

**Uwaga - jak poprzednio całość piszemy w js, bez użycia HTML-a**

### 1. Kolizja z terenem - raycaster

```
var ray = new THREE.Ray(player.position, player.getWorldDirection())

raycaster.ray = ray

var intersects = raycaster.intersectObject(mesh_terenu);
if (intersects[0]) {
  console.log(intersects[0].distance) // odległość od vertexa na wprost
  console.log(intersects[0].point) // współrzędne vertexa na wprost
}
```

### 2. KokpitScreen.js - ekran ze wskaźnikami

**Uwaga: stosujemy jak najmniejsze wielkości Canvasa, w związku z wydajnością**

utworzenie canvasa

```
var canvas, context;
//
canvas = document.createElement("canvas");
canvas.width = ? // bez px;
canvas.height = ? // bez px;
canvas.style.position = "absolute";
canvas.style.left = "?px";
canvas.style.top = "?px";
canvas.style.background = "rgba(r, g, b, a)";
context = canvas.getContext("2d");
document.getElementById("div").appendChild(canvas);
```

### 3. wskaźnik szybkości

wygeneruj wskaźnik szybkości z linii i tekstu na Canvasie

wskaźnik rysuje się na bieżąco w publicznej funkcji **update()** pliku **KokpitScreen.js**

- maska - obszar poza nią nie jest rysowany!

```
context.rect(x, y, szer, wys);
context.clip();
```

- stałe odświeżanie określonego obszaru, wywołanie wrażenia przesuwania jakiegoś kształtu

```
context.clearRect(x, y, szer, wys);
```

- narysowanie linii

```
context.beginPath();
context.lineWidth = grubosc;
context.strokeStyle = "rgba(r, g, b, a)";
context.moveTo(x, y); // początek linii
context.lineTo(x, y); // koniec linii
context.stroke();
context.closePath()
```

- napisy

```
context.font = "40px arial";
```

```
context.fillStyle = "rgba(r, g, b, a)";  
context.fillText("napis", x, y);
```

- wypełniony prostokąt

```
context.beginPath();  
context.lineWidth = 5;  
context.fillStyle = "rgba(r, g, b, a)";  
context.fillRect(x, y, szer, wys);
```

- niewypełniony prostokąt

```
context.strokeStyle = "rgba(r, g, b, a)";  
context.strokeRect(x, y, szer, wys);  
context.closePath();
```

- wywołuj funkcję **update()** w **animateScene()** w **Main.js**, podając wartość z Slidera jako jej parametr

#### 4. wskaźnik wysokości

- wykonanie tak samo jak powyżej, w kolejnym Canvasie

#### 5. wskaźnik kierunków świata

- wykonanie tak samo jak powyżej, w osobnym Canvasie
- obrót jakiegoś kształtu dokładnie wokół swojego środka:

```
function obracajTo() {  
  
    context2.clearRect(0, 0, 300, 300); // odświeżanie całego canvasa  
    context2.translate(150, 150) // przesunięcie punktu rysowania  
    context2.rotate(Math.PI / 180) // obrót  
    context2.translate(-150, -150) // przesunięcie z powrotem  
    context2.fillStyle = "rgba(r, g, b, a)";  
    context2.fillRect(100, 100, 100, 100); // narysowanie kwadratu  
  
}
```

obracajTo() wywołuj w **animateScene()**

#### 6. gotowy przykład ilustrujący działanie powyższych metod:

[http://admins.spec.pl/hostingasp.pl/Stefa%C5%84czyk\\_Dariusz/THREEJS/utils/canvasdemo.html](http://admins.spec.pl/hostingasp.pl/Stefa%C5%84czyk_Dariusz/THREEJS/utils/canvasdemo.html)

Punktacja na dziś - 4 pkt

- kolizja z terenem - widoczny w czasie lotu punkt kolizji - proszę o teren o niskiej gęstości
- wskaźnik horyzontu - canvas
- wskaźnik szybkości - canvas
- wskaźnik wysokości - canvas