Dokumentace klauzurní práce

Cartomantica

[Odkaz]

STŘEDNÍ ŠKOLA FILMOVÁ, MULTIMEDIÁLNÍ A POČÍTAČOVÝCH TECHNOLOGIÍ, s.r.o.

Vypracoval: Hanuš Valenta 2.C 2024

1. Úvod

V dnešní době je považováno za nezbytné, aby každý, kdo se zabývá tvorbou map, měl k dispozici kvalitní nástroj pro jejich vytváření. Pro tvůrce map je klíčové, aby jejich program umožňoval ukázat dovednosti v oblasti designu, práce s různými vrstvami a technologiemi a schopnost řešit složité výzvy při plánování map. Díky těmto funkcím mohou uživatelé snadno vytvářet kvalitní výstupy.

2. Co je v dokumentu uvedeno

V dokumentu jsou uvedeny informace o využitých technologiích a jejich následném praktickém využití při vývoji Cartimantiky. Podrobně je popsáno, jakým způsobem byly tyto technologie aplikovány v různých fázích vývoje a jakým přínosem byly pro celkovou funkčnost a design. Zároveň je věnována pozornost popisu jednotlivých adresářů a souborů, které byly vytvořeny v rámci projektu. Každý adresář a soubor je detailně popsán, včetně jejich specifických funkcí a způsobu, jakým přispívají k celkové struktuře a organizaci projektu. Díky těmto informacím je možné lépe porozumět nejen technickým aspektům, ale také jeho praktickému použití a údržbě.

3. Popis použitých technologií

3.1 HTML

HTML je zkratka pro Hypertext Markup Language a je základním jazykem pro tvorbu webových stránek. Používá se k definování struktury a obsahu stránek pomocí různých značek a elementů. [1]

3.2 CSS

CSS, zkratka z anglického názvu Cascading Style Sheets, je jazyk používaný pro stylování webových stránek. Jeho hlavním účelem je definovat vzhled a prezentaci obsahu HTML dokumentů. CSS umožňuje nastavovat různé vlastnosti elementů, jako je barva, velikost, poloha, písmo a mnoho dalšího. [2]

3.3 JavaScript

JavaScript je primárně známý jako skriptovací jazyk pro webové stránky, kde je používán k interaktivitě, dynamickému obsahu a manipulaci s HTML a CSS. S jeho pomocí je možné reagovat na uživatelské interakce, validovat formuláře, vytvářet animace, dynamicky měnit obsah stránek a mnoho dalšího. [3]

```
function createWindow() {
    const win = new BrowserWindow({
        width: 800,
        height: 600,
        icon: path.join(__dirname, 'icon.png'),
        webPreferences: {
            nodeIntegration: false,
            contextIsolation: true,
            preload: path.join(__dirname, 'preload.js')
        }
    });

    win.loadFile('./web/src/index.html');
    win.maximize();

    win.setMenuBarVisibility(false);
}
```

[1]

[1] Ukázka javascript kódu z index.js na vytváření okna.

3.4 NodeJS

Node.js je runtime prostředí, které je využíváno pro vývoj serverových aplikací. Je postaveno na JavaScriptovém engine, který umožňuje běh JavaScriptu mimo webový prohlížeč. Umožňuje práci s asynchronními operacemi díky událostmi řízené architektuře, čímž je dosaženo vysoké škálovatelnosti a efektivity při zpracování velkého množství současných požadavků. Node.js poskytuje různé moduly pro práci s http. [4]

3.5 Three.js

Three.js je open-source JavaScriptová knihovna, která umožňuje tvorbu a zobrazování 3D grafiky přímo ve webovém prohlížeči. Je postavena na WebGL, což je standardní webová API pro práci s 3D grafikou v prohlížeči, a umožňuje vývojářům vytvářet působivé 3D scény a interaktivní vizualizace bez nutnosti použití pluginů či externích programů. [5]

```
import * as THREE from '../../node_modules/three/build/three.module.js';
import { GLTFLoader } from '../../node_modules/three/examples/jsm/loaders/GLTFLoader.js';
import { TransformControls } from 'three/examples/jsm/controls/TransformControls.js';

import { populateObjectList } from './modules/populateObjectList.js';
import { createHandDrawnOutline } from './modules/createHandDrawnOutline.js';

import { createPathMaterial, createPathGeometry } from './shaders/pathShader.js';
import { createWaterMaterial, createWaterGeometry } from './shaders/waterShader.js';

const scene = new THREE.Scene();
const loader = new GLTFLoader();
scene.background = new THREE.Color(0xffffff);

const aspect = window.innerWidth / window.innerHeight;
const camera = new THREE.OrthographicCamera(-10 * aspect, 10 * aspect, 10, -10, 0.1, 1000);
camera.position.set(0, 50, 0);
camera.lookAt(0, 0, 0);
```

[2]

3.6 Electron.js

Electron.js je open-source framework, který umožňuje vývoj desktopových aplikací pomocí webových technologií, jako jsou HTML, CSS a JavaScript. Je postaven na kombinaci Node.js a Chromium, což umožňuje vývojářům vytvářet multiplatformní aplikace, které běží na Windows, macOS i Linuxu, aniž by museli psát kód pro každou platformu zvlášť. Díky této technologii mohou vývojáři využívat své znalosti webového vývoje k tvorbě plnohodnotných desktopových aplikací, které kombinují výkonnost nativních aplikací a flexibilitu moderního webu. [6]

[2] Ukázka vytváření node.js scény.

4. Použitý editor

4.1 VSCodium

VSCodium je open-source distribuce Visual Studio Code, která je poskytována bez integrovaných služeb Microsoftu, jako je například Telemetry. To znamená, že VSCodium je úplně stejný jako Visual Studio Code, ale neobsahuje žádný kód, který by sledoval nebo sbíral informace o uživatelském chování. [7]

4.2 Výhody VSCodium

VSCodium nabízí širokou škálu funkcí pro vývojáře, včetně podpory pro mnoho programovacích jazyků, integrovaných nástrojů pro ladění a správu verzí a rozsáhlého ekosystému rozšíření. VSCodium je ideální volbou pro ty, kteří hledají výkonný a flexibilní textový editor, který respektuje jejich soukromí a bezpečnost.

5. Popis adresářů

5.1 web\src

Zde je celý projekt a jeho soubory, spolu s nezbytnými knihovnami.

5.2 Cartomantica

V několika adresářích jsou umístěny veškeré potřebné grafiky, ikony, fonty a modely, které jsou nezbytné pro funkci applikace.

5.3 node_modules

Všechny knihovny používané pro Node.js spolu s jejich skripty jsou umístěny v této složce.

5.4 models

Zde najdete kolekci modelů ve formátu gltf které jsou z spawn menu importovat hned po otevření applikace.

5.5 modules

Zde se nacházejí některé funkce dost velké pro jejich exportování do vlastních modulů.

5.6 shaders

Zde se nachází soubory pro tvorbu shaderů pro path tool projektu.

6. Popis souborů

6.1 index.html

Tento soubor je hlavní stránka applikace která se načte po otevření a následně je stylizovaná a zfunkčnena pomocá css a javascriptu.

6.2 styles.css

Zde se nachází stylizace pro celý projekt pomocí stylizovacího jazyka css.

6.3 index.js

Vytvoření okna a načtení indexu hned po otevření programu.

6.4 main.js

Hlavní soubor funcí obsahující scénu a volající hlavní animate funkci.

6.5 objects.json

List objektů tvořených pomocí jednoduchých tvarů, také s listem gltf objektů které jsou dostupné s applikací.

6.6 preload.js

Soubor pro Electronn zobrazuje verze Chrome, Node.js a Electronu v HTML a umožňuje bezpečně zavřít aplikaci pomocí tlačítka nebo jiné akce v uživatelském rozhraní.

```
window.addEventListener('DOMContentLoaded', () => {
    const replaceText = (selector, text) => {
      const element = document.getElementById(selector)
      if (element) element.innerText = text
    }

    for (const dependency of ['chrome', 'node', 'electron']) {
      replaceText(`${dependency}-version`, process.versions[dependency])
    }
  })

const { contextBridge, ipcRenderer } = require('electron');

window.electronAPI = { ipcRenderer, contextBridge };

contextBridge.exposeInMainWorld('electronAPI', {
    closeApp: () => ipcRenderer.invoke('close-app'),
});
```

3

[3] Kód ukazuje verze v HTML a umožňuje zavřít aplikaci.

6.7 createHandDrawnOutline.js

Tento kód vytváří obrys kolem objektu v Three.js pomocí jeho hran. Obrys má nastavenou barvu, tloušťku a mírně větší velikost než původní objekt. Výsledek se přidá jako součást objektu.

6.8 populateObjectList.js

Kód načte data z JSON souboru a přidá je jako možnosti do HTML <select> prvku.
Při chybě zobrazí hlášku v konzoli.

```
import * as THREE from 'three';

export function createHandDrawnOutline(mesh, color = 0x000000, thickness = 1) {
    const edgesGeometry = new THREE.EdgesGeometry(mesh.geometry);
    const outlineMaterial = new THREE.LineBasicMaterial({
        color: color,
        linewidth: thickness,
    });

    const outline = new THREE.LineSegments(edgesGeometry, outlineMaterial);

    outline.scale.set(1.01, 1.01, 1.01);
    outline.renderOrder = 1;
    outline.raycast = () => {};

    mesh.add(outline);

    return outline;
}
```

[4]

```
export async function populateObjectList() {
    try {
        const response = await fetch('objects.json');
        const objects = await response.json();

        const objectSelect = document.getElementById('objectSelect');
        objectSelect.innerHTML = '';

        objects.forEach((object) => {
            const option = document.createElement('option');
            option.value = object.type;
            option.textContent = `${object.type.charAt(0).toUpperCase() +
  object.type.slice(1)}`;
            objectSelect.appendChild(option);
        });
    } catch (error) {
        console.error('Failed to populate object list:', error);
    }
}
```

[5]

[4] Kód vytvoří obrys kolem objektu v Three.js.
[5] Příklad načte data z JSON a přidá je jako možnosti do HTML <select> prvku.

```
function createWindow() {
    const win = new BrowserWindow({
        width: 800,
        height: 600,
        icon: path.join(__dirname, 'icon.png'),
        webPreferences: {
            nodeIntegration: false,
            contextIsolation: true,
            preload: path.join(__dirname, 'preload.js')
        }
    });

    win.loadFile('./web/src/index.html');
    win.maximize();

    win.setMenuBarVisibility(false);
}
```

[6]

```
{
        "type": "Maxwell",
        "geometry": "GLTF",
        "modelPath": "./models/maxwell.glb",
        "scale": [
            0.1,
            0.1,
            0.1
        ]
},
```

[7]

[6] kód vytváří nové okno v aplikaci Electron s nastavenými rozměry, ikonou a bezpečnostními preferencemi. Načte HTML soubor a otevře okno na maximální velikost bez viditelného menu.

[7] Ukázka z objects.json obsahuje údaje o 3D modelu s typem "Maxwell", formátu geometrie GLTF a cestě k souboru modelu. Model je zobrazen ve zmenšené velikosti díky měřítku 0.1 na každé ose.

6.9 package.json a package-lock.json

Oba soubory, package.json a package-lock.json, jsou důležité pro správu závislostí Node.js aplikace. package.json obsahuje metadata a seznam knihoven aplikace, zatímco package-lock.json udržováním přesného stavu těchto knihoven pomáhá zajistit konzistenci mezi různými instalacemi aplikace. [8]

6.10 .gitignore

Soubor .gitignore určuje, které soubory a složky Git nebude sledovat nebo přidávat do verzovacího systému. Používá se k ignorování dočasných, generovaných nebo citlivých souborů.[9]

6.11 waterShader.js

Vytváří materiál a geometrii pro simulaci vody v Three.js. Materiál používá vlastní shadery k vytvoření vlnového efektu a noise textury pro realistický vzhled vody. Geometrie je založena na křivce, která se tvaruje do 3D objektu pro zobrazení vody.

```
vertexShader: `
            varying vec2 vUv;
            uniform float time;

            void main() {
                vUv = uv;
                vec3 transformed = position;

                // Wave-like undulation effect
                float waveFrequency = 5.0;
                float waveAmplitude = 0.05;
                transformed.y += sin(position.x * waveFrequency + time) * waveAmplitude;
                transformed.y += cos(position.z * waveFrequency + time) * waveAmplitude;

                gl_Position = projectionMatrix * modelViewMatrix * vec4(transformed, 1.0);
            }
        `
```

[8]

[8] Tento vertex shader vytváří vlnění na objektu pomocí sinusových a kosinusových funkcí, které upravují pozice bodů podle času a jejich souřadnic. Výsledkem je efekt pohybujících se vln na vodní ploše. Transformace je provedena pomocí projekčního a modelového pohledu.

6.12 pathShader.js

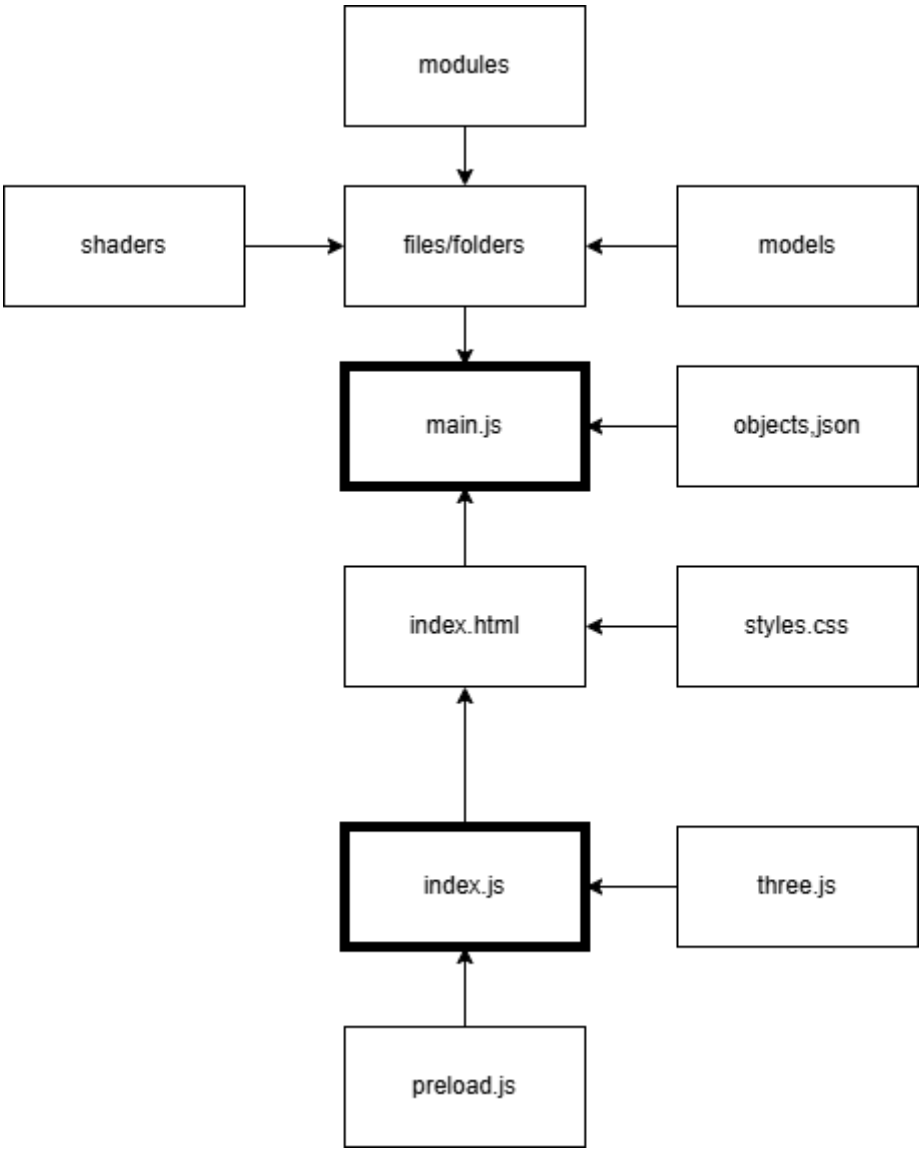
vytváří materiál a geometrii pro zobrazení cesty v Three.js. Materiál používá shadery k vytvoření efektu vlnění a organického vzhledu textury s náhodným šumem, který mění barvu a průhlednost cesty. Geometrie cesty je definována pomocí 3D křivky, která je tvarována do cestovitého objektu.

```
function animate() {
    requestAnimationFrame(animate);

    updateCameraPosition();
    const deltaTime = 0.05;

    updateObjectPosition();

    updateSunPosition(deltaTime);

    if (isEditMode && selectedObject) {
        rotationSpeed *= (1 - rotationFriction);
        selectedObject.rotation.y += rotationSpeed;

        scalingVelocity *= (1 - scalingFriction);
        const newScale = selectedObject.scale.x + scalingVelocity;

        if (newScale > 0.1) {
            selectedObject.scale.set(newScale, newScale, newScale);
        }
    }

    renderer.render(scene, camera);
}

animate();
```

[9]

[9] Hlavní animační smyčka v main.js pro vykreslování scény v aplikaci. V každém cyklu aktualizuje pozici kamery, objektů a slunce, a pokud je aktivní režim úprav, také provádí rotaci a změnu velikosti vybraného objektu s postupným zpomalením. Nakonec je scéna vykreslena s novými hodnotami.

7. Schéma applikace



8. Zdroje

[1] WIKIEPDIA. HTML. Online. 2003, 29. května 2024. Dostupné z: https://en.wikipedia.org/wiki/HTML. [cit. 2024-12-08].

[2] WIKIEPDIA. CSS. Online. 2001, 6. května 2024. Dostupné z: https://en.wikipedia.org/wiki/CSS. [cit. 2024-12-08].

[3] WIKIEPDIA. JavaScript. Online. 2001, 18. dubna 2024. Dostupné z: https://en.wikipedia.org/wiki/JavaScript. [cit. 2024-12-08].

[4] WIKIEPDIA. Node.js. Online. 2009, 2. června 2024. Dostupné z: https://en.wikipedia.org/wiki/Node.js. [cit. 2024-12-08].

[5] WIKIEPDIA. Three.js. Online. 2012, 11. ledna 2024. Dostupné z: https://en.wikipedia.org/wiki/Three.js. [cit. 2024-12-08].

[6] OPENJS FOUNDATION AND ELECTRON CONTRIBUTORS. Electron.js. Online. 2023. Dostupné z: https://www.electronjs.org/. [cit. 2024-12-08].

[7] TIM O'BRIEN. VSCodium. Online. 2019, 8. ledna 2024. Dostupné z: https://vscodium.com/#why. [cit. 2024-12-08].

[8] EDWARD THOMSON. Package.json. Online. 2019, 22. září 2020. Dostupné z: https://docs.npmjs.com/cli/v6/configuring-npm/package-json. [cit. 2024-12-08].

[9] SOFTWARE FREEDOM CONSERVANCY, INC. Git-scm. Online. 2024. Dostupné z: https://git-scm.com/docs/gitignore. [cit. 2024-12-08].