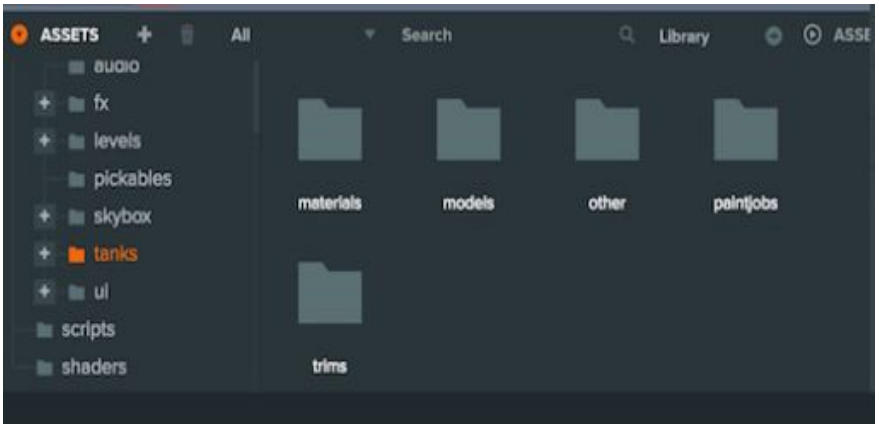
PlayCanvas nieoficjalnie

tylko o PlayCanvas

Autor: bunnymq

Spis treści

Część I - ogólnie



Rozdział 1

PlayCanvas, cechy

1.1 PlayCanvas - co to?

Silnik gier stworzony w WebGL, nie bazuje, nie korzysta z biblioteki 3D threejs, **został napisany od zera,** też jako platforma w chmurze do tworzenia gier, wizualizacji, konfiguratorów produktów (np. konfigurator samochodu). Można w nim zrobić bardzo zaawansowaną grafikę, dzięki możliwościom silnika.

Ma on zintegrowany silnik fizyki Ammo. Z PlayCanvas można korzystać jako z platformy z użyciem edytora graficznego i edytora kodu w chmurze lub jako engine-only, czyli mając projekt lokalnie na swoim laptopie i korzystając tylko z silnika w wybranym IDE lub edytorze kodu (VS Code, Atom, Sublime Text), coś jak jest to realizowane w three.js.

Z engine-only jest jedna zaleta możesz korzystać z gita i wrzucać na githuba, w przypadku platformy musisz korzystać z PlayCanvas'owego systemu kontroli wersji i checkpointów, a nie commitów jak to działa w git.

Nazwa przypomina trochę jakby to miało chodzić o klasyczny canvas, HTML5, czyli 2D, ale jednak jest to bogaty silnik 3D, podobny silnik do PlayCanvas to Babylon.js, też warty spróbowania, a PlayCanvas polecam naprawdę fajny silnik, ale szczerze dopiero od niedawna była wcześniej jedna wada, był limit miejsca na zasoby 100 MB jeżeli korzystałeś z platformy. Teraz limit ten jest 1GB, czyli możesz trzymać zasoby na platformie za darmo jeśli razem nie przekraczają 1 GB, żeby np. mieć 10 GB lub więcej musisz mieć wykupioną miesięczną subskrypcję.

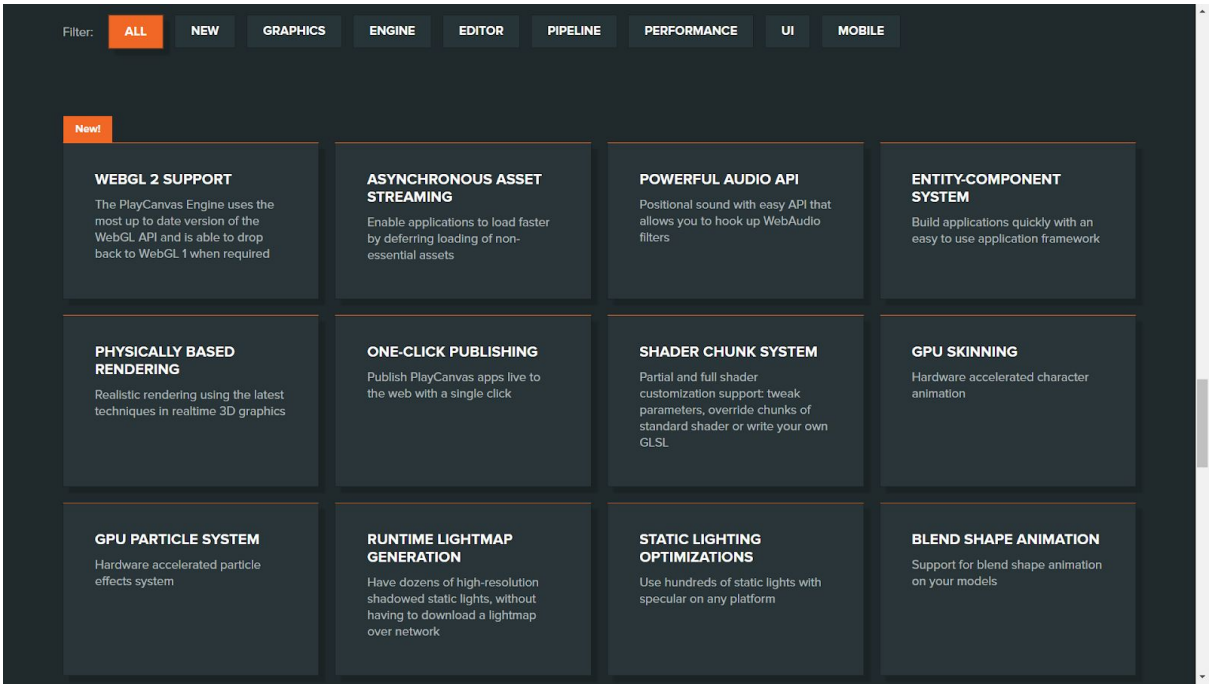
Społeczność nie jest mała, dokumentacja jest dobrze napisana, tylko jest jedna wada nie ma na ten temat książek ani po angielsku ani po polsku.

W przypadku problemów możesz znaleźć post dotyczący twojego problemu lub możesz zapytać na forum tworząc nowy post. Nie jest tak że nikt nie odpowiada na twoje pytanie,

bardzo szybko dostajesz odpowiedź, a nawet możliwe rozwiązanie problemu, co jest bardzo mocną zaletą forum PlayCanvas. Link do forum podaję tu [PlayCanvas Discussion]

Z ciekawości przeglądałem kod silnika, jest on naprawdę duży, chociaż nie aż tak ogromny jak to jest w przypadku Unreal Engine.

1.2 cechy (features)



PlayCanvas posiada następujące cechy:

1. wsparcie WebGL 2
2. asynchroniczne streamowanie zasobów
3. audio API
4. ECS (Entity Component System) - o tym w części Encja
5. renderowanie oparte o fizykę (PBR)
6. system shader chunk
7. skinning GPU
8. system cząsteczek GPU
9. generowanie mapy świateł w czasie rzeczywistym
10. animacja mieszania kształtu
11. miękkie cienie i light cookie
12. importer zasobów i menedżer

13. potok graficzny liniowy i HDR
14. API urządzeń wejścia
15. renderer fontu SDF
16. silnik fizyki dotyczący rigidbody
17. narzędzia dla responsywnych interfejsów
18. wsparcie WebVR
19. rozwój i testowanie na urządzeniu mobilnym
20. filtrowanie zasobów
21. edytowanie scen w czasie rzeczywistym
22. prefiltering tekstury sześciennej
23. profiler
24. kompresja tekstur (DXT, PVR i ETC1)
25. edytor materiału
26. wieloplatformowość

1. **wsparcie WebGL 2**
   Silnik używa najnowsze WebGL API, ale jest wstecznie kompatybilny z WebGL wersji pierwszej.
2. **asynchroniczne streamowanie zasobów**
   Asynchroniczne, a więc co za tym idzie szybsze ładowanie się aplikacji, poprzez opóźnienie ładowania mniej ważnych zasobów.
3. **audio API**
   Pozycyjny dźwięk pozwala na przyczepianie filtrów WebAudio.
4. **ECS (Entity Component System) - o tym w części Encja**
   Twórz aplikacje szybko z wykorzystaniem ECS.
5. **renderowanie oparte o fizykę (PBR)**
   Zapewnij realizm w renderingu z użyciem najnowszych technik czasu rzeczywistego w grafice 3D
6. **system shader chunk**
   Częściowa i pełna customizacja shaderów: dostosuj parametry, nadpisuj kawałki standardowego shadera lub pisz swój kod GLSL
7. **skinning GPU**
   Przyspieszana sprzętowo animacji postaci
8. **system cząsteczek GPU**

Przyspieszany sprzętowo system cząsteczek

9. **generowanie mapy świateł w czasie rzeczywistym**

   Możesz mieć wiele statycznych świateł wysokiej rozdzielczości

10. **animacja mieszania kształtu**

    Wsparcie dla animacji mieszania kształtu modeli

11. **miękkie cienie i light cookie**

    Wybieraj spośród wielu algorytmów cieni.

    Light cookies zapewniają fajne efekty tanim kosztem wydajnościowym

12. **importer zasobów i menedżer**

    Importuj zasoby: modele 3D i animacje (FBX, OBJ, DAE, 3DS), tekstury i tekstury HDR, pliki audio i inne

13. **potok graficzny liniowy i HDR**

    potok graficzny liniowy i HDR: korekcja gammy, tonemapping, wsparcie dla tekstur sześciennych HDR i mapy świateł

14. **API urządzeń wejścia**

    Wsparcie klawiatury, myszy, gamepada, ekranów dotykowych

15. **renderer fontu SDF**

    Konwertuj TTF, OTF na zasoby fontu (podobnie jak w Unity)

16. **silnik fizyki rigidbody**

    Wbudowany w PlayCanvas system fizyki Ammo, który jest portem Bulleta, pozwala na łatwiejszą implementację fizyki w grze

17. **narzędzia dla responsywnych interfejsów**

    Komponenty pozwalające na tworzenie responsywnych interfejsów 2D i 3D

18. **wsparcie WebVR**

    Wsparcie dla najnowszych standardów WebVR

19. **rozwój i testowanie na urządzeniu mobilnym**

    Szybkie iteracje z użyciem aktualizacji na bieżąco na urządzeniu mobilnym

20. **filtrowanie zasobów**

    Przeszukuj i filtruj swoją kolekcję zasobów

21. **edytowanie scen w czasie rzeczywistym**

    Zmiany na bieżąco w stylu kolaboracji z Google Docs

22. **prefiltering tekstury sześciennej**

Ustaw oświetlenie bazujące na obrazie (IBL) z użyciem tylko jednego kliknięcia przycisku

23. **profiler**

    Wyświetla wykresy, statystyki związane z wydajnością w czasie rzeczywistym

24. **kompresja tekstur (DXT, PVR i ETC1)**

    kompresja tekstur jednym kliknięciem

25. **edytor materiału**

    Szybko dostosowywuj widoczne wizualnie zmiany w parametrach materiałów z wykorzystaniem edytora

26. **wieloplatformowość**

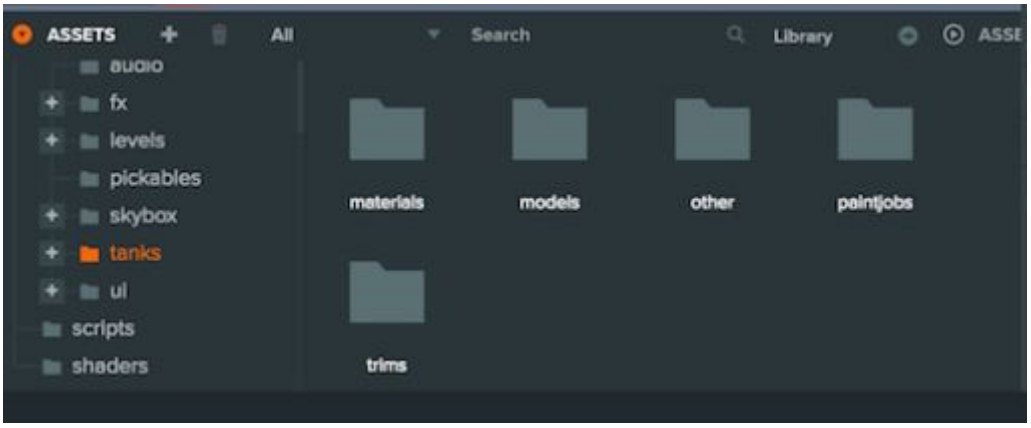
    Uruchom edytor na każdym urządzeniu: desktop, laptop, tablet, smartfon

Jak widzisz PlayCanvas posiada wiele funkcjonalności.

Przejdę teraz do omówienia zasobów.

Rozdział 2

zasoby (assets)



Zasoby mogą być różnego typu np. model, animacja, obrazki dla tekstur (.png,.jpg) i audio.

Poniżej omawiam wszystkie rodzaje zasobów dostępne w PlayCanvas:

- materiał
- Phonga
- fizyczny (physical)
- tekstura
- model
- animacja
- tekstura sześcienna (cubemap)
- HTML
- audio
- CSS
- shader

- font
- duszek (sprite)
- prefab (w pc pod nazwą template)
- moduł Wasm (Wasm module, WebAssembly module)

2.1 materiał

Ogólnie materiał definiuje właściwości powierzchni takie jak kolor, połyskliwość itd. Dokładnie czego to dotyczy odsyłam do podręczników z grafiki komputerowej.
W PlayCanvas materiał to jeden z typu zasobów.
Ma 2 podtypy: Phonga i fizyczny.

Phonga

Model cieniowania Phonga to przestarzały element, zaleca się korzystania z modelu fizycznego.
Więcej na temat modelu cieniowania Phonga możesz znaleźć tutaj Phong Material | Learn PlayCanvas

fizyczny (physical)

Materiał fizyczny reprezentuje zaawansowany model cieniowania wysokiej jakości, dlatego jest zalecany w stosowaniu dla uzyskania imponujących efektów.
Szczegółowe info na temat właściwości materiału fizycznego dostępne jest tutaj Physical Material | Learn PlayCanvas

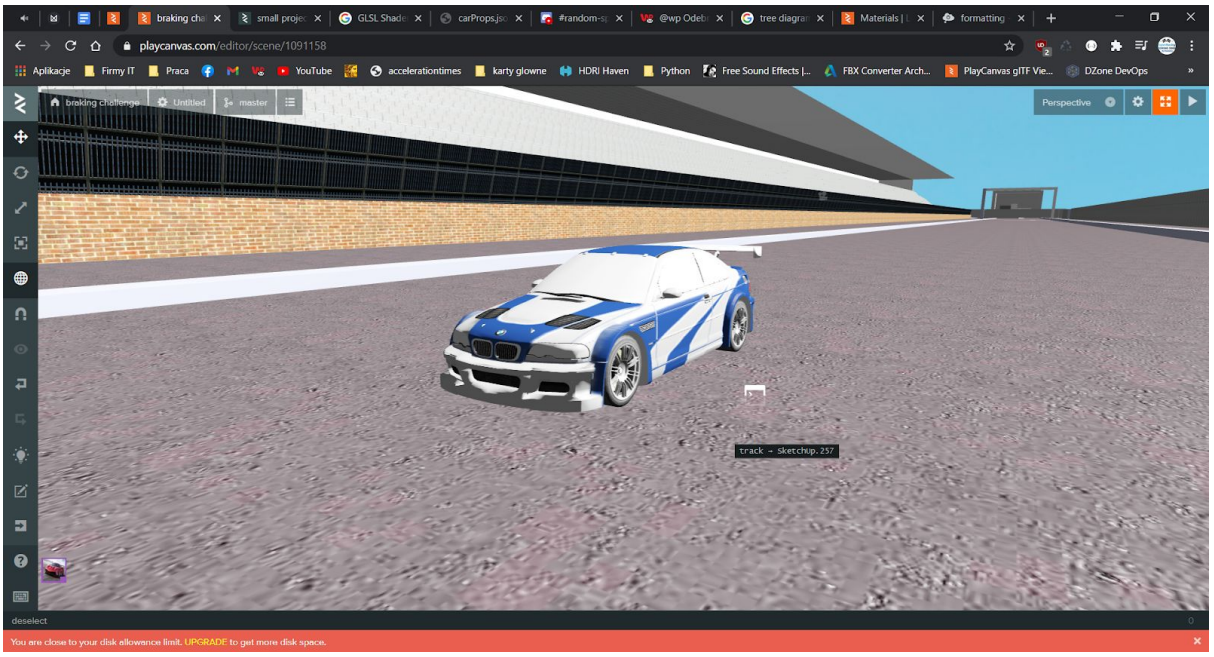
Następujące regiony dotyczące tego materiału: **offset i tiling**, **ambient** (związane z ambient occlussion, okluzją otoczenia), **diffuse** (rozproszenie, diffuse to inaczej albedo), **specular** (daje połyskliwość), **emissive** (emituje światło), **opacity** (przezroczystość), **normals**

(związane z mapą normalną), **parallax** (związane z mapa wysokościową), **environment** (refleksy), **lightmap** (mapa świateł),

2.2 tekstura

Tekstura to obraz, który może zostać przypisany do materiału.
Poniżej wyróżniłem mapy tekstur, które przydaja się przy multiteksturowaniu, aby uzyskać bardziej szczególowy wygląd materiału.
Rodzaje map tekstur: okluzja otoczenia (ambient occlusion, AO map), tekstura sześcienna (cubemap), środowiskowa (env map), rozproszenia (diffuse map), odbicia (specular map), emissive (emissive map), przezroczystości (opacity map), normalna (normal map), wysokościowa (height map), oświetlenia (light map).
Więcej o teksturach tutaj [Textures | Learn PlayCanvas]

2.3 model



Modele 3D i animacje są tworzone poza PlayCanvas, eksportowane np. z Blendera, Wings3D, Maya lub 3DS Max i importowane do PlayCanvas.
Zaleca się korzystać z formatu fbx dla uzyskania najlepszych wyników i tak model zostanie przekonwertowany na glb (tzn fbx pozostanie jako źródłowy format, ale zostanie stworzony glb jako docelowy format i co za tym idzie będą dwa formaty fbx i glb danego modelu).

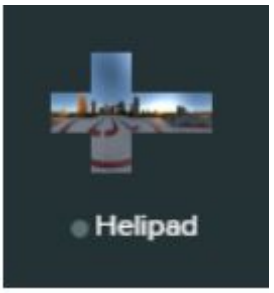
Więcej o modelach [Models | Learn PlayCanvas]

2.4 animacja

Zasób animacja jest używany, aby odtwarzać pojedynczą animację na modelu 3D.
Formaty pełnych scen zawierają animacje, np. jest to gltf, dae, fbx.

Więcej o animacji [Animation | Learn PlayCanvas]

2.5 tekstura sześcienna (cubemap)



Tekstura sześcienna to specjalny typ tekstury składający się z 6 zasobów tekstur. Stosowana jest jako skybox lub mapa środowiska.

Więcej o cubemap [Cubemaps | Learn PlayCanvas]

2.6 HTML

Zasób HTML zawiera kod HTML.

Aby wczytać HTML musisz napisać taki kawałek kodu js:

```
this.element = document.createElement('div');
this.element.classList.add('container');
document.body.appendChild(this.element);
this.element.innerHTML = this.html.resource;
```

Teraz opiszę szybko działanie kodu.

Tworzy element div dynamicznie, później dodaje klasę o nazwie container. Następnie podczepia tego diva do <body> i ustawia zawartość twojego htmla jako element potomny dla div'a container.

To jest jeden ze sposobów.
Oczywiście musisz jeszcze w tym przypadku dodać atrybut z nazwą **html** (jeżeli w kodzie nazwałeś jako this.html, o atrybutach później), napisać kod html, przeciągnąć plik html w edytorze do miejsca gdzie można podczepiać czy to encje czy to różne zasoby, w tym przypadku jest to ui pod komponentem script, w ui jest właśnie atrybut typu zasób o nazwie html, dopiero wtedy masz zawartość HTML na swojej stronie.

Więcej o HTML
HTML | Learn PlayCanvas

2.7 audio

Zasób audio to plik dźwiękowy.
Audio | Learn PlayCanvas

2.8 CSS

Zasób CSS zawiera kod CSS.
Styl CSS podczepia się do strony podobnie jak w przypadku zasobu HTML, czyli dodajesz atrybut o nazwie css, tworzysz kod CSS, przeciągasz plik css w edytorze do miejsca gdzie można podczepiać czy to encje czy to różne zasoby, czyli np. ui pod komponentem script, w ui jest właśnie atrybut typu zasób o nazwie css, dopiero wtedy masz zawartość CSS na swojej stronie, a więc zastosowany wygląd.
Kod do podczepiania CSSa jest trochę inny niż było to przy zasobie HTML.
Tym razem pokażę trochę inny sposób:

```
// get asset from registry by id
const asset = app.assets.get(32);

// create element
const style = pc.createStyle(asset.resource || '');
document.head.appendChild(style);

// when asset resource loads/changes,
// update html of element
asset.on('load', function() {
    style.innerHTML = asset.resource;
});
```

```
// make sure assets loads
app.assets.load(asset);
```

Więc tak pobierany jest zasób CSS z rejestru zasobów, tworzony element i wczytywany zasób.

2.9 shader

Zasób shader zawiera kod GLSL, możesz też przesłać pliki z rozszerzeniem .vert, .frag lub .glsl

```
const vertexShader = this.app.assets.find('my_vertex_shader');
const fragmentShader = this.app.assets.find('my_fragment_shader');
const shaderDefinition = {
    attributes: {
        aPosition: pc.SEMANTIC_POSITION,
        aUv0: pc.SEMANTIC_TEXCOORD0
    },
    vshader: vertexShader.resource,
    fshader: fragmentShader.resource
};

const shader = new pc.Shader(this.app.graphicsDevice, shaderDefinition);
const material = new pc.Material();
material.setShader(shader);
```

Dwie pierwsze linijki dotyczą szukania w rejestrze zasobów shadera wierzchołków i fragmentów.
Dalej zdefiniowany jest shader z atrybutami: pozycja i uv. Do właściwości vshader i fshader doczepiana jest zawartość zasobu shader, najpierw shadera wierzchołków, później fragmentów. Jako przedostatni krok tworzony jest shader i materiał. Wreszcie ustawiany jest shader na materiał.

2.10 font

Zasób fontu zawiera obrazek z wszystkimi znakami fontu, Używa się go do wyświetlenia tekstu.
Więcej o font tutaj Fonts | Learn PlayCanvas

2.11 duszek (sprite)

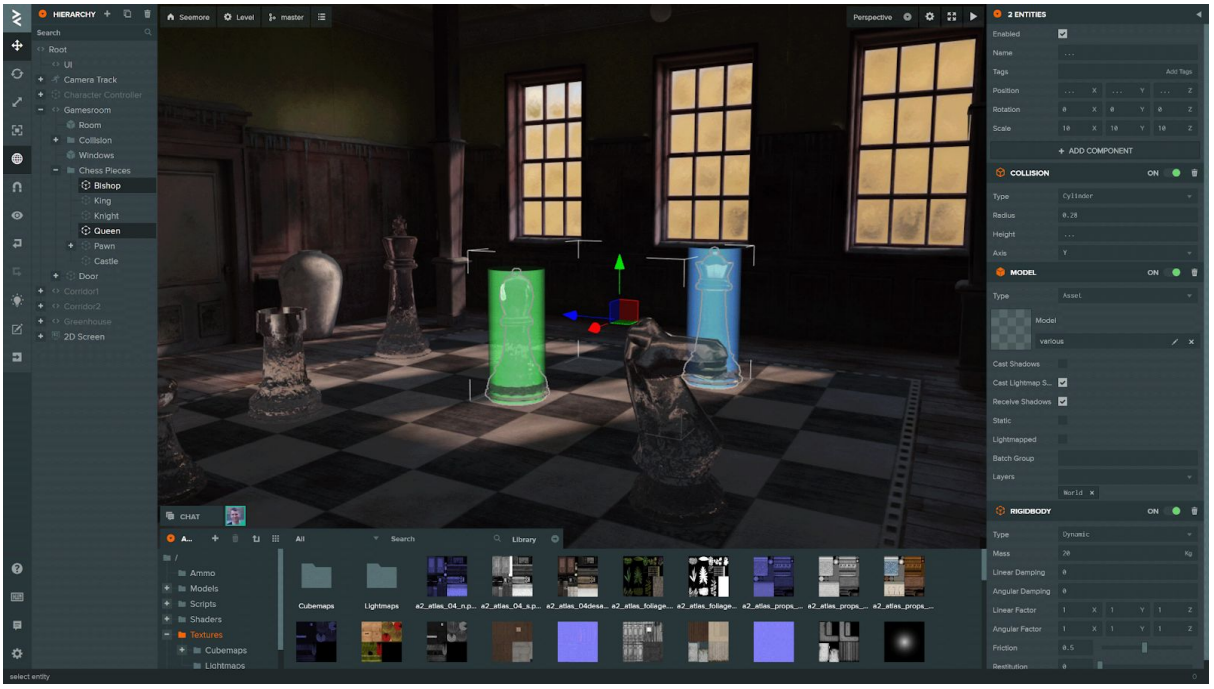
Sprite to grafika 2D, ze względu na to że książka dotyczy tworzenia gry w 3D, temat 2D zostaje pominięty
Więcej o sprite Sprite | Learn PlayCanvas

2.12 prefab (w pc pod nazwą template)

Prefabrykat, czyli zasób zawierający część encji, pozwalający na tworzenie wielu instancji, dlatego przydatny jest przy konstruowaniu obiektów wyglądających tak samo np. 1000 drzew jednego typu, 10 takich samych przycisków itd. W PlayCanvas nie ma jeszcze wariantów prefabów, czyli np. jest bazowy prefab samochód, a np. chcę mieć różne samochody mające te same cechy, ale inne wartości, np. prędkość maksymalna lub przyspieszenie.
Więcej o prefabach Template | Learn PlayCanvas

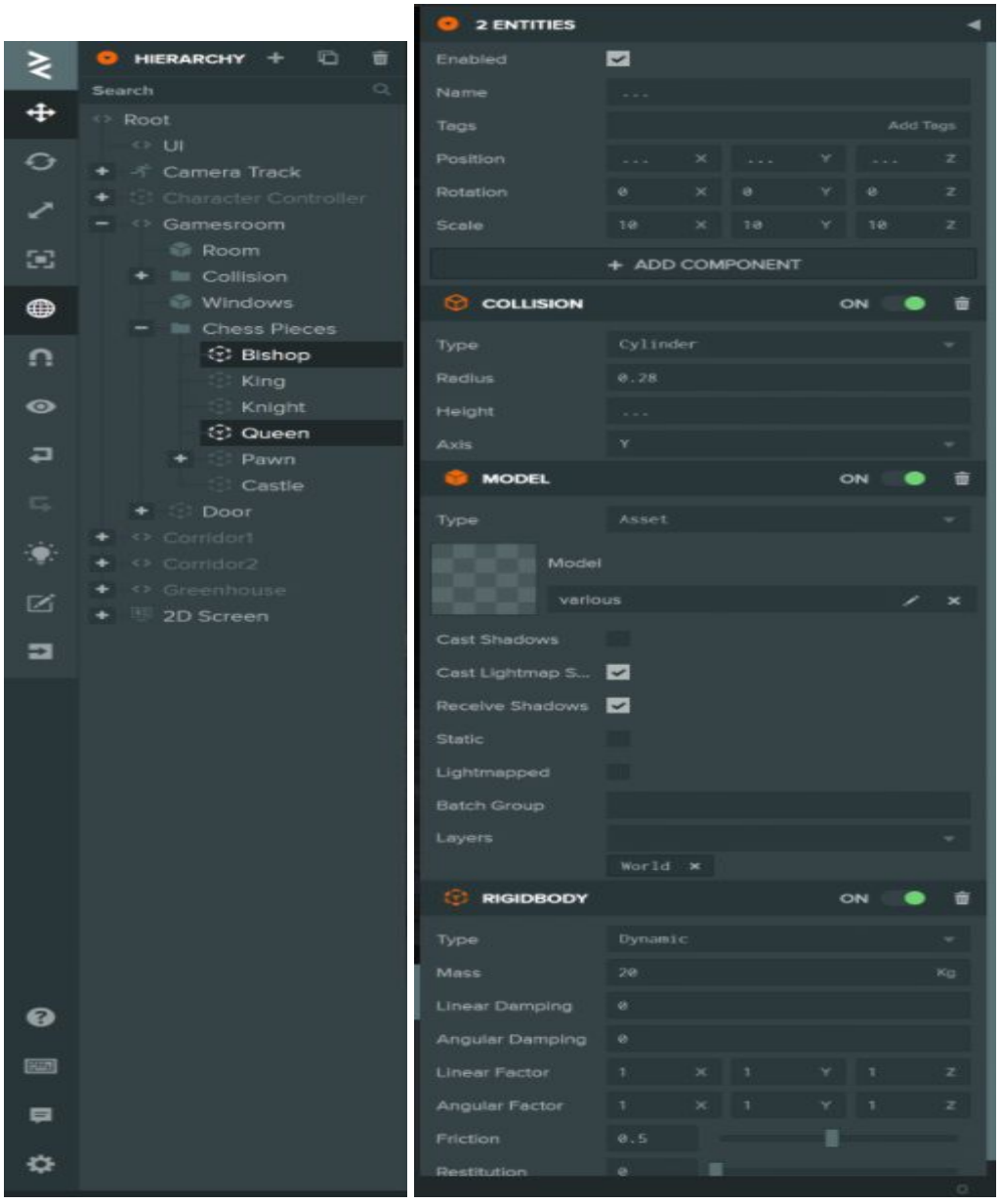
Pominąłem temat zasobu Wasm. Myślę, że w miarę ok omówiłem możliwe zasoby w PlayCanvas.
Przejdę do części edytor.

Część II - edytor

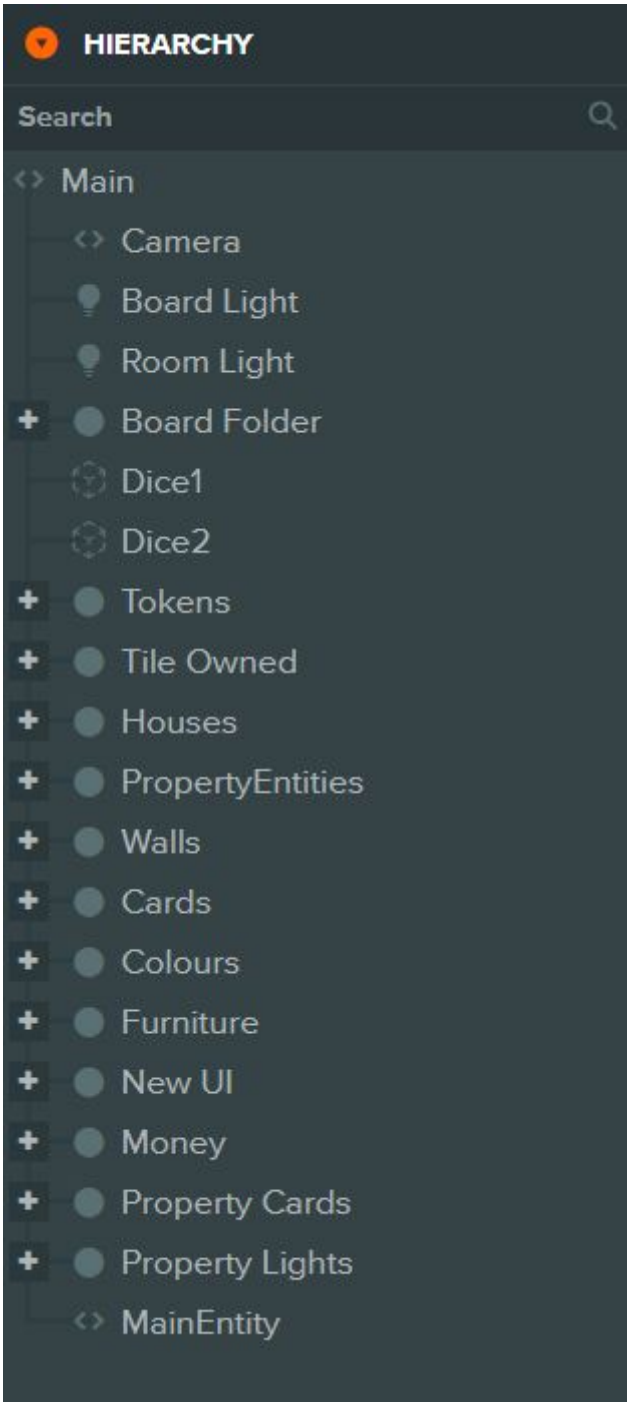


Rozdział 3

UI



3.1 Panel hierarchia



Panel hierarchii zawiera graf sceny, widok drzewa. Graf ten składa się z korzenia, domyślnie nazywa się on Root, w tym przypadku na rysunku nazywa się **Main**, może też nazywać się Game.

Main w tym przypadku jest rodzicem takich encji jak:

Camera, Board i Room Light, Board Folder, Dice1, Dice2, Tokens, Tile Owned, Houses, PropertyEntities, Walls, Cards, Colours, Furniture, New UI, Money, Property Cards, Property Lights i MainEntity.

Jest to fragment gry planszowej Monopoly.

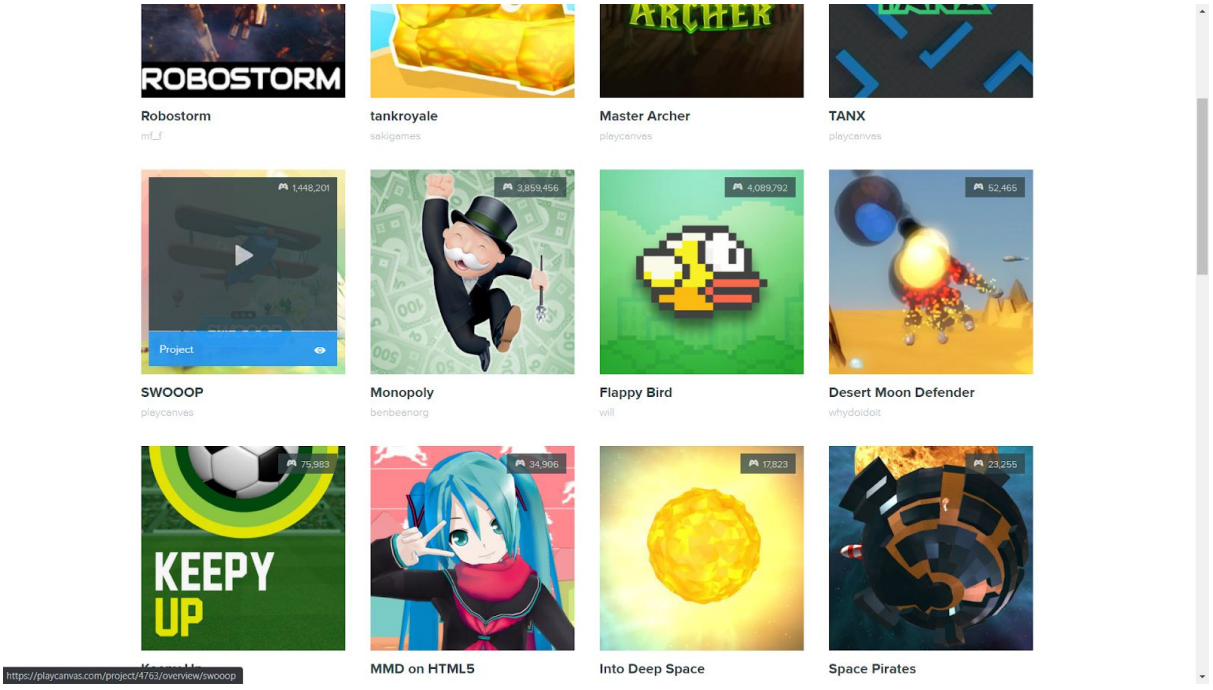
Dodatkowo te encje są rodzicami (wskazuje na to znaczek plusa) innych encji itd.

Jak widzisz ta struktura jest bardzo złożona, dlatego tak ważne jest grupowanie obiektów, np. jak to zostało przedstawione na rysunku. Grupowanie obiektów to jedna z dobrych praktyk.

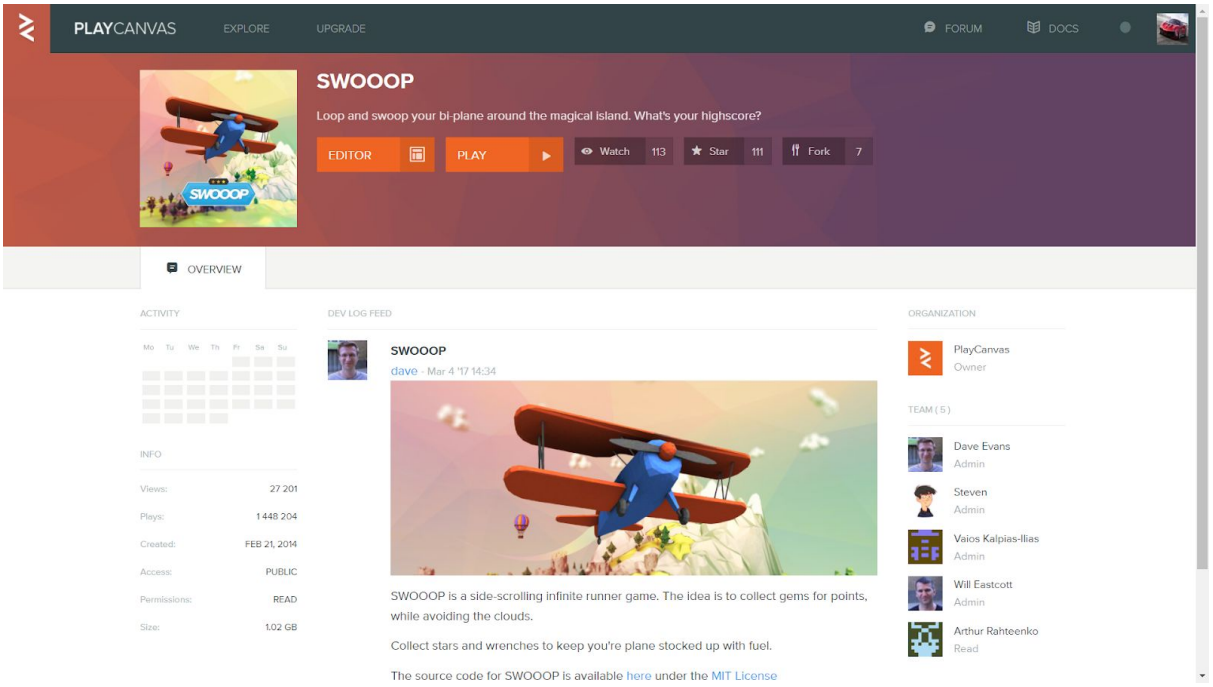
Hierarchiczna struktura pozwala na dobrą organizację elementów gry.

Jako mała dygresja: dlatego popatrz jak to jest realizowane na innych przykładach gier stworzonych w PlayCanvas, przejdź do strony PlayCanvas (musisz być zalogowany, zarejestrowany aby widzieć zawartość EXPLORE, gdy już się zalogujesz przejdź do explore, zobaczysz tam różne projekty, kliknij na **Project** przy danym projekcie.

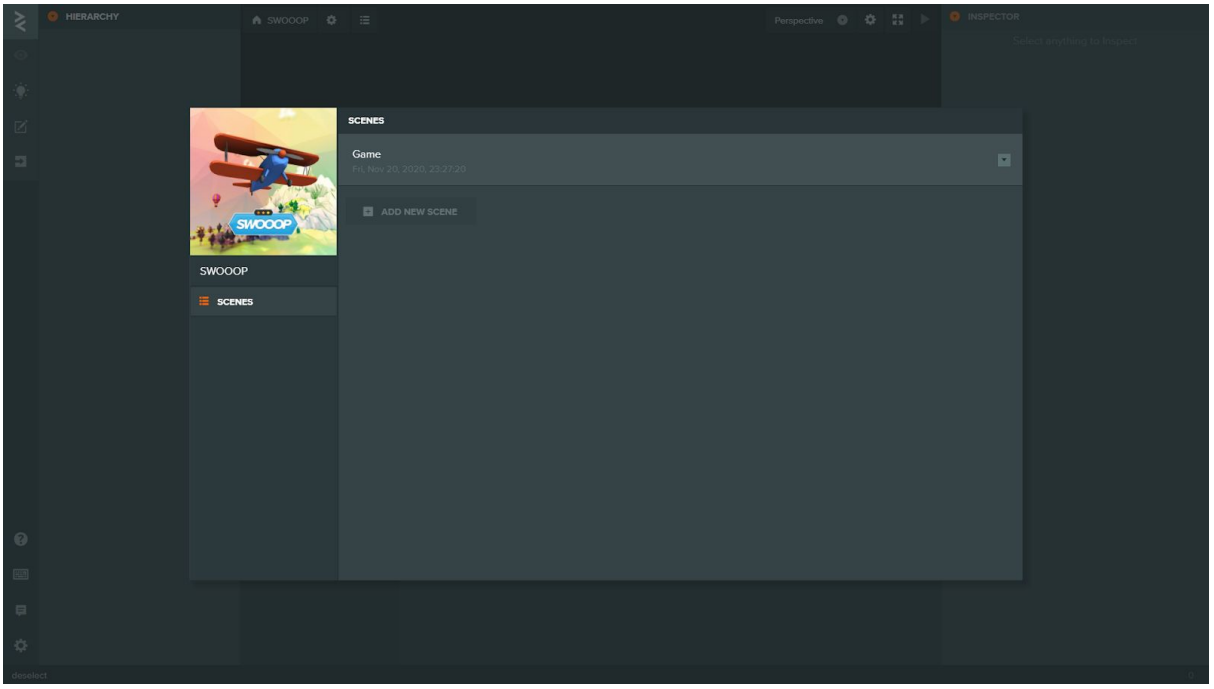
Ja wybrałem SWOOP, jest to gra typu endless runner.



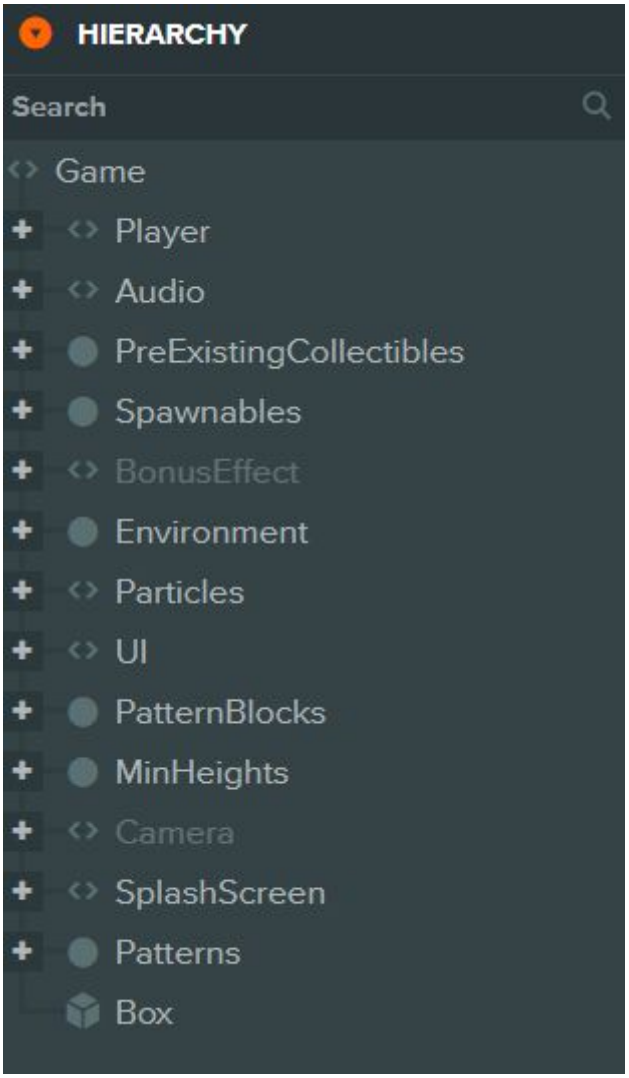
, przejdzie to do dalej overview projektu



kliknij **EDITOR**,

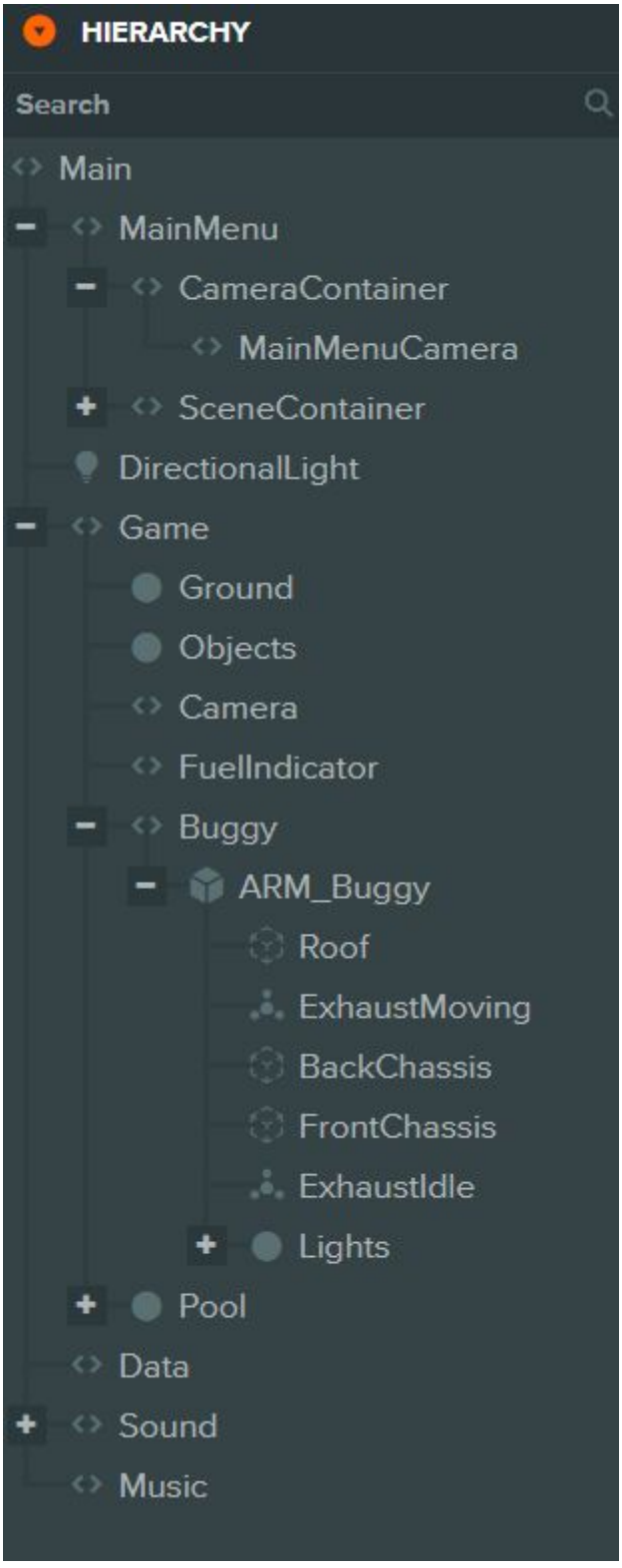


kliknij na scenę w tym przypadku Game i już możesz zobaczyć hierarchię.



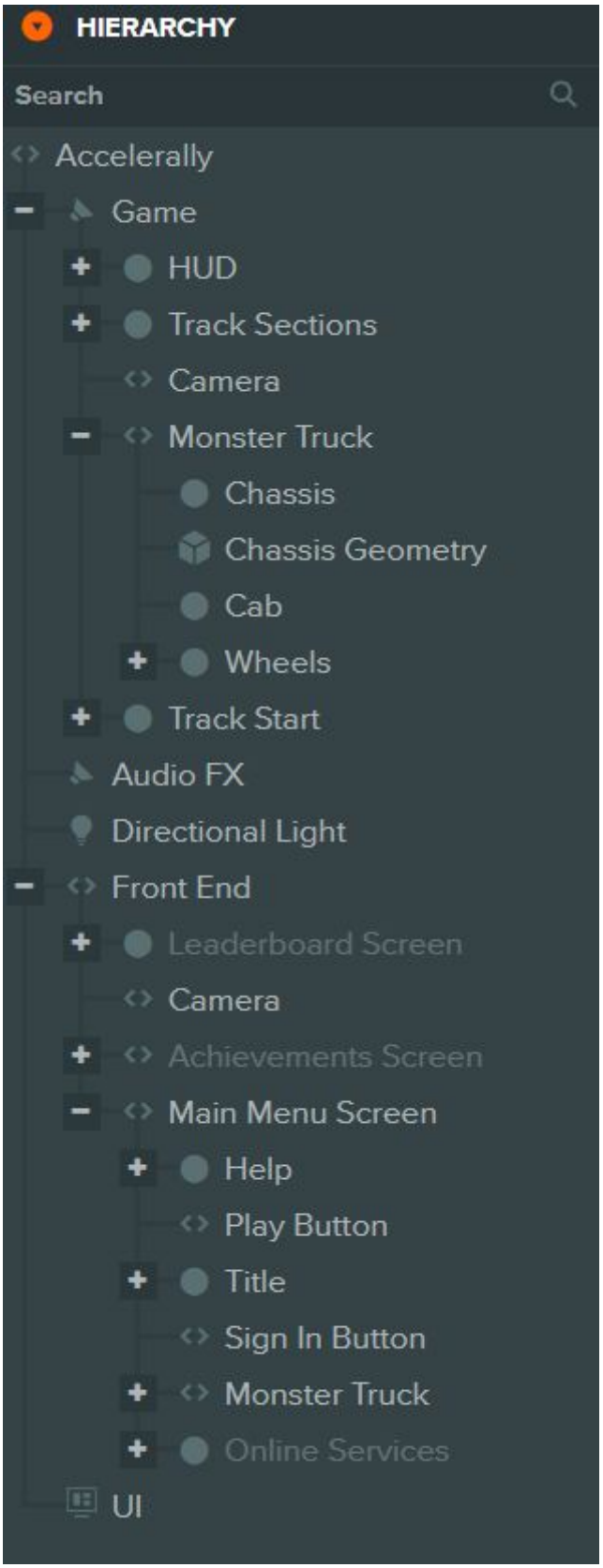
Pokażę jeszcze parę innych hierarchii

np. hierarchię ze Space Buggy

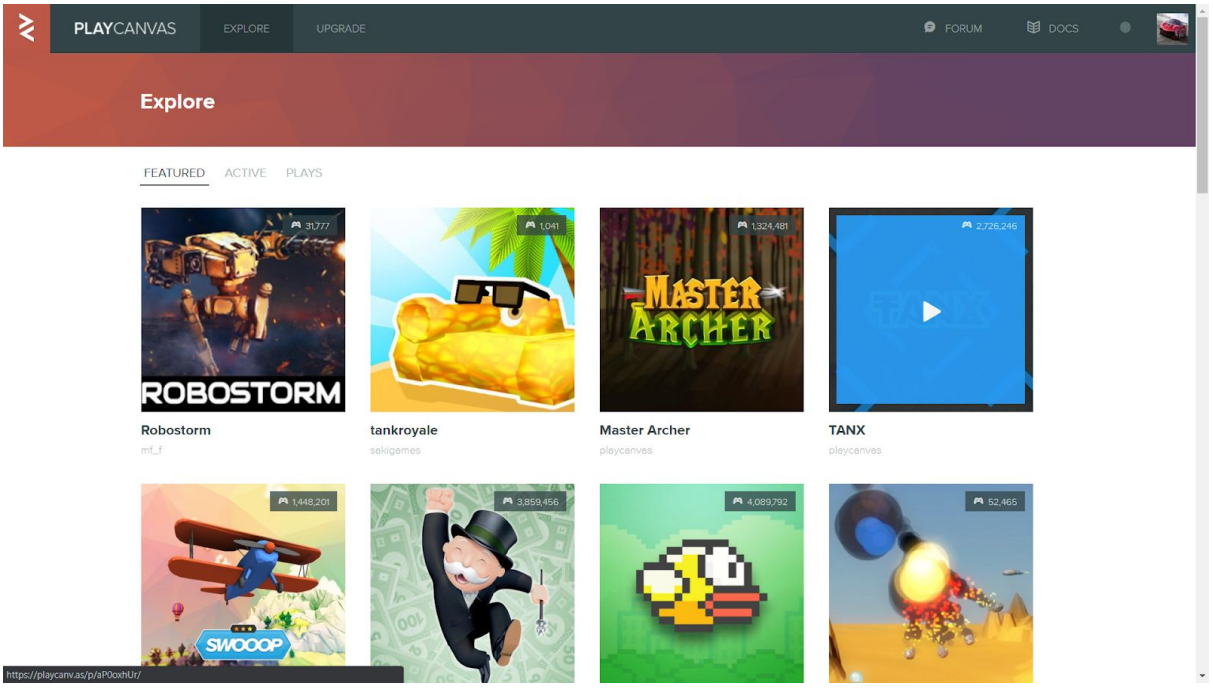


Ciężko jest uchwycić na obrazku całą, rozwiniętą hierarchię.

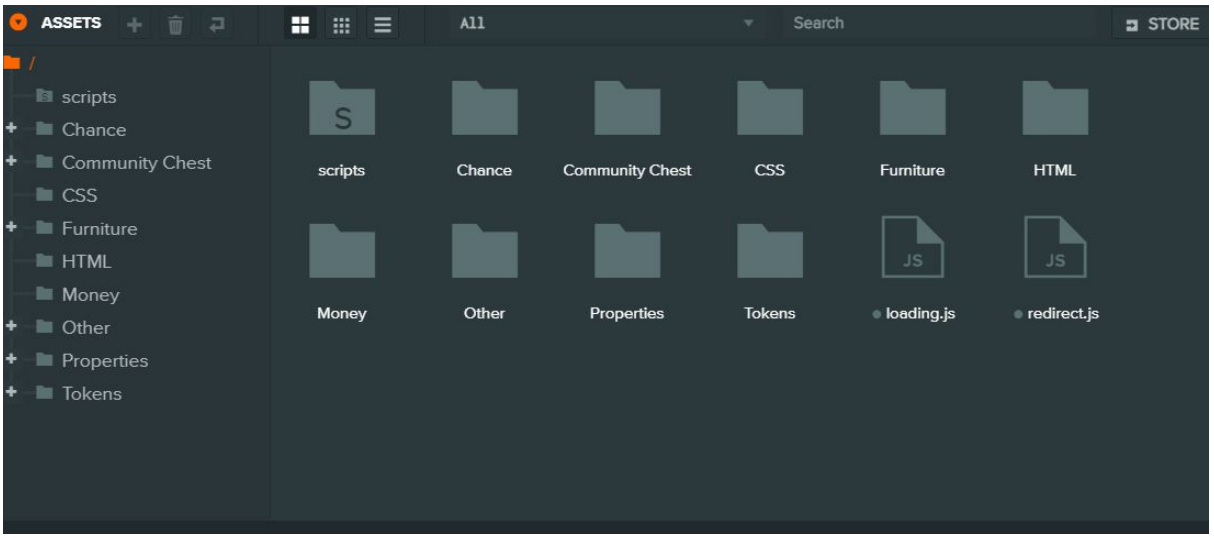
Pokażę jeszcze jedną hierarchię z gry **Accelerally** i na tym zakończę o hierarchiach.



Niektóre projekty mają zablokowaną opcję **Project** (np. TANX), można tylko nacisnąć PLAY aby zagrać w grę , rysunek poniżej.


PLAYCANVAS
EXPLORE
UPGRADE
FORUM
DOCS
Explore
FEATURED
ACTIVE
PLAYS
31,777
ROBOSTORM
Robostorm
mf_f
1,041
tankroyale
sakigames
1,324,481
MASTER ARCHER
Master Archer
playcanvas
2,726,246
TANX
playcanvas
1,448,201
SWOOOP
3,859,456
4,089,792
52,465
https://playcanv.as/p/aP0oxhUr/

3.2 Panel Zasoby



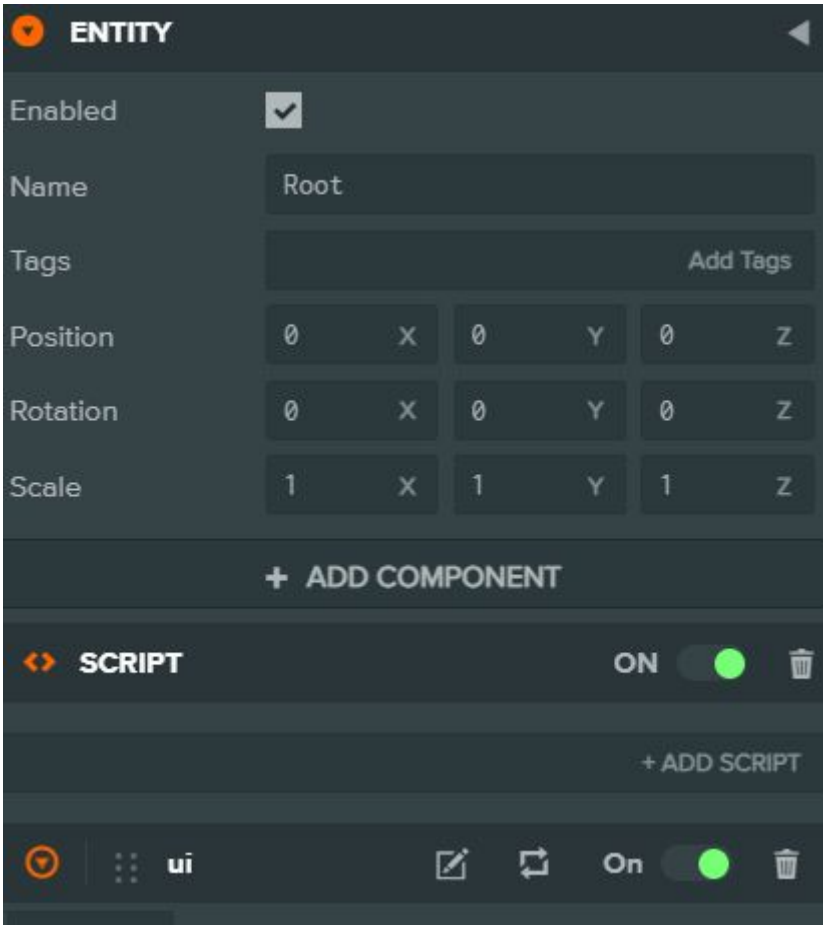
Zasoby jest najlepiej organizować w folderach, np. skrypty w **scripts**, materiały w **materials**, modele w **models**, tekstury w **textures** itp.

Po lewej na rysunku widzisz strukturę folderów: / to korzeń (root) w nim znajdują się foldery w tym przypadku: scripts, Chance, Community Chest, CSS, Furniture, HTML, Money, Other, Properties, Tokens, podobnie tak jak to masz zorganizowane w systemie plików na systemie operacyjnym.

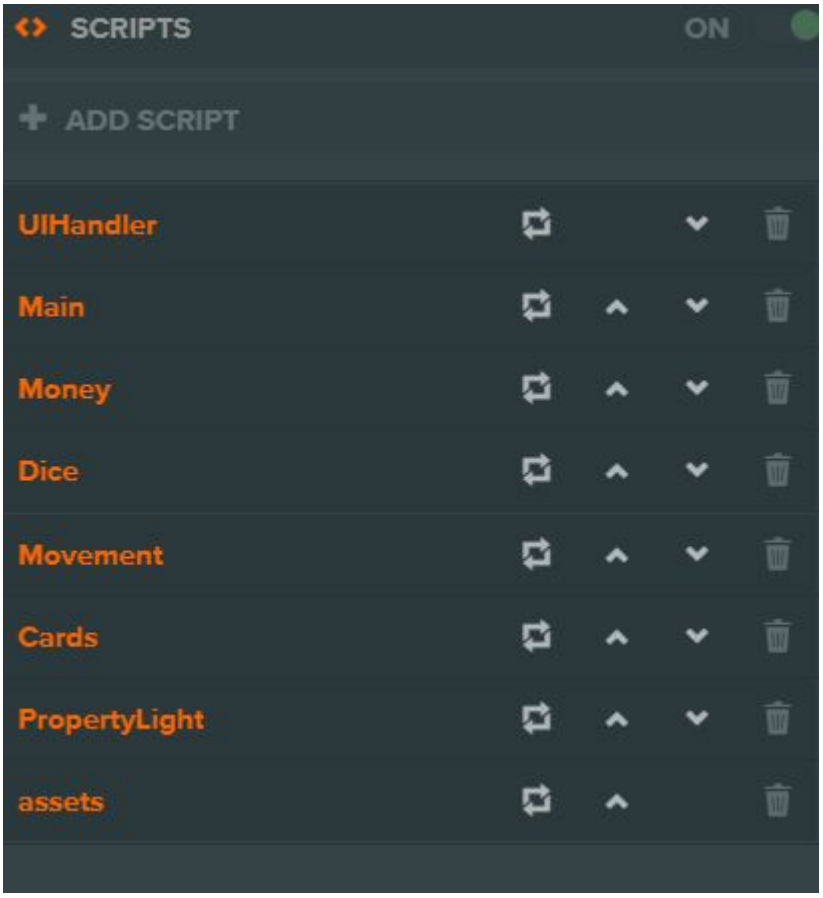
Po prawej widać foldery i pliki, foldery wyżej wymienione, 2 pliki: loading.js i redirect.js

Tutaj możesz przesłać swoje zasoby poprzez upload, możesz przefiltrować kategoriami (tu gdzie jest All), wyszukać dany zasób (Search), dodać nowy zasób lub usunąć istniejący, możesz też wejść na PlayCanvas Store.

3.3 Panel inspektor



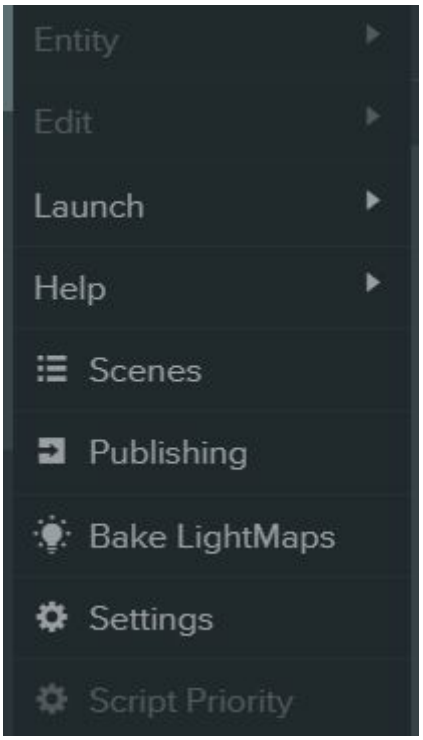
Tutaj możesz włączyć / wyłączyć encję (Enabled), nazwać encję (Name)
dodać tagi (Tags), ustawić transformację: pozycję (position), orientację (rotation) i rozmiar (scale).
Co ważne wszystkie te własności są w przestrzeni **lokalnej**, modelu (local space, model space).
Orientację ustawia się przy pomocy tak zwanych kątów Eulera.
Możesz też dodać komponenty (+ add component).
W tym przypadku dodanym komponentem jest komponent script w nim znajduje się kod ui.js.
Encja może mieć wiele doczepionych skryptów js, np. UIHandler, Main, Money, Dice, Movement, Cards, PropertyLight, assets,
jak to zostało pokazane na rysunku.



To tyle na temat inspektora, zostało jeszcze do mówienia menu i toolbar.

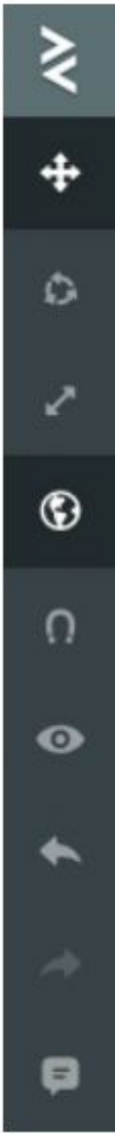
Przejdę do menu.

3.4 Panel Menu

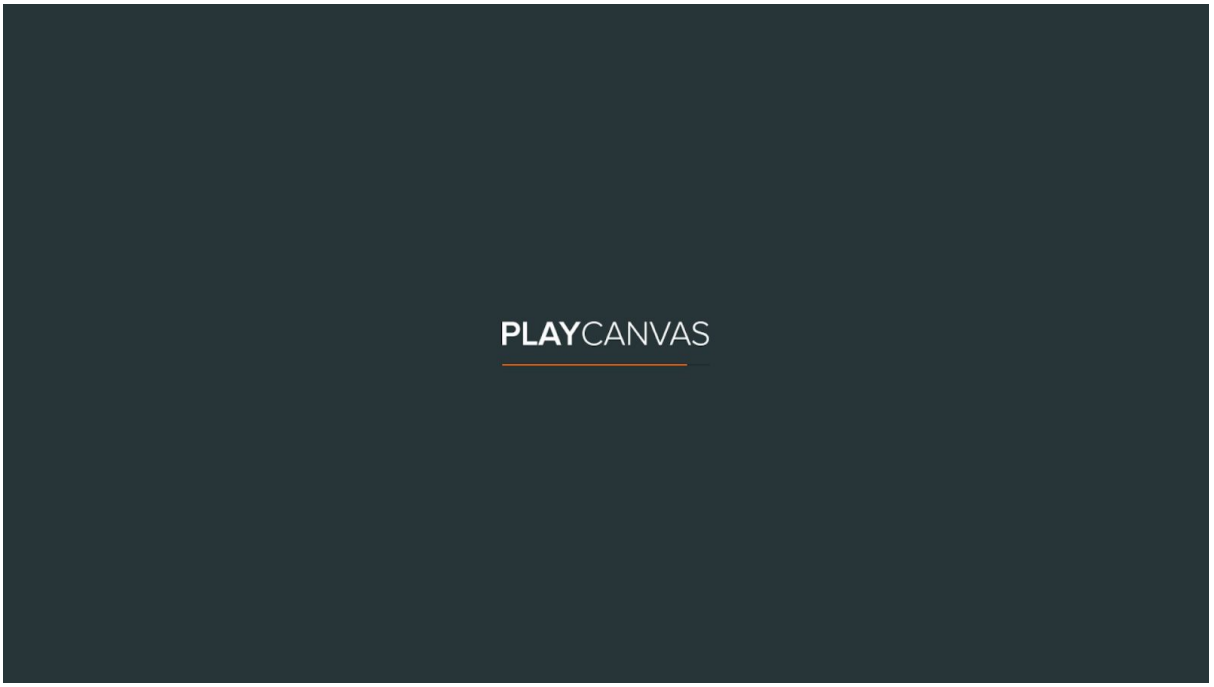


Menu można pokazać klikając na przycisk z ikoną PlayCanvas.
Menu zawiera listę wszystkich poleceń, które możesz wykonać na scenie.
Tutaj można zrobić następujące rzeczy, dodać encję, edytować, uruchomić grę, dostać się do pomocy, wyświetlić listę dostępnych scen, opublikować grę, wypalić mapę świateł, otworzyć ustawienia, ustawić priorytet wykonywania skryptów.
Ogólnie jest to skrót jeżeli nie możesz znaleźć przycisku lub nie pamiętasz skrótu klawiszowego.

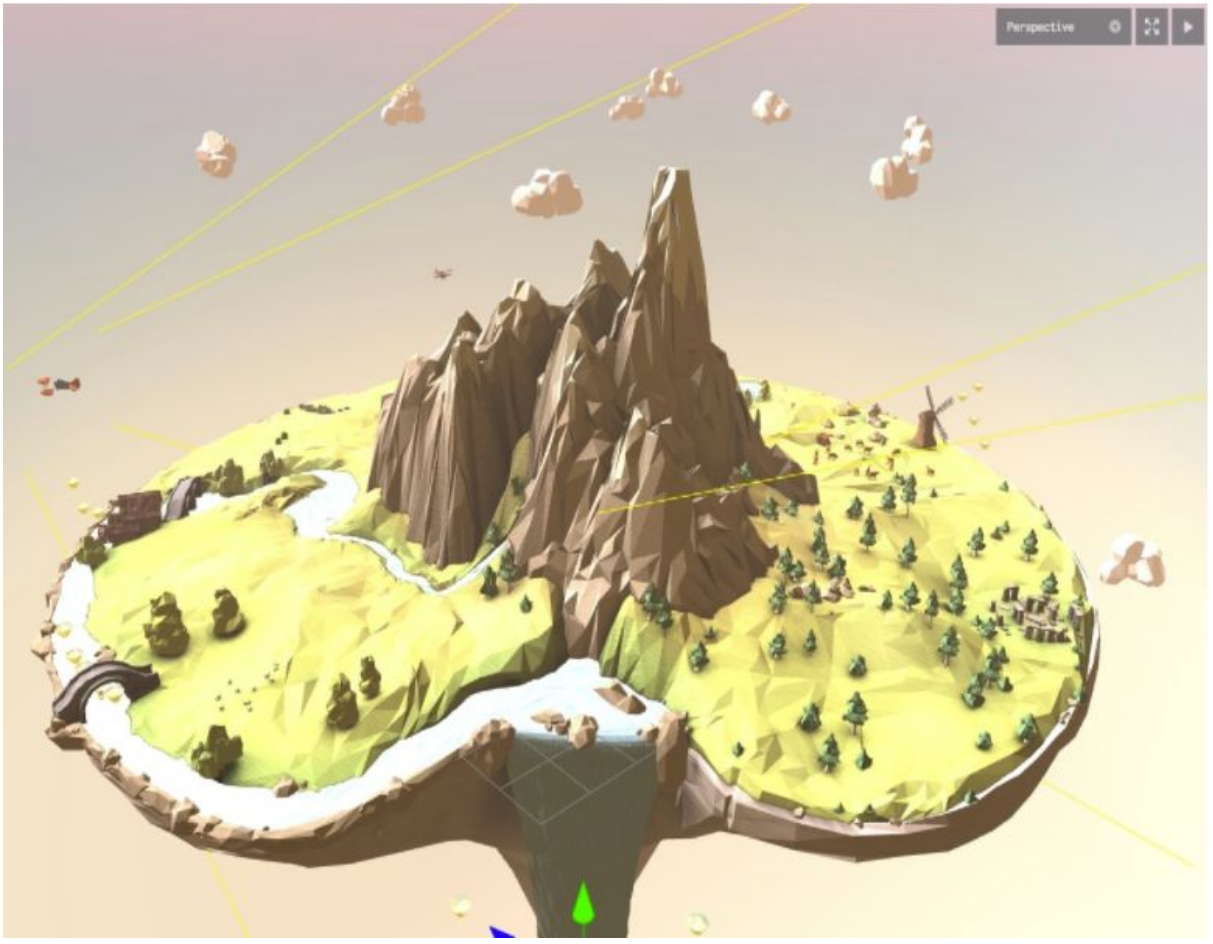
3.5 Panel Toolbar



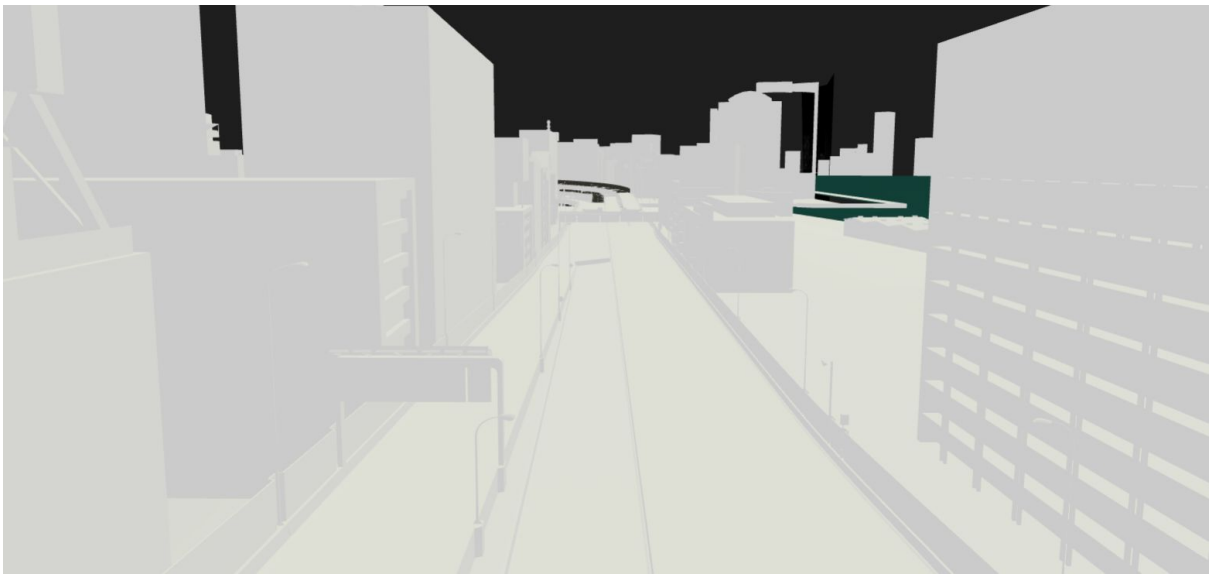
Panel toolbar, pasek narzędzi zawiera najczęstsze polecenia dostępne w wygodny sposób. Najbardziej przydatny jest przycisk uruchom (skrót ctrl+enter), który włącza grę w nowej karcie, wczytywana jest scena na której się znajdujesz i po załadowaniu możesz testować, grać.



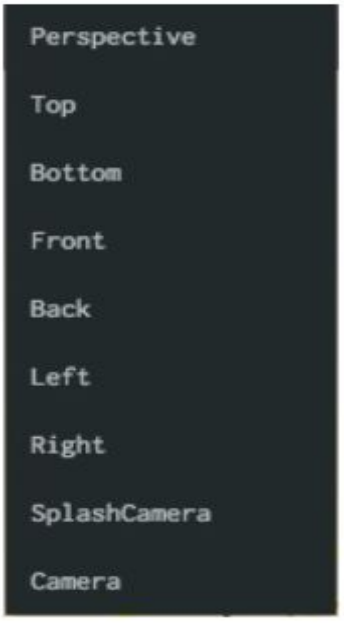
3.6 Viewport



Viewport pokazuje scenę aktualnie wyświetlaną. Możesz poruszać się po scenie z klawiszami WASD i strzałkami góra, dół, lewo, prawo. Trzymając shift przyspieszasz prędkość kamery, możesz w ten sposób przeglądać scenę szybciej, jeżeli np. przestrzeń jest duża, np. tu.
Jest to model miasta, mod do gry Assetto Corsa (nie zwracaj uwagi na brak tekstur), tutaj bardzo przydatny jest sposób szybkiego przeglądania sceny.

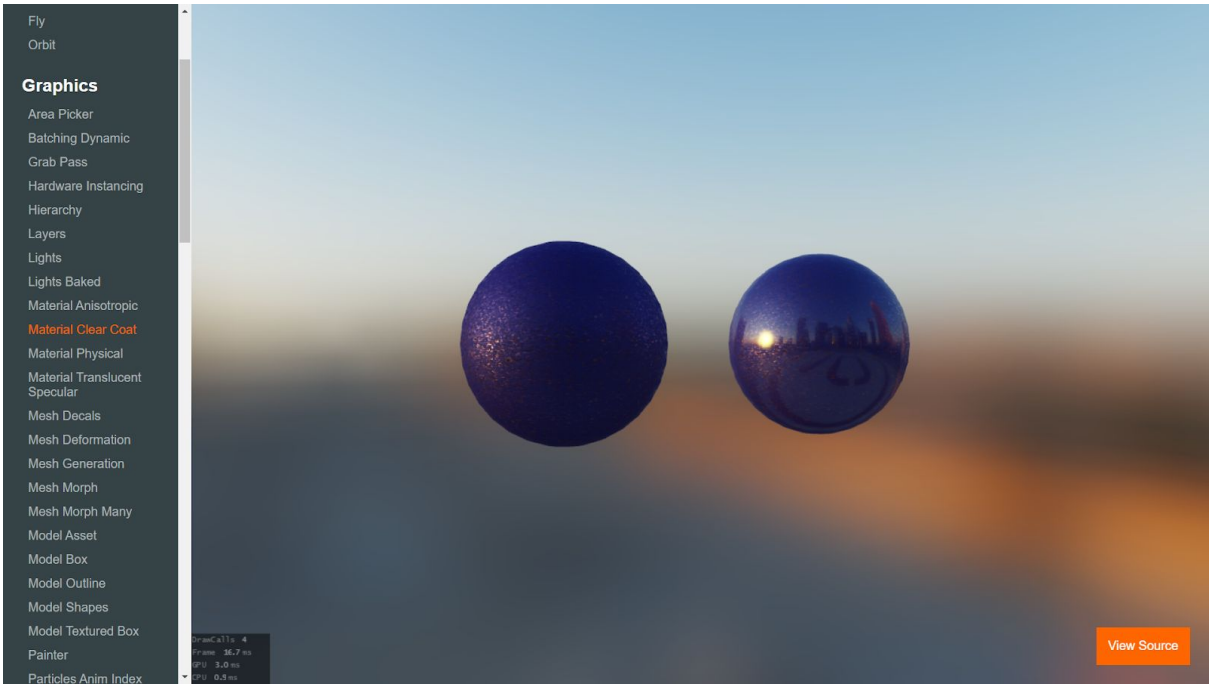


Wracając do viewportu możesz ustawić kamerę na perspektywę lub ortograficzną, w przypadku orto są to widoki lewo, prawo, góra, dół, przód, tył (left, right, top, bottom, front, back). Jeszcze możesz zmienić na widok z innej kamery, np. kamery śledzącej (jeżeli taką ustawiłeś w hierarchii). Na tym przykładzie dodatkowe kamery to SplashCamera i Camera. Na marginesie kamery to po prostu macierze.



Teraz przejdę do omówienia części dotyczącej silnika.

Część III - silnik



Rozdział 4

skrypty

```
/*jshint esversion: 6 */

class Ui extends pc.ScriptType {

    // initialize code called once per entity
    initialize() {
        this.initHTML();
    }
    initHTML(){
        this.element = document.createElement('div');
        this.element.classList.add('container');
        document.body.appendChild(this.element);
        this.element.innerHTML = this.html.resource;
    }

    // update code called every frame
    update(dt) {

    }

}

pc.registerScript(Ui, 'ui');

Ui.attributes.add('html', {type: 'asset'});

// swap method called for script hot-reloading
// inherit your script state here
// Ui.prototype.swap = function(old) { };

// to learn more about script anatomy, please read:
// http://developer.playcanvas.com/en/user-manual/scripting/
```

3.1 inicjalizacja i aktualizowanie

initialize()

update()

3.2 atrybuty aka `attributes.add()`

atrybuty:

- encji
- zasobu
    - typ zasobu
        - tekstura
        - model
        - materiał
- koloru
- krzywej
- wyliczenia
- JSON

3.3 Komunikacja - zdarzenia

zdarzenia skryptA-skryptB

zdarzenia aplikacji

Rozdział 5

Grafika

5.1 Kamera

5.2 Oświetlenie

5.3 PBR

5.4 System cząsteczek

5.5 Przetwarzanie końcowe

5.6

Rozdział 6

API omówienie

Application

pc.app

assets - rejestr zasobów (AssetRegistry)

scene - scena

root - korzeń

input:

keyboard - klawiatura

mouse - mysz

touch - urządzenie mobilne

gamepad

Część IV - Encja - obiekt gry

Rozdział 7

Encja, Komponenty i Systemy

Encja

Komponenty (18)

- Animation
- Audio Listener
- Button
- Camera
- Collision
- Element
- Layout Child
- Layout Group
- Light
- Model
- Particle System (system cząsteczek)
- Rigid Body (bryła sztywna)
- Screen
- Script
- Scrollbar
- Scrollview
- Sound
- Sprite (duszek)

Systemy

Część V - praktyka i przykłady

Rozdział 8

projekt - braking challenge

Rozdział 9

PlayCanvas - przykłady

After the Flood

Casino

Dodatek A

API

pc

callbacks

guid

math

path

platform

script

string

Animation

Animation

AnimationComponent

AnimationComponentSystem

AnimationHandler

Asset

Asset

AssetReference

AssetRegistry

Audio

AudioHandler

AudioListenerComponent

AudioListenerComponentSystem

Batch

Batch

BatchGroup

BatchManager

Component

Component

ComponentSystem

ComponentSystemRegistry

Element

ElementComponent

ElementComponentSystem

ElementDragHelper

ElementInput

ElementInputEvent

ElementMouseEvent

ElementTouchEvent

Layout

LayoutChildComponent

LayoutChildComponentSystem

LayoutGroupComponent

LayoutGroupComponentSystem

Model

Model

ModelComponent

ModelComponentSystem

ModelHandler

Morph

Morph

MorphInstance

MorphTarget

Script

ScriptAttributes

ScriptComponent

ScriptComponentSystem

ScriptHandler

ScriptRegistry

ScriptType

Sound

Sound

SoundComponent

SoundComponentSystem

SoundInstance

SoundInstance3d

SoundManager

SoundSlot

Sprite

Sprite

SpriteAnimationClip

SpriteComponent

SpriteComponentSystem

SpriteHandler

Texture

Texture

TextureAtlas

TextureAtlasHandler

TextureHandler

Touch

Touch

TouchDevice

TouchEvent

Vec

Vec2

Vec3

Vec4

Vertex

VertexBuffer

VertexFormat

VertexIterator

XR

XrHitTest

XrHitTestSource

XrInput

XrInputSource

XrManager

pozostałe

Application

BasicMaterial

bounding

BoundingBox

BoundingSphere

button

ButtonComponent

ButtonComponentSystem

camera

CameraComponent

CameraComponentSystem

collision

CollisionComponent

CollisionComponentSystem

Color

contact

ContactPoint

ContactResult

container

ContainerHandler

ContainerResource

Controller

CubemapHandler

curve

Curve

CurveSet

Entity

EventHandler

font

Font

FontHandler

ForwardRenderer

Frustum

GamePads

GraphicsDevice

GraphNode

Http

I18n

IndexBuffer

keyboard

Keyboard

KeyboardEvent

layer

Layer

LayerComposition

light

LightComponent

LightComponentSystem

Lightmapper

Mat

Mat3

Mat4

Material

Material

MaterialHandler

Mesh

Mesh

MeshInstance

Mouse

Mouse

MouseEvent

Node

OrientedBox

particle system

ParticleSystemComponent

ParticleSystemComponentSystem

Picker

post effect

PostEffect

PostEffectQueue

Quat

ray

Ray

RaycastResult

RenderTarget

resource

ResourceHandler

ResourceLoader

rigidbody

RigidBodyComponent

RigidBodyComponentSystem

scene

Scene

SceneHandler

scope

ScopeId

ScopeSpace

screen

ScreenComponent

ScreenComponentSystem

scrollbar

ScrollbarComponent

ScrollbarComponentSystem

scrollview

ScrollViewComponent

ScrollViewComponentSystem

Shader

SingleContactResult

Skeleton

skin

Skin

SkinInstance

StandardMaterial

StencilParameters

Tags

TransformFeedback

stałe

Dodatek B

PlayCanvas przykłady

PlayCanvas Examples

Animacja

- Blend
- Tweening

Kamera

- First Person
- Fly
- Orbit

Grafika

- Area Picker
- Batching Dynamic
- Grab Pass
- Hardware Instancing
- Hierarchy
- Layers
- Lights
- Lights Baked
- Material Anisotropic
- Material Clear Coat
- Material Physical
- Material Translucent Specular

- Mesh Decals
- Mesh Deformation
- Mesh Generation
- Mesh Morph
- Mesh Morph Many
- Model Asset
- Model Box
- Model Outline
- Model Shapes
- Model Textured Box
- Painter
- Particles Anim Index
- Particles Random Sprites
- Particles Snow
- Particles Sparks
- Point Cloud
- Point Cloud Simulation
- Portal
- Post Effects
- Render To Cubemap
- Render To Texture
- Shader Burn
- Shader Toon
- Shader Wobble
- Texture Basis
- Transform Feedback

Loadery

- Loader Glb
- Loader Obj

urządzenia wejścia

- Gamepad
- Keyboard
- Mouse

Różne

- Mini Stats
- Multi Application

Fizyka

- Compound Collision
- Falling Shapes
- Raycast
- Vehicle

Dźwięk

- Positional

Spine

- Alien
- Dragon
- Goblins
- Hero
- Spineboy

interfejs użytkownika

- Button Basic
- Button Particle
- Button Sprite
- Scroll View
- Text Basic
- Text Canvas Font
- Text Drop Shadow
- Text Localization
- Text Markup
- Text Outline
- Text Typewriter
- Text Wrap
- Various

mieszana rzeczywistość (vr, ar)

- Ar Basic

- Ar Hit Test
- Vr Basic
- Vr Controllers
- Vr Hands
- Vr Movement
- Xr Picking