

# THE SECRET MAZE

**Nika Demšar (63210382), Urška Frelj Uhelj (63230067), Anja Klančar (63220142)**

*Univerza v Ljubljani, Fakulteta za računalništvo in informatiko*

*E-pošta: nd3657@student.uni-lj.si, uf88958@student.uni-lj.si, ak01265@student.uni-lj.si*

## Opis

*Naša igra "The Secret maze" je igra, kjer se igralec poda skozi labirint, po katerem išče in pobira zlate kovančke. Ustvarjena je s pomočjo programskih jezikov JavaScript in HTML in izkorišča moč tehnologije WebGPU. Razvoj je podprt z uporabo knjižnice WebGPU Examples.*

*Igralec se lahko poljubno giblje po omejenem prostoru na sredini katerega se nahaja kamnit labirint. Okolje je osvetljeno s soncem, ki se nahaja na statičnem nebu nad labirintom. V labirintu je skritih 10 kovancev cilindrične oblike. Igra je narejena v realističnem stilu.*

## 1 Pregled igre

Naša igra spada med miselne in pustolovske. Je zelo preprosta in namenjena vsakemu, ki se želi sprostiti. Ne vsebuje nobenih utripajočih elementov, ki bi lahko uporabnika zmotile. Odvija se v mirnem okolju, ni časovno omejena in igralca ne spravi pod nikakršen pritisk.

### 1.1 Opis sveta

Svet v igri je sestavljen iz statičnih travnatih tal in modrega neba. Opečnat labirint je kvadratne oblike in ima en vhod (začetek) in en izhod (cilj). Izdelan je v realističnem stilu, osebk pa se premikajo v dveh dimenzijah. Nad vsem dogajanjem se nahajata še sonce in oblak.

#### 1.1.1 Pregled

Uporabnik se premika skozi labirint in zbira kovančke, ki so skriti med njegovimi stenami. V stene labirinta se lahko le zaleti, skozi njih pa ne more. Ko se uporabnik premakne na mesto kovančka, ga pobere in kovanček izgine.

#### 1.1.2 Ozadje

Ozadje je sestavljeno iz neba, tal in žive meje. Uporabnik s tlemi interaktira tako, da se po njih premika (hodi), z živo mejo tako, da se vanjo lahko zabije (collision detection). Z nebom in soncem uporabnik ne interaktira.

#### 1.1.3 Ključne lokacije

Ključna lokacija, ki igra pomembno vlogo pri uporabniku je labirint, saj se v njem odvija vse dogajanje. Uporabnik se po njem premika, išče pravo pot in tako skuša poiskati izhod. Labirint ima samo eno pravo pot, ki je označena s kovančki. Če uporabnik zaide, bo najverjetneje prišel

v slepo ulico, iz katere so bo moral vrniti nazaj in najti boljše pot.

V labirintu sta najpomembnejši dve lokaciji - vhod in izhod. Na vhodu se igra prične. Prav tako se uporabnik tja vrne, če kdaj vmes pritisne na gumb "RESTART". Na izhodu pa se igra zaključi, vendar samo tedaj, ko je igralec iz labirinta prinesli vseh 10 zlatih kovancev.

#### 1.1.4 Velikost

Celotna ploskev na kateri ležijo tla in labirint, se razteza na 2224 kvadratnih metrih (36,42 metrov široka in 61,08 metrov dolga).

Svet je zajet izključno na območju travnika z labirintom iz pogleda/višine uporabnika.

#### 1.1.5 Objekti

**Labirint, kovanec, tla, ograja** - objekte smo izdelale same v Blenderju in jim kasneje dodale teksture, ki smo jih prenesle iz spletnih strani blender kit (kamnit labirint) <sup>1</sup> in Poly Heaven (travnata tla) <sup>2</sup> - slike formata .png.

**Nebo** sestavljata sonce in oblak, ki smo ju prenesle iz spleta <sup>3 4</sup> in ju združile v sliko s pomočjo programa Inkscape. <sup>5</sup>

Vsi objekti so bili na koncu združeni v celoto v programu Blender, iz katerega nato naša igra črpa informacije in grafike. Slika 1 in slika 2 prikazujeta pogleda v programu Blender, kjer lahko vidimo vse objekte, uporabljene v igri, združene v celoto.

#### 1.1.6 Čas

V našem svetu se čas premika tako kot v realnosti. Uporabnik se premika s hitrostjo normalne hoje človeka. 5 minut v igri predstavlja 5 minut v realnosti.

## 1.2 Igralni pogon in uporabljene tehnologije

V igri je uporabljena tehnologija WebGPU in <sup>6</sup> knjižnica WebGPU examples, iz katerega je bil vzet primer za collision (05-collision - 01-aabb-aabb). Za osvetlitev modela smo uporabili Lambertovim osvetlitveni model. Za

<sup>1</sup><https://www.blenderkit.com/asset-gallery-detail/74ae7d74-864b-4d55-84b0-a10f342ffb79/>

<sup>2</sup>[https://polyhaven.com/a/rocky\\_terrain\\_02](https://polyhaven.com/a/rocky_terrain_02)

<sup>3</sup><https://www.pngwing.com/en/free-png-dkyaw>

<sup>4</sup><https://www.pngwing.com/en/free-png-bzbni>

<sup>5</sup><https://inkscape.en.softonic.com>

<sup>6</sup><https://github.com/UL-FRI-LGM/webgpu-examples>



Slika 1: Blender - pogled v okolje s soncem.



Slika 2: Blender - pogled v okolje z labirintom

osnovo smo uporabili upodabljajnik *Renderer.js*. Uporabili smo tudi senčilnik pod imenom *Shader.wgsl*.

### 1.3 Pogled

Pogled v naši igri je iz definiran iz perspektive uporabnika. V igri se nahaja kamera *FisrtPersonController*, ki se z uporabnikom premika po prostoru. Uporabnik se lahko s pomočjo miške obrača v vse smeri (lahko pogleda desno, levo, v tla, v sonce) in si s tem prilagaja svoje vidno polje.

## 2 Osebek

Uporabnik ima nadzor nad osebkom - uporabnik je osebek. Osebaka fizično ne vidimo, saj je kamera postavljena tako, da okolje spremljamo iz njegove perspektive. Uporabnik z osebkom upravlja tako, da ga pomika po labirintu. Za to lahko uporablja tipke WASD ali puščice na tipkovnici. S tem se bo osebek premikal naprej, nazaj, desno ali levo, v odvisnosti od njegovega trenutnega pogleda.

## 3 Uporabniški vmesnik

Uporabniški vmesnik je v naši igri zasnovan minimalistično, da uporabnika ne moti pri raziskovanju 3D sveta in labirinta. Razdelimo ga lahko na tri glavne elemente:

- **Začetni zaslon:** vsebuje kratka navodila in gumb **START**, s katerim igralec vstopi v igro. (slika 3)
- **Glavni zaslon igre:** predstavlja sam labirint, po katerem se igralec sprehaja. V zgornjem levem kotu je prikazano, koliko kovancev je že pobral in koliko jih mora še zbrati, da doseže cilj. Tik pod

tem njim je gumb **RESTART**, s katerim lahko igralec kadar koli ponovno začne igro.

- **Končni zaslon:** prikaže se, ko igralec zbere vseh 10 kovancev in doseže izhod iz labirinta. Na zaslonu se izpiše čestitka ob uspehu in prikaže gumb **START AGAIN** za ponovno igranje.



Slika 3: Začetni zaslon, ko zaženemo igrico.



Slika 4: Začetni pogled iz perspektive igralca/uporabnika.



Slika 5: Končni zaslon, ki se pojavi ob zmagi.

## 4 Glasba in zvok

V igri ni glasbe. Dodan je <sup>7</sup>zvočni efekt, ki ga lahko uporabnik sliši, ko pobere kovanec. S tem dosežemo, da je uporabnik siguren, da mu je uspelo pravilno pobrati vsak kovanec posebaj.

<sup>7</sup><https://pixabay.com/sound-effects/coin-recieved-230517/>

## 5 Gameplay

Ko prižgemo igro, se najprej pojavi zaslon, na katerem sta naslov in navodila za igro. Ko si uporabnik prebere navodila, ga pod njimi čaka gumb **START** s katerim lahko začne igro. Takoj je postavljen v svet, točno pred vhod v labirint (slika 3), kjer s nahaja tudi prvi zlati kovanček. Uporabnik s pomočjo tipk na tipkovnici prosto potuje po labirintu in išče pot ven. Ker se kovanci nahajajo samo po pravilni poti, je vsak dodaten pobran kovanec na poti za uporabnika indikator, da potuje po pravi poti. Igro lahko uporabnik zaključi le, če je našel pot iz labirinta in pri tem pobral vseh 10 zlatih kovancev.

V levem zgornjem kotu je ves čas igre tudi števec kovancev, ki služi uporabniku kot informacija, koliko je še oddaljen od cilja in uspešnega zaključka igre.

Kadarkoli v igri se lahko uporabnik odloči, da bo poskusil znova tako, da pritisne na gumb **”RESTART”**, ki se nahaja v zgornjem levem kotu pri števcu kovancev.

Ko uporabniku uspe pobrati vseh 10 kovancev in priti na cilj, se pojavi novo okno, kjer ti igra čestita ob zmagi in ti ponudi, da se vrneš na začetek in igro ponoviš. Slika 4 prikazuje končni zaslon, ki se pojavi ob zmagi.

## 6 Zaključki in možne nadgradnje

Izdelavo igre smo začele popolnoma brez predznanja (z izjemo tistega, ki smo ga pridobile na vajah), saj se nobena od nas še ni nikoli srečala z izdelavo 3D iger in uporabo WebGPU tehnologije. Na začetku smo imele nekaj težav in nekajkrat spremenile idejo.

Naučile smo se uporabe programa blender, programskih jezikov JavaScript in HTML. Naučile smo se postaviti objekt v prostor, ga osvetliti in mu omogočiti, da se premika in ima interakcije z okoljem. Začetna ideja je predvidevala izvedbo gre, kjer igralec teče po svetu v ravni črti, se izogiba oviram in poskuša priti čim dlje. Nato se je s časoma ideja spremenila, naredile smo prvi labirint in nadaljevale z drugo idejo.

Menimo, da smo pri predmetu pridobile kar nekaj koristnega znanja, posebej pa nam je bila v pomoč knjižnica primerov WebGPU Examples, brez katere bi seminar težko izvedle.