# Code dans INDEX .html

## Style

<style>

html, body {

overflow: hidden;

width: 100%;

height: 100%;

margin: 0;

padding: 0;

}

#hud { position: absolute; z-index: 1; width: 640px; padding: 5px 0; font-family: Verdana, Geneva, sans-serif; font-size: 0.8em; background-color: rgba(255,0,0,0.4); color: black; border-bottom: 2px solid black; box-sizing: border-box; -moz-box-sizing: border-box; -webkit-box-sizing: border-box; }

#hud .hud { background-color: rgba(255,255,255,0.6); padding: 5px; border: 1px solid black; margin: 0 5px; transition-property: background-color; transition-duration: 2s; -webkit-transition-property: background-color; -webkit-transition-duration: 2s; }

#hud #current\_lap\_time { float: left; }

#hud #last\_lap\_time { float: left; }

#hud #fast\_lap\_time { display: block; width: 12em; margin: 0 auto; text-align: center; transition-property: background-color; transition-duration: 2s; -webkit-transition-property: background-color; -webkit-transition-duration: 2s; }

#hud .value { color: black; font-weight: bold; }

#hud .fastest { background-color: rgba(255,215,0,0.5); }

#hide { display: none; }

#renderCanvas {

width: 100%;

height: 100%;

touch-action: none;

}

</style>

## Divs

<div id="hud">

<span id="current\_lap\_time" class="hud">Time: <span id="current\_lap\_time\_value" class="value"></span></span>

<span id="last\_lap\_time" class="hud hide">Last Lap: <span id="last\_lap\_time\_value" class="value"></span></span>

<span id="fast\_lap\_time" class="hud">Fastest Lap: <span id="fast\_lap\_time\_value" class="value">0.0</span></span>

</div>

# CODE DANS MAIN .js

Attention !!!!!!! vroom est mon cube de test, à changer par la voiture

Variable en haut :

var lap = 0;

var totalLap = 3;

var pastCP = false;

var finished = false;

var currentLapTime = 0;

var lastLapTime = 0;

var bestLapTime = 0;

var newTime = 0;

var elapsed = 0;

var lapTime = 0;

## fonction dans le main (ne se lance qu’une fois)

//Set the elapsed time

Timer = function () {

newTime = Date.now();

elapsed = (newTime - startTime);

};

function TimeToString(time){

var minutes = Math.floor((time/1000) / 60); //minutes

var seconds = Math.floor((time/1000) - (minutes\*60)); //secondes

return minutes.toString() + "." + seconds.toString();

}

function RenderTime(id, pTime){

document.getElementById(id).innerHTML = TimeToString(pTime);

}

## Dans la RenderLoop (enfin à faire loop quoi)

## Collisions et timers

Timer();

lapTime = elapsed - lastLapTime;

RenderTime("current\_lap\_time\_value", lapTime);

if (vroom.intersectsMesh(finishLine, false) && pastCP)

{

if(lapTime < bestLapTime || bestLapTime == 0){

bestLapTime = lapTime;

RenderTime("fast\_lap\_time\_value", bestLapTime);

}

RenderTime("last\_lap\_time\_value", lapTime);

lastLapTime = elapsed;

lapTime = 0;

lap+=1;

pastCP = false;

if(lap == totalLap)

{

finished = true;

console.log("GAME FINISHED");

}

}

if (vroom.intersectsMesh(checkPoint, false))

{

pastCP = true;

}

# Code dans MainScene.js

ATTENTION !!!! vroom à supprimer et à remplacer par la voiture

(et si vous avez la solution, changer « finishLine » et « checkPoint » en variable non globale

var matBB = new BABYLON.StandardMaterial("matBB", this.scene);

matBB.emissiveColor = new BABYLON.Color3(1, 1, 1);

matBB.wireframe = true;

//finish line

finishLine = new BABYLON.Mesh.CreateBox("finishLine", 2, this.scene);

var textureFinish = new BABYLON.Texture("assets/damier.png", this.scene);

var materialFinish = new BABYLON.StandardMaterial("fl", this.scene);

materialFinish.diffuseTexture = textureFinish;

finishLine.material = materialFinish;

finishLine.position = new BABYLON.Vector3(15, 5,-30);

finishLine.scaling = new BABYLON.Vector3(3, 0.05, 11);

finishLine.computeWorldMatrix(true);

//finishLine Collider

var finishCollider = BABYLON.Mesh.CreateBox("finishCollider", 20, this.scene);

finishCollider.material = matBB;

finishCollider.position = new BABYLON.Vector3(15, 5,-30);

finishCollider.scaling = new BABYLON.Vector3(0.4, 0.05, 1);

//checkpoint

checkPoint = new BABYLON.Mesh.CreateBox("finishLine", 2, this.scene);

checkPoint.id = "checkPoint";

checkPoint.position = new BABYLON.Vector3(15, 5, 30);

checkPoint.scaling = new BABYLON.Vector3(0.3, 0.1, 11);

var textureCP = new BABYLON.Texture("assets/cp.jpg", this.scene);

var materialCP = new BABYLON.StandardMaterial("cp", this.scene);

materialCP.diffuseTexture = textureCP;

checkPoint.material = materialCP;

checkPoint.computeWorldMatrix(true);

//checkPoint Collider

var checkPointCollider = BABYLON.Mesh.CreateBox("finishCollider", 20, this.scene);

checkPointCollider.material = matBB;

checkPointCollider.position = new BABYLON.Vector3(15, 5, 30);

checkPointCollider.scaling = new BABYLON.Vector3(0.4, 0.05, 1);

vroom = new BABYLON.Mesh.CreateBox("vroom", 2, this.scene);

vroom.id = "vroom";

vroom.position = new BABYLON.Vector3(30, 7,-30);

vroom.scaling = new BABYLON.Vector3(2, 2, 2);

var camera = new BABYLON.FollowCamera("FollowCam", new BABYLON.Vector3(0, 15, -45), this.scene);

camera.target = vroom;

this.scene.activeCamera = camera;