Vytvořte si vlastní webovou hru

Pavol Hejný

Web vs. Desktop

Výhody

- Přenositelnost (URL)

Problémy

- Výkon
- Místo v uložišti prohlížeče
- "Seamfull"

Další možnosti:

- Unity, (Flash, Java)

Browser APIs

- WebGL
- Pointer Lock API
- Fullscreen API
- WebWorkers
- WebSockets
- Messages
- Polyfill Hand.js

Canvas

```
<canvas id="scene"></canvas>
```

```
const sceneCanvas = window.document.getElementById('scene');
const ctx = canvas.getContext("2d");

ctx.fillStyle = '#ff0000';
ctx.beginPath();
ctx.arc(10,10, 5, 0, Math.PI * 2, true);
ctx.closePath();
ctx.fill();
```

2D vs. 3D

- **2D CanvasRenderingContext2D:** Matter.js
- **3D WebGl:** Babylon.js, Three.js, A-Frame

Další možnosti:

- 2D pomocí WebGL
- isometrie

UI (Canvas vs. html)

- React & MobX

```
export default class UIDataModel {
    @observable health = 100;
    @observable energy = 50;
}
```

```
export default observer(({uiDataModel})=>{
    return (
        <div className="counters">
            <div className="health">
                {uiDataModel.health.toString()}
            </div>
            <div className="energy">
                {uiDataModel.energy.toString()}
            </div>
        </div>
    );
});
```

Fyzika

- Pouze gravitace vs. vše se vším
- Cannon.js, Oimo.js
- Binding např. pro Babylon.js

Realtime (vs. tahové)

- drawLoop
- requestAnimationFrame(cb) vs. setInterval(cb,ms)
- performance.now()

```
function drawLoop(){
        //update(...);
        //render(...);
        requestAnimationFrame(drawLoop);
}

requestAnimationFrame(drawLoop);
```

Multiplayer

- **WebSockets**
- Pouze ukládání výsledků na server

Babylon.js + Oimo.js

https://www.babylonjs-playground.com/#DCR6ZG#5

https://github.com/hejny/3d-project

https://drive.google.com/file/d/1omHIR3KcjLjHgye7tYdhHKPQZ4IawVCl/view

Scéna

```
const scene = new BABYLON.Scene(engine);
const camera = new BABYLON.FreeCamera(
  'camera1',
  new BABYLON.Vector3(0, 5, -10),
  scene
);
camera.attachControl(canvas, true);
camera.setTarget(BABYLON.Vector3.Zero());
const light = new BABYLON.HemisphericLight(
  'light1',
  new BABYLON.Vector3(0, 1, 0),
  scene
);
scene.enablePhysics(
  BABYLON.Vector3.Zero(),
  new BABYLON.OimoJSPlugin()
);
```

"Planeta"

```
for(let z=-2;z<=2;z++){
  for(let y=0;y<5;y++){
    for(let x=-2;x<=2;x++){
      const boxMesh = BABYLON.Mesh.CreateBox(
        'box',
        0.2,
        scene
      );
      boxMesh.position = new BABYLON.Vector3(
        x*.2,
        (y+.5)*.2,
        z*.2
      );
      boxMesh.physicsImpostor = new BABYLON.PhysicsImpostor(
        boxMesh,
        BABYLON.PhysicsImpostor.BoxImpostor,
        { mass: 1, restitution: 0.3 },
        scene
      );
    }
  }
}
```

Asteroid

```
const asteroidMesh = BABYLON.Mesh.CreateSphere(
  'asteroid',
  16,
  0.33,
  scene
);
asteroidMesh.position = new BABYLON.Vector3(0,-10,0);
asteroidMesh.physicsImpostor = new BABYLON.PhysicsImpostor(
  asteroidMesh,
  BABYLON.PhysicsImpostor.SphereImpostor,
  { mass: 100, restitution: 0.3 },
  scene
);
asteroidMesh.physicsImpostor.setLinearVelocity(
  asteroidMesh.position.scale(-.2)
);
```

Materiály

Ambient

Okolní konstantní osvětlení, co osvětluje předmět rovnoměrně bez ohledu na směr osvětlení.

Diffuse

Světlo rozptýlené do všech stran. Díky této složce je na předmětu vytvořený "3D efekt".

Specular

Světlo odrážející se převážně v jednom směru, co tvoří odlesk.

Emissive

Světlo vyzařované z meshe, hodí se např. pro vytvoření monitoru nebo kina ve scéně.

```
const material = new BABYLON.StandardMaterial(
    'stone-plain',
    scene
);
const texture = new BABYLON.Texture(
    process.env.PUBLIC_URL +`/assets/textures/stone-plain.jpg`,
    scene
);
texture.uScale = 1;
texture.vScale = 1;
material.diffuseTexture = texture;
material.specularColor = BABYLON.Color3.FromHexString('#ffeacb');
material.emissiveTexture = texture;
```

```
mesh.material = material;
```

```
mesh.material = materialFactory.getMaterial('stone-plain');
```

Ground

```
const groundMesh = BABYLON.Mesh.CreateGround(
  "ground",
  1000, 1000,
  2,
  scene
);
groundMesh.material = materialFactory.getMaterial('grass');
groundMesh.physicsImpostor = new BABYLON.PhysicsImpostor(
  groundMesh,
  BABYLON.PhysicsImpostor.BoxImpostor,
  { mass: 0, restitution: 0.1},
  scene
);
```

Brick

```
export default class Brick{
    public mesh:BABYLON.AbstractMesh;

    constructor(
        private _world:World,
        private _materialName:string,
        private _physicalProperties:{mass:number,restitution:number},
        private _size:BABYLON.Vector3,
        private _position:BABYLON.Vector3,
        private _rotation:BABYLON.Vector3 = BABYLON.Vector3.Zero(),
        private _linearVelocity:BABYLON.Vector3 = BABYLON.Vector3.Zero(),
        private _angularVelocity:BABYLON.Vector3 = BABYLON.Vector3.Zero(),

    )
```

```
const globalScale = 10;
const width = this._size.x;
const height = this._size.y;
const depth = this._size.z;
const faceUV = [
    new BABYLON.Vector4(0, 0, width / globalScale, height / globalScale),
    new BABYLON.Vector4(0, 0, width / globalScale, height / globalScale),

    new BABYLON.Vector4(0, 0, height / globalScale, depth / globalScale),
    new BABYLON.Vector4(0, 0, height / globalScale, depth / globalScale),

    new BABYLON.Vector4(0, 0, depth / globalScale, width / globalScale),
    new BABYLON.Vector4(0, 0, depth / globalScale, width / globalScale),
];
const meshOptions = {width, height, depth, faceUV};
this.mesh = BABYLON.MeshBuilder.CreateBox('BoxBrick', meshOptions, this._world.s

this.mesh.material = this._world.materialFactory.getMaterial(this._materialName)
this.mesh.physicsImpostor = new BABYLON.PhysicsImpostor(
    this.mesh,
    BABYLON.PhysicsImpostor.BoxImpostor,
    this._physicalProperties,
    this._world.scene
);
```

Děkuji za pozornost

- www.pavolhejny.com
- www.webappgames.com
- https://hejny.github.io/3d-project/
- https://github.com/hejny/3d-project
- https://www.babylonjs-playground.com/#DCR6ZG#5
- https://www.itnetwork.cz/vytvor-si-vlastni-webovou-hru/