

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tim Thuma (63230333), Tilen Medved (63230207),  
Luka Hribar (63230109)

## **Frodo's Nightmares**

SEMINARSKA NALOGA PRI PREDMETU RAČUNALNIŠKA GRAFIKA

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Iztok Lebar Bajec

Ljubljana, januar 2025

## ***Abstract***

V okviru seminarske naloge za predmet Računalniška grafika smo ustvarili igro ugankarskega žanra z naslovom Frodo's Nightmares. Igra je napisana v programskem jeziku Javascript in za delovanje uporablja WebGPU vmesnik. Glavni osebek Frodo je postavljen v sceno hiše iz katere mora pobegniti tako, da rešuje uganke. Uganke so sestavljene iz različnih interakcij z objekti, postavljenimi po hiši, na koncu pa mora igralec pobrati ključ, ki odpre izhodna vrata. Scena je realistična, toda poenostavljena za lažje delo. Kamera na igralca gleda iz tretjeosebnega položaja in se premika skupaj s Frodom. Pri kameri gre za perspektivno projekcijo. V sceni je več luči, najpomembnejši pa sta petrolejka in ročna svetilka s katerima Frodo osvetljuje prostor okoli sebe. Za osvetlitev smo uporabili Blinn-Phongov osvetlitveni model. Gibanje Frodota je animirano tako, da se nad posameznimi deli telesa izvedejo ustrezne rotacije.

# 1 Pregled igre

Igralec se v igri znajde ujet v hiši sredi gozda, njegov namen pa je iz nje pobegniti. Na svoji poti mora reševati uganke, da na koncu pride do ključa, ki mu zagotovi izhod. Igra je ugankarskega žanra, vsaka uganke pa je toliko zahtevna, da je rešljiva v približno tridesetih sekundah.

## 1.1 Opis sveta

Okolje v katerega je postavljen igralec je hiša, ki vsebuje različne predmete. Prvi predmeti s katerimi se sreča igralec so tla in stene hiše, ki omejujejo prostor, po katerem se igralec lahko premika. Stena, skozi katero gleda kamera, je samo simulirana, tako da igralec ne more pasti ven iz hiše. Igralec se lahko po hiši premika v treh dimenzijah in tako raziskuje, kakšne so možne interakcije z različnimi predmeti.

Hiša ima dve nadstropji, ki sta razdeljeni na štiri sobe. V njih se nahajajo različni objekti s katerimi igralec lahko interaktira, nekateri pa so tam samo za okras. Stil tekstur je realističen, same modele pa smo poenostavili, da je delo z njimi lažje in hitrejšo.



Slika 1: Celotna scena v Blenderju.

Nekaj objektov, ki so v igri smo dobili iz spleta, nekatere pa smo izdelali sami. Sami smo izdelali Frodota, stene, ograjo, lestev, ključ, petrolejko in trampolin. Kompleksnejše okraske kot so vaze in globus pa smo dobili s spleta<sup>1 2</sup>. Vse slikovne texture kot so zid,

<sup>1</sup><https://www.cgtrader.com/3d-models/various/various-models/viking-pots-vases-low-poly-game-ready>

<sup>2</sup><https://www.cgtrader.com/free-3d-models/interior/interior-office/low-poly-world->

kamen in les smo prav tako pridobili s spleta<sup>3 4 5</sup>. Za 3D modeliranje smo uporabljali program Blender.

## 1.2 Ozadje

V ozadju se nahaja gozd, ki je prikazan tako, kot da bi bila svetu orisana kocka, na katero je prilepljena tekstura gozda (skybox), ki smo jo našli na spletu<sup>6</sup>. Hiša je obdana s tlemi, na katerih je tekstura z listi<sup>7</sup>. Ostali predmeti, ki spadajo v ozadje sveta so tudi različni okraski kot so vaze, globus<sup>8</sup> in slike na stenah<sup>9</sup>.



Slika 2: Izgled skyboxa v igri.

V hišo smo dodali tudi kroglo, na katero se preslika skybox tekstura ozadja, tako da igralec lahko vidi, kakšno je vreme izven hiše.

---

globe

<sup>3</sup><https://media.freestocktextures.com/cache/12/3e/123ef654c1c7e28fafaa983956e0509e.jpg>

<sup>4</sup>[https://pngtree.com/freebackground/wood-grain-texture-wooden-flooring-design-with-wooden-floor-textures\\_12863779.html](https://pngtree.com/freebackground/wood-grain-texture-wooden-flooring-design-with-wooden-floor-textures_12863779.html)

<sup>5</sup>[https://img.pikbest.com/wp/202344/wood-surface-exterior-view-of-log-house-wall-textured-and-characterful-knots\\_9926058.jpg!bw700](https://img.pikbest.com/wp/202344/wood-surface-exterior-view-of-log-house-wall-textured-and-characterful-knots_9926058.jpg!bw700)

<sup>6</sup><https://www.pngwing.com/en/free-png-zlcpw>

<sup>7</sup><https://cdn.materialsoftheworld.com/wp-content/uploads/2022/05/28094405/materials-of-the-world-forest-ground-leaves-render-04.jpg>

<sup>8</sup><https://www.cgtrader.com/free-3d-models/interior/interior-office/low-poly-world-globe>

<sup>9</sup><https://beks.pl/>

## 1.3 Velikost

Igralec poleg Frodota vidi še njegovo neposredno okolico. Frodo je velik približno 110 cm, kamera pa vidi približno 6 metrov širine okoli njega. Igralec interaktira z objekti, ki so približno polovični njegovi velikosti, sama hiša pa je dolga 20 m in široka 12 m.

## 2 Igralni pogon in uporabljene tehnologije

Za izdelavo igre smo uporabljali programski jezik Javascript in WebGPU v mensik. WebGPU nam omogoča, da podatke o mrežah modelov, teksturah, lučeh, kameri in ostalo pošiljamo v grafični cevovod. Te podatke nato obdelamo v senčilnikih, ki so napisani v jeziku WebGPU Shading Language (WGSL) in tako dobimo barve pikselov na zaslonu. Pri delu smo si pomagali s knjižnico `webgpu-utils`<sup>10</sup> in s primeri iz repozitorija `webgpu-examples`<sup>11</sup>.

V Javascriptu na začetku igre naložimo vse potrebne 3D modele in pripravimo grafični cevovod. Nato se za vsako sličico (frame) kliče metoda "render", ki ustrezno posodobi svet glede na trenutno stanje in vhode, ki jih je podal igralec.

Da ne pride do popačenosti tekstur, se z uporabo WebGPU ob nalaganju tekstur ustvari še mip map nivoji, kjer je dimenzija vsakega višjega nivoja enaka polovici prejšnjega. Za pravilno implementacijo trilinearne interpolacije smo si pomagali s knjižnico `webgpu-utils`.

Za zaznavanje trkov uporabljamo "axis aligned bound box"-e (AABB). Torej za vsak predmet, v katerega se lahko zaletiš izračunamo maksimalne in minimalne točke na vsaki osi in potem za vsako sličico preverjamo ali se je uporabnik zaletel. Če se je, ga premaknemo za toliko v vsaki smeri, da ni več v trku.

V drugi sobi mora igralec z miško klikniti na model gumba v sceni. Zaznavanje, če je bil gumb pritisnjen, je izvedeno tako, da se iz mesta na zaslon, kamor je igralec kliknil, sledi žarek na sceno. Za sledenje žarka moramo koordinate klika pretvoriti iz koordinatnega sistema zaslona v koordinatni sistem sveta. Če ta žarek seka AABB gumba, se smatra, da je bil gumb pritisnjen.

Za boljšo učinkovitost v grafični cevovod pošljemo samo oglišča modelov, ki so dejansko vidni na zaslonu. To pomeni, da na CPU predčasno izračunamo, kateri AABB-ji so v vidnem polju kamere. Na žalost to nima nobenega izmerljivega vpliva na hitrost delovanja, saj je naša scena premajhna.

## 3 Pogled

Za pogled v svet smo uporabili kamero s perspektivno projekcijo. Kamera je treosebna in je postavljena tako, kot da bi gledala skozi enega izmed zidov. Ko se igralec premakne, se nad kamero izvede enaka translacija, kot nad igralcem.

Po tem ko igralec pobere ključ, je kamera nekaj časa animirana, da igralec lahko vidi, katera vrata so se odprla.

---

<sup>10</sup><https://github.com/greggman/webgpu-utils>

<sup>11</sup><https://github.com/UL-FRI-LGM/webgpu-examples>

## 4 Osebek

Glavni osebek v igri je Frodo, ki ga igralec upravlja s tipkami W (naprej), A (levo), S (dol), D (desno) in presledek (skok). Nad njim ves čas deluje gravitacija s pospeškom  $20 \text{ m/s}^2$ . Frodo v roki drži luč, katere tip se lahko zamenja s tipko Q. Ko se Frodo dovolj približa izbranim škatlam, z njimi interaktira s tipko E. Ta lahko odpre škatlo ali pa jo premakne. Ko se Frodo premika, se njegovo telo animira. Animacije so izvedene tako, da je njegovo telo razsekano na različne dele, ki se rotirajo s kvadratnim lajšanjem hitrosti.

## 5 Gameplay

Igra se začne v zgornjem nadstropju hiše. Tam se igralec spozna s tipkami za interakcijo in vidi prve okraske. V tem nadstropju se nahaja tudi platforma, ki se podre, ko igralec stopi nanjo. Nato pride v prvo nadstropje, kjer se nahajajo tri sobe.

V prvi sobi so postavljene škatle, ki jih igralec lahko odpira. Ko odpre pravo škatlo, dobi petrolejko in odprejo se mu vrata v naslednjo sobo. Tam se nahajajo škatla, globus, ograja in gumb za nadzor ograje. Igralec mora z miško pritisniti na gumb, ki odpre ograjo in povleči škatlo, da lahko preskoči zid do naslednje sobe. V tretji sobi se nahaja prvi ključ, do katerega pride tako, da postavi razmetane kocke na označeno mesto, kar povzroči, da se aktivira platforma, ki preko trampolina vodi do ključa. Nato lahko pride v zadnjo sobo, kjer igralca čakata še lestev in zadnji ključ.

Ko igralec pade na tla izven hiše (torej gre skozi izhodna vrata), se igra konča.



Slika 3: Igra po končani prvi sobi spodnjega nadstropja.





Slika 4: Tretja soba z rešeno uganko in ročno svetilko kot lučjo.

## 6 Luči

V sceni so štiri luči. Prva je petrolejka, ki je izvedena kot točkovni vir svetlobe in oddaja oranžno barvo. Njena intenziteta se rahlo spreminja s časom, kot da bi se plamen v njej premikal. Druga luč je ročna svetilka, ki je izvedena kot reflektorska luč bele barve. Prehod med osvetljenim in neosvetljenim delom je gladek, kar je izvedeno s smoothstep funkcijo v WGSL. Potem pa imamo še dve točkovni luči, ki osvetljujejo vrata. Te dve luči se prižgeta, ko se začne animirana kamera. Za osvetlitveni model smo izvedli Blinn-Phongov osvetlitveni model, ki nam omogoča izvedbo neidealnega odboja na določenih predmetih.

## 7 Glasba

V ozadju se ves čas pretvarja glasba, ki smo jo vzeli iz Youtube-ove glasbene knjižnice<sup>12</sup>. Različni zvoki se predvajajo tudi, ko hodiš, skočiš in pobereš ključ. Tudi te smo vzeli s spleta<sup>13</sup>.

## 8 Zaključki in možne nadgradnje

Na začetku dela smo imeli v mislih več ugank, ampak smo se nato osredotočili samo na nekaj izbranih. Izgled igre bi lahko še izboljšali, če bi nam uspelo dodati sence v sistem osvetljevanja. Animacij s kostmi nam ni uspelo prenesti iz Blender-ja v WebGPU, tako da

<sup>12</sup><https://www.youtube.com/audiolibrary>

<sup>13</sup><https://www.zapsplat.com>

trenutne animacije ne izgledajo najbolj popolno. Gibanje kamere bi bilo lahko narejeno bolj gladko, tako da bi se kamera za igralcem premikala z zamikom.