

iSYSTEM Labs tekmovanje: izdelava igre za platformo Miško3 - Terrsko

Gabrijel Okorn, Sebastijan Trojer

maj 2023

1 Opis projekta

Terrsko je prilagojena različica igre Terraria, ki temelji na preprosti a hkrati razburljivi ideji: gradnji in preživetju. Igralec vstopi v svet igre, kjer lahko razišče nov svet in gradi svoje domove. Igra je polna raznih materialov, ki jih igralci lahko uporabljajo za oblikovanje svojega sveta, gradnjo stavb ter raziskovanje okolice.

1.1 Opis igre in mehanik

Igra ima dva načina - osnovni način in način gradnje, ki ga aktiviramo z gumbom OK. V osnovnem načinu z joystickom upravljamo junaka, s smernimi gumbi pa lahko kopljemo v izbrano smer. Način gradnje nam po drugi strani omogoča polaganje poljubnih trdnih materialov v svet. S smernimi gumbi določimo koordinato, kamor želimo da se blok izriše, s tipko OK pa material tja tudi položimo. Ta način zapustimo s pritiskom tipke ESC. Branje gumbov je implementirano s prekinitvami, branje joysticka pa deluje preko DMA.

Ko dosežemo rob sveta, gremo lahko čez in se ponovno pojavimo na drugi strani.

2 Opis implementacije in delovanja

Svet v Terrsko je zgrajen iz blokov, velikosti 4x4 pikslov, kar omogoča dovolj podroben izgled okolja. Glavni koncepti igrice Terrsko so igralec, svet in NPC-ji, ki se sprehajajo po svetu. Sledi opis njihove implementacije.

2.1 Svet

Močno omejen prostor za shranjevanje podatkov (128kB RAM) je zahteval premišljen pristop k gradnji igre. Odločila sva se, da materiale v svet zakodirava samo s štirimi biti, ter tako prihraniva 50% prostora, v primerjavi z

uporabo vseh 8 bitov, hkrati pa še vedno dopustiva uporabo 16 različnih materialov. Taka implementacija zahteva računanje vrednosti materialov na vsako sličico, saj je prostora dovolj zgolj za 4 bitno zastavico, iz katere se nato morajo izračunati barva, koordinate in ostale lastnosti (kolizija, uničljivost itd.). Pomankanje prostora tudi ne dopušča implementacije osvetlitve, saj je sprotnega računanja preveč, prostora za dovolj veliko matriko predizračunanih števil pa ni. Trenutna različica projekta z raznimi optimizacijami teče pri 100 FPS.

2.2 Implementacija igralca

Igralec in vsi neigralski liki v igrici Terrsko imajo implementirane nekatere sile, kot smo jih vajeni iz resničnega sveta. Gravitacija je upoštevana pri vsakem liku in vpliva na padanje lika v vertikalni smeri. To je implementirano tako, da se njegova pozicija v svetu spreminja pospešeno. Igralec je občutljiv na svojo okolico in se ob dotiku trdih snovi ustavi. V primeru, da se nahaja v lavi ali pa da je padel z višje pozicije, izgubi življenjske točke, zaradi česar lahko umre.

2.3 NPCji

V Terrskotu živijo tudi drugi ljudje, kot na primer rudar in čarodej, izgleda pa da so bolj asocialni kot je videti na prvi pogled. Prav tako ima Terrsko raznoliko biodiverziteto krav. Za varčnost s prostorom so slike kompresirane na sledeč način - matrika barv je razdeljena na dva dela: seznam barv in matrika, ki predstavlja sliko, le da vsaka vrednost na nekem indeksu sovpada z indeksom barve, hkrati pa vsaka taka vrednost zavzema zgolj 4 bite. Kompresija je potrebna, saj drugače modeli objektov zavzamejo preveč prostora, tako pa namesto $w \times h \times 2$ bajtov za predstavitev modela potrebujemo samo $w \times h \times \frac{1}{2}$ bajtov.

2.4 Omejitve in problemi

Izrisovanje ostaja časovno zahtevna operacija, kar igrici Terrsko omeji število NPC-jev in dreves. V ta namen bi bilo potrebno implementirati hitrejši algoritem za izrisovanje slikic. Izziv nama je predstavljala tudi osvetlitev. Je pomemben del igrice, ki je že implementiran, a ni vključen, saj je njegov izračun kompleksen, kar upočasni igro in vpliva na igralčevo izkušnjo. Tudi ta algoritem potrebuje optimizacijo pri izračunavanju. Zaradi težav pri izrisovanju sva povečala flash latency na 10 period.

2.5 Napredne programske rešitve

2.5.1 Oblikovanje sveta

Ko govorimo o naprednih programskih rešitvah, se je vredno ustaviti pri algoritmih, uporabljenih pri oblikovanju sveta. Prvi tak algoritem je diamond-square algoritem, ki se uporablja za generacijo razgibanega terena. Ta algoritem je sicer zasnovan za 3D okolja, ampak je bil prirejen za naše potrebe. Rezultat tega algoritma je nato zglajen z Gaussovim filtrom, saj je teren sicer še vedno

preveč stopničast. Podobno so v dveh korakih generirane jame - v prvem koraku matriko jam generiramo z Cellular automata algoritmom, ki poskrbi za lepo porazdelitev in povezavo jam, v drugem koraku pa z morfološkimi operacijami, kot so odpiranje in zapiranje, te jame dodatno oblikujemo, da ustrezajo našim potrebam.

2.5.2 Upoštevanje hitrosti igre

Ob igranju želimo, da se hitrost premikanja likov in igralca ne spreminja. To pomeni, da na to, kako hitro se bomo premikali ne sme vplivati hitrost delovanja mikroprocesorja, saj ta lahko ob večjih zahtevah tako počasneje izrisuje na zaslon. Ta problem sva rešila z uporabo časovnika, s katerim meriva čas, ki preteče od začetka do konca 'frame-a' in tako nadzorujemo, koliko časa bo prikazana ena slika, preden jo bo nadomestila druga.

2.5.3 Hitrostno izrisovanje sveta

Za mikroprocesor je izrisovanje slike na zaslon časovno zahtevna operacija. Tako si nisva morala privoščiti preprostega izrisovanja vsake točke posebej na zaslon, saj je svet kmalu začel zaostajati. V ta namen sva implementirala optimizacijsko metodo blitting, s katero sva za vsak blok preverjala ali se je njegova podoba iz prejšnjega 'frame-a' spremenila, ter tako drastično zmanjšala čas, potreben za izris.

3 Projektno vodenje

Razvoj projekta je močno olajšala uporaba Githuba, ki je omogočal enostavno nadziranje verzij igre. Github je bil še posebno pomemben, ker sva delo imela razdeljeno na dva dela - razvoj junaka ter mehanik, vezanih nanj, ter razvoj različnih načinov generiranja čimbolj zanimivega sveta, kljub veliki prostorski omejitvi. Večina dela je torej bila ločena, nekatere dele pa sva vseeno implementirala skupaj. Za poenostavljeno risanje na zaslon sva uporabila knjižnico μ GUI.

4 Povzetek

Z upoštevanjem vseh omejitev, sva razvila igro, ki zaenkrat vsebinsko še ni zelo dopolnjena, ima pa vse funkcionalnosti in mehanike, ki jih neka taka igra potrebuje, tako da predstavlja zelo dobro izhodišče za nadgradnje. Na razpolago je še cca. 50kB RAMa, tako da tudi prostor ne bi smel več biti problem, ob razvijanju pa sva vedno imela v mislih tudi najboljše prakse za čimlažje dodajanje razširitev.