Verkefni 3

Sturla Freyr Magnússon : **sfm1@hi.is**
sturla-freyr.github.io/

TÖL105M-Haustönn 2024

HÁSKÓLI ÍSLANDS

Three.js Frogger

Source code

1 Ferlið

Ég byrjaði á því að setja upp þróunarumhverfið með npm og vite sem er nútímalegra en við höfum fengið að sjá en mig langaði að prófa nýjasta nýtt. Fyrsta skrefið var að setja upp einfaldan three.js grunn:

```
const scene = new THREE.Scene();
const camera = new THREE.PerspectiveCamera(75,
    window.innerWidth / window.innerHeight, 0.1, 1000);
const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);
```

Næst fékk ég hlutinn sem var í leikmannsstað til að hreyfast og bætti við grunnlýsingu með `AmbientLight` og `DirectionalLight` ásamt myndavél sem fylgir leikmanninum. Ég ákvað að nota `PerspectiveCamera` með sjónarhorn sem gefur góða yfirsýn yfir leiksvæðið. Myndavélin er stillt þannig að hún horfir niður á leikmanninn með ákveðnu offset:

```
setupCamera() {
    this.camera.position.set(0, 10, 45);
    this.camera.lookAt(this.playerSystem.mesh.position);
}
```

Eftir að hafa fengið grunnsenuna til að virka einbeitti ég mér að því að bæta við einföldum hlutum eins og gólfi og hindrunum. Fyrst notaði ég bara einföld form eins og kassa og sívalinga með `BoxGeometry` og `CylinderGeometry`. Þegar það var komið fór ég að bæta við einfaldri eðlisfræði og árekstrarprófunum. Ég bjó til `PHYSICS_CONFIG` hlut til að halda utan um eðlisfræðibreytur:

```
const PHYSICS_CONFIG = {
    gravity: 0.002,
    platformGravity: 0.001,
    groundLevel: -0.5,
    jumpCooldown: 500,
    platformFriction: 0.8,
    groundFriction: 0.95,
    stickyForce: 0.5
}
```

Á þessum tímapunkti var ég kominn með ansi mikið af spaghetti kóða og ákvað að það væri best að taka smá pásu og skipuleggja verkefnið almennilega. Ég eyddi góðum tíma í að skipta öllu upp í kerfi:

- `PlayerSystem`: Sér um hreyfingar og stöðu leikmanna
- `CameraSystem`: Stjórnar myndavélinni og fylgir leikmanni
- `LightingSystem`: Sér um lýsingu og skugga
- `MovementSystem`: Stjórnar hreyfingum hluta eins og bíla og trjádrumba
- `SceneManager`: Heldur utan um alla hluti í senunni og árekstra

Þegar grunnkerfin voru komin fór ég að bæta við textúrum og flóknari módelum. Ég notaði `GLTFLoader` fyrir bílana og `PLYLoader` fyrir froskinn:

```
const loader = new PLYLoader();
loader.load(
    'textures/frog.ply',
    (geometry) => {
        geometry.scale(0.33, 0.33, 0.33);
        geometry.computeVertexNormals();
        const material = new THREE.MeshPhongMaterial({
            color: 0x55ff55,
            shininess: 30,
            specular: 0x444444
        });
        const plyMesh = new THREE.Mesh(geometry, material);
        // ... uppsetning á mesh
    }
);
```

Einn flóknasti hluti verkefnisins var að bæta við billboarding fyrir tré og gras. Þetta krafðist þess að ég skrifaði sérstakan shader:

```
const vegetationMaterial = new THREE.ShaderMaterial({
    uniforms: {
        map: { value: texture },
        threshold: { value: threshold },
        tint: { value: tintColor }
    },
    vertexShader: `
        varying vec2 vUv;
        void main() {
            vUv = uv;
            gl_Position = projectionMatrix * modelViewMatrix * vec4(position, 1.0);
        }
    `,
```

```
    fragmentShader: `
        uniform sampler2D map;
        uniform float threshold;
        uniform vec3 tint;
        varying vec2 vUv;
        void main() {
            vec4 texColor = texture2D(map, vUv);
            float brightness = (texColor.r + texColor.g + texColor.b) / 3.0;
            vec3 finalColor = mix(texColor.rgb, texColor.rgb * tint, 0.5);
            gl_FragColor = vec4(finalColor, step(threshold, brightness));
        }
    `,
    transparent: true,
    side: THREE.DoubleSide
});
```

Eins og sjá má ef litið er á kóðann er verkefnið orðið nokkuð stórt og flókið með mörgum kerfum sem vinna saman. Kóðinn er nokkuð vel skipulagður þó ég segi sjálfur frá og því tiltölulega auðvelt að bæta við nýjum eiginleikum. Stærstu áskoranirnar voru að innleiða collision við kúpt yfirborð og að innleiða billboarding.

2 Tæknilegar útfærslur

Þó að leikurinn sjálfur sé ekki mjög skemmtilegur þá er verkefnið gott dæmi um hvernig hægt er að nota helstu aðferðir tölvugrafíkur í raunverulegri útfærslu:

- **Scene Graph**: Heildarhönnun á senunni með skipulögðu object hierarchy
- **Collision Detection**: Árekstrarkerfi sem tekst á við mismunandi rúmfræðileg form
- **Lighting & Shadows**: DirectionalLight með PCFSoftShadowMap fyrir raunsæja birtudreifingu
- **Dynamic Texturing**: Mismunandi aðferðir við texture mapping fyrir ólík yfirborð
- **Custom Shaders**: GLSL shaders fyrir sérhæfða grafíska eiginleika
- **3D Asset Pipeline**: Vinnsla með bæði GLTF og PLY snið fyrir mismunandi tilgang

3 Framtíðarplön

Ef ég hefði meiri tíma myndi ég vilja:

- Bæta við sigurstöðu þegar froskur nær útí enda (núna er markmiðið bara að deyja ekki sem er kannski aðeins of nihilískt)
- Bæta árekstursprófanir fyrir trjádrumba og sérstaklega fyrir skjaldbökur
- Bæta eðlisfræði leiksins
- Bæta afköst með því að:
  - Nota instance mesh fyrir gras og tré
  - Setja upp frustum culling fyrir billboards
  - Nota LOD (Level of Detail) fyrir fjarlæga hluti
- Bæta við hljóðum og particle effects