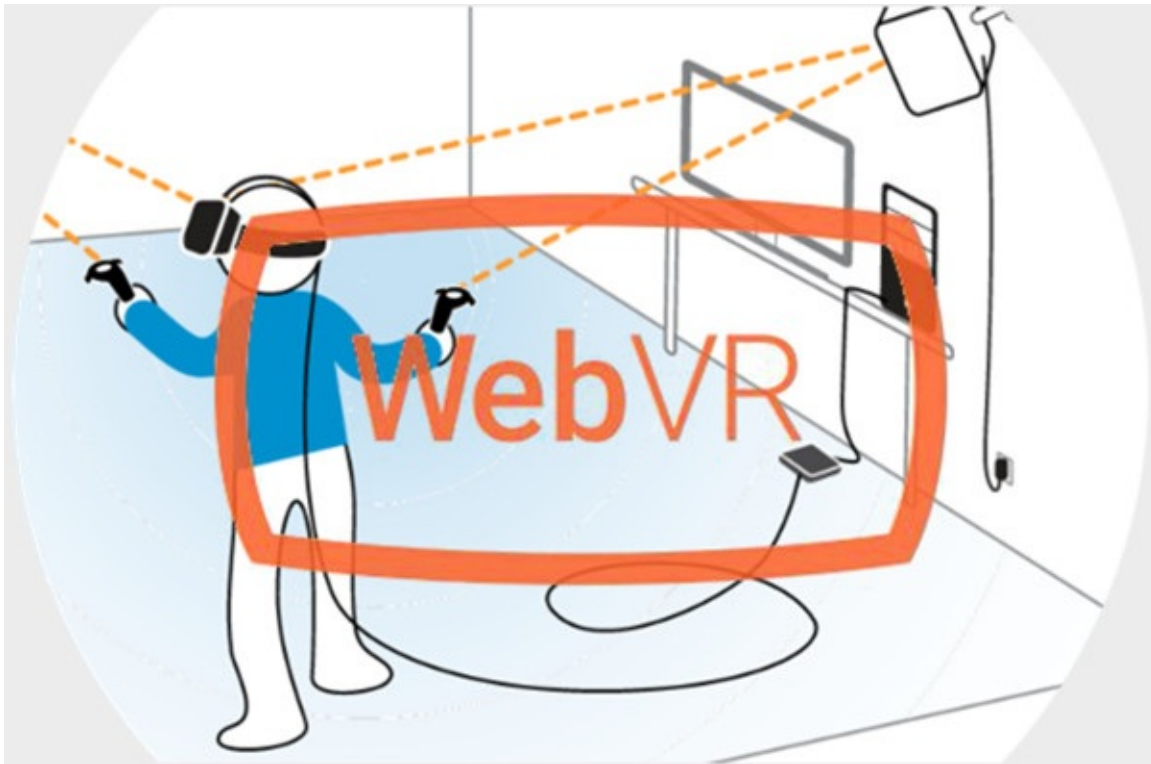
```
<a-scene>
  <a-box color="red" position="0 2 -5" rotation="0 45 45"
scale="2 2 2"></a-box>
</a-scene>
```

Teraz widzimy nasze pudełko!



14.4. Domyślne elementy sterujące

W przypadku płaskich wyświetlaczy (np. laptopa, komputera stacjonarnego) domyślny schemat sterowania pozwala rozglądać się za pomocą przeciągnięcia myszą i przeciągnięcia za pomocą klawiszy WASD lub strzałek. W smartfonie, aby obrócić kamerę, możemy obrócić telefon. Chociaż A-Frame jest dostosowany do WebVR – ten domyślny schemat sterowania umożliwia oglądanie sceny bez zestawu VR.



Sprawdź instrukcje na stronie domowej Mozilla VR, aby skonfigurować i wprowadzić WebVR.

Wchodząc do rzeczywistości wirtualnej, po kliknięciu ikony gogli z podłączonym zestawem słuchawkowym VR (np. Oculus Rift, HTC Vive), możemy doświadczyć tej sceny w realistycznej rzeczywistości wirtualnej. Jeśli dostępna jest skala pokojowa, możemy fizycznie poruszać się po scenie.

14.5. Dodawanie środowiska

A-Frame pozwala programistom tworzyć i udostępniać komponenty wielokrotnego użytku, z których inni mogą z łatwością korzystać. Komponent środowiskowy @feiss'a w ramach procedur generuje dla nas różne środowiska z pojedynczą linią HTML. Komponent środowiska jest świetnym i

łatwym sposobem na wizualne organizowanie naszej aplikacji VR, zapewniając kilkunastu środowiskom wiele parametrów.

Najpierw wpisz komponent środowiska za pomocą znacznika skryptu po A-Frame:

```
<head>
<script
src="https://aframe.io/releases/0.7.0/aframe.min.js">
</script>
<script src="https://unpkg.com/aframe-environment-
component/dist/aframe-environment-component.min.js">
</script>
</head>
```

Następnie, wewnątrz sceny, dodaj jednostkę z dołączonym komponentem środowiska. Możemy określić wartość fabryczną (np. forest) wraz z wieloma innymi parametrami (np. 200 drzew):

```
<a-scene>
  <a-box color="red" position="0 2 -5" rotation="0 45 45"
scale="2 2 2"></a-box>

  <!--To doszło: -->
  <a-entity environment="preset: forest; numDressing:
500"></a-entity>
</a-scene>
```

14.6. Stosowanie tekstury obrazu

Upewnij się, że obsługujesz kod HTML za pomocą lokalnego serwera, aby tekstury ładowały się prawidłowo.

Możemy zastosować teksturę obrazu do pudełka z obrazem, wideo lub `<canvas>`, używając atrybutu `src`, tak jak byśmy mieli standardowy element ``. Powinniśmy również usunąć `color="red"`, który ustawiliśmy tak, aby kolor nie mieszał się z teksturą. Domyślnym kolorem materiału jest white (biały), więc wystarczy usunąć atrybutu koloru.

```
<a-scene>
  <a-box      src="https://i.imgur.com/mYmmbrp.jpg"
position="0 2 -5" rotation="0 45 45"
  scale="2 2 2"></a-box>
  <a-sky color="#222"></a-sky>
</a-scene>
```

Teraz zobaczymy nasze pudełko z teksturą obrazu pobraną z Internetu.



Korzystanie z systemu zarządzania zasobami

Korzystanie z systemu zarządzania zasobami zalecamy w celu podniesienia wydajności. System zarządzania zasobami ułatwia bowiem przeglądarce przechowywanie zasobów (np. obrazów, filmów, modeli), a z A-Frame mamy pewność, że wszystkie zasoby zostaną pobrane przed renderowaniem.

Jeśli zdefiniujemy `` w systemie zarządzania zasobami, później plik `three.js` nie musi wewnętrznie tworzyć elementu ``. Tworzenie `` przez nas samych daje nam także większą kontrolę i umożliwia ponowne wykorzystanie tekstury na wielu obiektach. A-Frame jest również wystarczająco inteligentny, aby ustawić `crossOrigin` i inne tego typu atrybuty, gdy jest to konieczne.

Aby użyć systemu zarządzania zasobami dla tekstury obrazu:

- dodaj `<a-assets>` do sceny,
- zdefiniuj teksturę jako `` w `<a-assets>`,
- nadaj znacznikowi `` identyfikator HTML (np. `id="boxTexture"`),
- odwołaj się do zasobu, używając identyfikatora w formacie selektora obiektowego modelu dokumentu (`src="#boxTexture"`).

```
<a-scene>
  <a-assets>
    
  </a-assets>

  <a-box src="#boxTexture" position="0 2 -5" rotation="0 45
```

```
45" scale="2 2 2"></a-box>
  <a-sky color="#222"></a-sky>
</a-scene>
```

14.7. Tworzenie środowiska niestandardowego (opcjonalnie)

Wcześniej omówiliśmy komponent środowiskowy generujący środowisko. Choć dobrze jest wiedzieć trochę o tworzeniu podstawowego środowiska do celów edukacyjnych.

14.8. Dodawanie tła do sceny

Możemy dodać tło z `<a-sky>`, które otacza scenę. `<a-sky>`, czyli materiał, który stosuje się wewnątrz dużej kuli, może mieć kolor jednolity, obraz 360-stopniowy lub wideo 360-stopniowe. Na przykład ciemnoszare tło byłoby:

```
<a-scene>
  <a-box color="red" position="0 2 -5" rotation="0 45 45"
scale="2 2 2"></a-box>
```

```
  <a-sky color="#222"></a-sky>
</a-scene>
```

Możemy także użyć tekstury obrazu, aby uzyskać 360-stopniowy obraz tła przy użyciu `src` zamiast `color`:

```
<a-scene>
  <a-assets>
```

```

        
    
    </a-assets>

    <a-box src="#boxTexture" position="0 2 -5" rotation="0 45
45" scale="2 2 2"></a-box>

    <a-sky src="#skyTexture"></a-sky>
</a-scene>

```

14.9. Dodawanie podłoża

Aby dodać teren, możemy użyć `<a-plane>`. Domyślnie płaszczyzny są zorientowane równoległe do osi XY. Aby znajdowała się równoległe do podłoża, musimy ustawić ją wzdłuż osi XZ. Możemy to zrobić, obracając płaszczyznę ujemną o 90 stopni na osi X.

```
<a-plane rotation="-90 0 0"></a-plane>
```

Ważne jest, aby teren był odpowiednio duży, byśmy mogli zwiększyć szerokość i wysokość. Ustawmy dla niego wartość 30 metrów w każdym kierunku:

```
<a-plane rotation="-90 0 0" width="30" height="30"></a-
plane>
```

Następnie możemy zastosować teksturę obrazu do naszego podłoża:

```
<a-assets>
  <!-- ... -->
  
  <!-- ... -->
</a-assets>

<!-- ... -->
<a-plane src="#groundTexture" rotation="-90 0 0"
width="30" height="30"></a-plane>
<!-- ... -->
```

Aby ustawić teksturę, możemy użyć atrybutu `repeat`. Atrybut ten przyjmuje dwie liczby: ile razy należy powtórzyć w kierunku X oraz ile razy w kierunku Y (powszechnie określane jako U i V dla tekstur).

```
<a-plane src="#groundTexture" rotation="-90 0 0"
width="30" height="30"
repeat="10 10"></a-plane>
```

14.10. Poprawianie oświetlenia

Możemy zmienić sposób oświetlania sceny za pomocą `<a-light>s`. Domyślnie, jeśli nie określamy żadnych świateł, A-Frame dodaje światło otoczenia i światło kierunkowe. Jeśli A-Frame nie doda światła, scena będzie czarna. Gdy dodamy

własne światła, domyślna konfiguracja oświetlenia zostanie usunięta i zastąpiona przez naszą konfigurację.

Dodamy światło otoczenia, które ma lekką niebiesko-zieloną barwę, która przypomina kolor nieba. Światła otoczenia są stosowane do wszystkich obiektów na scenie (biorąc pod uwagę, że zastosowano przynajmniej domyślny materiał).

Dodamy również światło punktowe. Światła punktowe są jak żarówki – możemy ustawić je wokół sceny, a wpływ światła punktowego na jednostkę zależy od odległości od niej:

```
<a-scene>
  <a-assets>
    
    
    
  </a-assets>

  <a-box src="#boxTexture" position="0 2 -5" rotation="0 45
45" scale="2 2 2"></a-box>

  <a-sky src="#skyTexture"></a-sky>

  <a-light type="ambient" color="#445451"></a-light>
  <a-light type="point" intensity="2" position="2 4 4"></a-
```

```
light>  
</a-scene>
```

14.11. Dodawanie animacji

`<a-animation>` może zostać uznana za przestarzałą w porównaniu z komponentem animacji.

Możemy dodawać animacje do ramki za pomocą wbudowanego systemu animacji A-Frame. Animacje interpolują lub przechodzą między wartościami w czasie. Możemy umieścić element `<a-animation>` jako element potomny jednostki. Załóżmy, że pudełko unosi się w górę i w dół, aby dodać ruch do sceny.

```
<a-scene>  
  <a-assets>  
      
  </a-assets>  
  <a-box src="#boxTexture" position="0 2 -5" rotation="0 45  
45" scale="2 2 2">  
    <a-animation attribute="position" to="0 2.2 -5"  
direction="alternate" dur="2000"  
repeat="indefinite"></a-animation>  
  </a-box>  
</a-scene>
```

`<a-animation>` służy do tego, by:

- animować atrybut pozycji,

- animować do 0 2.2. -5, który jest 20 centymetrów wyższy,
- zmieniać kierunek animacji w każdym powtarzającym się cyklu animacji,
- nadać czas trwania 2000 milisekund w każdym cyklu,
- powtarzać animację w nieskończoność.



14.12. Zaawansowane szczegóły

`<a-animation>` przechwytuje pętlę renderowania A-Frame tak, że wywoływana jest tylko raz na ramkę. Jeśli potrzebujesz większej kontroli i chcesz ręcznie interpolować lub animować wartość, możesz napisać komponent A-Frame za pomocą obsługi znaczników tick i biblioteki, takiej jak Tween.js (która na chwilę obecną jest dostępna na stronie AFRAME.TWEEN). Aby osiągnąć najlepszą wydajność, operacje na klatkach raz-na-klatkę powinny być wykonywane na poziomie A-frame. Staraj się nie tworzyć własnych requestAnimationFrame, gdy A-frame już je posiada.

KONIEC WERSJI DEMONSTRACYJNEJ

*Dziękujemy za skorzystanie z oferty naszego wydawnictwa i życzymy miło
spędzonych chwil przy kolejnych naszych publikacjach.*

Wydawnictwo Psychoskok

