

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 tretjeosebne položaja in se premika skupaj s Frodom. Pri kameri gre za perspektivno projekcijo. V sceni sta dve luči - "lantern", ki oddaja rdečo točkovno svetlobo in "flashlight", ki oddaja belo reflektorsko svetlobo. Za osvetlitev smo uporabili Blinn-Phongov osvetlitveni model. Gibanje Froda je animirano z linearnimi animacijami.

1 Pregled igre

Igralec se v igri znajde ujet v temni 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. To vključuje škatle, vrata, trampolin, okraske in podobno. Igralec se lahko po hiši premika v treh dimenzijah in tako raziskuje, kakšne so možne interakcije z različnimi predmeti. Hiša se nahaja v gozdu, ki je prikazan s skybox ozadjem, torej okoli sveta je orisana kocka, ki je enakih dimenzij kot zaslon (TO NEVEM ČE JE ČIST RES). Stil tekstur je realističen, same modele pa smo seveda poenostavili, da je delo z njimi lažje in hitrejšo.

1.2 Pregled

Prvi predmeti s katerim se sreča igralec so tla in stene hiše, ki omejuje prostor, po katerem se igralec lahko premika. Stena, skozi katero gleda kamera, je samo simulirana, tako da igralec ne more pasti ven iz hiše. Hiša ima dve nadstropji in igralec začne v zgornjem nadstropju. Tam se 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 ključne sobe.

V prvi sobi so postavljene škatle, ki jih igralec lahko odpira in ko odpre pravo škatlo, dobi petrolejko in odprejo se mu vrata v naslednjo sobo. Tam se nahaja še ena škatla, globus, ograja in gumb za nadzor ograje. Ko igralec reši to uganke, pride v tretjo sobo, kjer se nahaj prvi ključ, do njega pa pride preko interakcije s še več kockami, trampolinom in premikajočo platformo. V zadnji sobi pa igralca čaka še lestev in zadnji ključ.

Da ne pride do popačenosti tekstur, se z uporabo WebGPU ob nalaganju tekstur ustvarijo 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`¹.

1.3 Ozadje

V ozadju se nahaja gozd, kar je izvedeno s skybox teksturo, ki smo jo našli na spletu². Hiša je obdana s tlemi, na katerih je tekstura z listi³. Ostali predmeti, ki spadajo v ozadje sveta so tudi stene hiše in različni okraski kot so vaze⁴, globus⁵ in slike na stenah. Slike so vzete iz publikacije Zdzisław Beksiński (TO DAJ BIBLIOGRAFIJO) <https://beks.pl/beksinski-obrazy/>

SLIKA SKYBOXA

¹<https://github.com/greggman/webgpu-utils>

²<https://www.pngwing.com/en/free-png-zlcp>

³

⁴<https://www.cgtrader.com/3d-models/various/various-models/viking-pots-vases-low-poly-game-ready>

⁵<https://www.cgtrader.com/free-3d-models/interior/interior-office/low-poly-world-globe>

Ozadje pa seveda ni velikokrat vidno, saj večino zaslona zasedajo zidovi. V hišo smo dodali tudi kroglo, na katero se preslika tekstura ozadja, tako da igralec lahko vidi, kakšno je vreme izven hiše. V hiši se nahajajo tudi različni okraski, s katerimi igralec ne more interaktivirati. Sem spadajo ...

1.4 Ključne lokacije

TO ZDRUŽI S “PREGLED” Svet je razdeljen na več sob, med katerimi igralec prehaja, ko rešuje uganke. Ciljna lokacija je izhod skozi zadnja vrata, ki se odprejo, ko igralec na eni izmed platform pobere ključ.

1.5 Velikost

Igralec poleg Frodota vidi še njegovo neposredno okolico tako, da je njegov pogled na svet precej ozek. Igralec interaktivira z objekti, ki so približno polovični njegovi velikosti, sama hiša pa je velika toliko, da se jo da po dolžini prehoditi v nekaj deset sekundah.

1.6 Objekti

Vse slikovne texture kot so zid, kamen in les smo pridobili s spleta (LINKS). Objekt, ki ga igralec največ premika je Frodo, ki smo ga sami zmodelirali v Blender-ju. Ta pa v bližini drugih objektov lahko spreminja tudi njihovo stanje. Med te sodijo škatle, ki se lahko odpirajo oz. premikamo. Frodo v roki po končanem prvem nivoju drži svetilko iz katere sveti luč, na koncu pa ga čaka ključ. Da pride do ključa mora uporabiti trampolini, ki ga odrine dovolj visoko, da lahko pristane na končni platformi. Tudi te modele smo naredili sami.

2 Igralni pogon

Za izdelavo igre smo uporabljali programski jezik Javascript in WebGPU vmesnik. WebGPU nam omogoča, da pošiljamo podatke o mrežah modelov, teksturah, lučeh, kameri in ostalo v grafični cevovod. Te podatke nato obdelamo v senčilnikih, ki so napisani v jeziku WebGPU Shading Language (WGSL) in tako dobimo barve pikslov na zaslonu. Za 3D modeliranje smo uporabljali program Blender. Pri delu si nismo pomagali z nobeno drugo knjižnico oziroma ogrodjem, smo si pa kar precej pomagali s primeri iz repozitorija webgpu-examples (<https://github.com/UL-FRI-LGM/webgpu-examples>). V Javascriptu na začetku igre naložimo vse potrebne 3D modele in pripravimo grafični cevovod. Nato pa se za vsako sličico (frame) kliče metoda render, ki ustrezno posodobi svet glede na trenutno stanje in vhode, ki jih je podal igralec.

3 Pogled

Za pogled v svet smo uporabili kamero s perspektivno projekcijo. Kamera je tretjeosebna 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.

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 m/s^2 . Frodo v roki drži luč. Tip luči lahko zamenjamo s tipko Q. Možni luči sta “lantern”, ki oddaja rdečo točkovno svetlobo in “flashlight”, ki oddaja belo reflektorsko svetlobo. 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 z linearnimi animacijami. Prav tako je animirano odpiranje vrat.

5 Uporabniški vmesnik

OPIŠ START SCREEN AND END SCREEN

6 Gameplay

Igra se začne v prvi sobi, kjer je postavljenih šest škatel, v eni izmed njih pa se nahaja luč. Igralec odpira te škatle in ko najde luč, se mu odprejo vrata do naslednje sobe. Tam mora na ustrezno mesto prestaviti novo škatlo, da lahko preskoči zid in pride do zadnje sobe. Tam se najprej sreča z uganko, kjer rabi razmetane škatle postaviti na ustrezno označeno mesto, nato pa z uporabo trampolina skoči do ključa, ki mu odpre vrata, da gre lahko ven. S tem igro premaga in na zaslonu se mu izpiše, koliko časa je porabil za reševanje

7 Luči

TUKAJ OPIŠI SHADERJE

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 narediti mapiranje senc. Animacij s kostmi nam ni uspelo prenesti iz Blender-ja v WebGPU, tako da trenutne animacije ne izgledajo najbolj popolno. Lahko bi ji dodali tudi predele z animirano kamero, npr. Največ samostojnega dela smo imeli z izdelavo 3D modelov, saj nam je veliko kode za uporabo WebGPU napisal že asistent.

TO NEVEM U KERO POGLAVJE SE DA

Preden se igra začne, se igralec najprej seznani z gumbi, ki jih bo uporabljal med igro (TO SE ŠE DODA). To je izvedeno kar s HTML elementi in se skrije, ko se igra začne. Na koncu igre igralec vidi, koliko časa je porabil za igranje in ima na voljo ponoviti igro.

Frustum culling - iz okoli 200 modelov na render do okoli 60. Se sicer ne pozna med samim izrisom, ampak je hitrejši loading time, ker node shrani v `this.gpuObjects` šele, ko ga res rabi. Ni pa izmerljive razlika, ker je naša scena dovolj majhna.

Background music: yt audio library

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