

Poročilo razvoja igre Balls to the Holes

Nejc Hirci¹, Aljaž Grubar²

¹Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

²Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Povzetek

V igri *Balls to the Holes*, sva želela na čimbolj zanimiv način izkoristiti okolje proceduralno generiranih otokov. Zato sva se odločila razviti 3d »platformer«, v katerem igramo kroglo, ki se želi uspešno izogniti oviram na poti do magične cevi skrite na vsakem otoku, s pomočjo katere bo lahko napredovala na naslednji otok. Generiranje otokov sva implementirala s pomočjo prirejenega Perlinovega šuma, za fizikalni pogon pa sva uporabila knjižnico Oimo.js.



Slika 1: primer igralne izkušnje

1 Pregled igre

Tako kot večina predstavnikov žanra 3d »platformerjev«, je igra precej preprosta v načinu igranja z jasnimi cilji in preizkuša predvsem igralčeve reakcijske spretnosti. Vseeno pa za nevedne igralce kombinacija tretjeosebne kamere in počasnih odzivov žoge na interakcijo zahteva morda nekaj več spretnosti.

Zgodba igre se vrti okoli razkačene Lune, ki se je znašla izgubljena na otoku sredi ničesar daleč od svojega doma. Poleg vsega se je izkazalo, da je otok izjemno neprijazen za življenje, saj na njem neprestano dežujejo nevarne bodičaste krogle in so oceani zaradi prevelikih koncentracij CO₂ postali izjemno korozivni. Na srečo pa se na otokih skrivajo magične cevi na dnu katerih se morda skriva rešitev za naš glavni lik. Poleg tega so prejšnji prebivalci otokov prijazno pozabili magične lebdeče »hotdodge«, ki lahko ponudijo edini vir energije mimoidočim popotnikom.

1.1 Opis sveta

Svet igre je postavljen na naključno generirane otoke z različnimi vegetacijami obdanimi z za igralca neskončnim oceanom. Otok je prepleten s padajočimi

bodičastimi krogli in obdan z oceanom strupene vode. Poleg tega se na njemu lahko najdejo tudi lebdeči »hotdodge« in magična cev, ki vodi na naslednji otok. Slog sveta je izjemno stiliziran, zato je vegetacija otokov predstavljena le s preprostimi barvnimi shemami in so modeli objektov relativno enostavni. Vsi dinamični objekti v igri se lahko premikajo po treh dimenzijah.

1.1.1 Ozadje

Ozadje sveta od igri, s katerim igralec ne more interaktivirati je zgolj nebo v ozadju in oddaljeni oceani, ki so predstavljeni kot preproste texture nalepljene na primitive ozadja kamere.

1.1.2 Ključne lokacije

Najpomembnejša lokacija v igri je rdeča cev, ki je naključno skrita na otoku in predstavlja cilj enega nivoja igre. Poleg tega je pomembno, da se igralčeva krogla pojavi vedno v enem od robov otoka in ob morebitni izgubi bo igralec ponovno začel igro iz tega mesta.

1.1.3 Velikost

Velikost sveta je omejena na teren enega otoka. Pri vsakem nivoju se lahko igralec giblje po otoku in spremlja padajoče bodičaste krogle, ki se jim mora izogniti.

1.1.4 Objekti

Proceduralno generirani otoki so ustvarjeni s pomočjo Perlinovega šuma. Najprej sva ustvarila višinsko sliko otoka, tako da sva seštel več oktav Perlinovega šuma s padajočimi utežmi, da sva dobila bolj zanimive in raznolike podrobnosti terena. Nato sva sliko množila z masko padajočo glede na razdaljo od središča, ki sva jo generirala z logistično funkcijo, da sva dobila primerno višinsko sliko za otok. Nazadnje sva glede višine primerno normalizirala z lepljenko, da sva dobila lepe sloje vegetacije in voda ter ustvarila teksturo. V pomoč nama je bila serija posnetkov avtorja Sebastiana Lague.¹

Rdeča cev predstavlja magično pot iz otoka. Luna z obrazom je krogla, ki jo upravlja igralec, bodičaste krogle pa so ovire na njegovi poti do cilja. Vse objekte z izjemo otokov sva pridobila iz spleta in jih uvozila v igro preko ene GLTF datoteke.

1.1.5 Čas

Čas je realističen, tako je 1 minuta v igri ekvivalentna 1 minuti v realnem času.

1.2 Igralni pogon in uporabljene tehnologije

Za fizikalni pogon sva uporabila že narejeno knjižnico Oimo.js², ki sva jo uporabila za računanje trkov med posameznimi objekti v sceni in za računanje njihovih pozicij v skladu z gravitacijo in trenutnimi hitrostmi.

1.3 Pogled

V igri sva uporabila tretjeosebno kamero, ki se rotira okoli igralčeve krogle. Tako lahko igralec med igro spremlja, kaj se dogaja povsod okoli njega v svetu.

2 Osebek

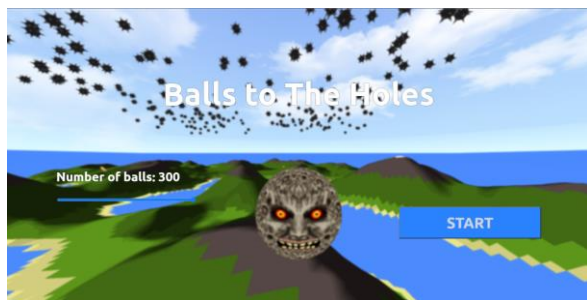
Glavni osebek v igri je luna, ki jo upravlja igralec. Njene premike po tereni lahko kontrolira s tipkami W, A, S, D, ključnega pomena za premikanje po terenu pa je tudi skok, ki ga sproži tipka spacebar.

3 Uporabniški vmesnik

Uporabniški vmesnik sestavljajo **začetni meni**, ki vsebuje gumb za začetek igre in pa tudi drseče okno za spreminjanje števila žog v sceni, pokazatelj **življskih točk**, ki se nahaja zgoraj levo in **končni meni**, ki vsebuje možnosti za napredovanje na naslednji nivo igre. Kadarkoli med igro lahko igralec pritisne tipko escape za ustavitev igre, kjer ima tudi možnost za ponovni poizkus.

Vmesnik je enostavno narejen iz html značk, večinoma gumbov, ki jim glede na uporabnikovo interakcijo nastavlja vidljivost in vrednosti. Pokazatelj življskih točk (njegova animacija) je narejen z uporabo knjižnice jQuery.

Takšna implementacija se nama zdi dovolj preprosta in še vedno nudi vse potrebne funkcionalnosti.



Slika 2: začetni meni

4 Glasba in zvok

V igri je kot glavna melodija uporabljena glasba, ki sva jo pridobila na spletu. Glasba se začne, ko uporabnik prične z igranjem nekega nivoja in pripomore k vzdušju igre. Poleg glavne melodije sva uporabila tudi kratke zvočne posnetke, ki se sprožijo, ko žoga izgubi življske točke ali poje »hotdog«. Posebni melodiji se predvajata tudi v primeru smrti in zaključku nivoja. Vse melodije in zvočne efekte sva pridobila na spletu.

5 Gameplay

Igra se prične, ko igralec nastavi željeno količino špičastih žog in pritisne na gum za začetek. Takoj se začnejo nad terenom generirati špičaste žoge, ki se jim more igralec skupaj z vodo spretno izmakniti in poiskati rdeč cilinder, ki mu omogoči napredovanje na naslednji nivo. Med iskanjem cilindra si lahko pomaga s pobiranjem »hotdogov«, ki mu dajo 30 življskih točk in se nahajajo nekje na terenu. Igra se konča šele, ko igralec umre oziroma preneha z igranjem, cilj pa je doseči čim več nivojev. Nivoji so naključno generirani tako, da je vsaka izkušnja unikatna.

6 Zaključki in možne nadgradnje

Pri izdelavi igre sva se naučila veliko o grafiki in izrisu posameznih objektov v sceni. Naučila sva se tudi osnove proceduralnega generiranja terena in izboljšala znanje JavaScripta. Pri predmetu sva še posebej v sklopu vaj pridobila vso potrebno znanje za izdelavo igre. V igro bi si želela dodati še več različnih ovir in dodatnih objektov za pobiranje, ki bi igralcu začasno dodali določene zmožnosti. Želela bi si tudi implementirati izris senc in bolj raznoliko generiranje terena, vendar bi bilo za vse to potrebno še nekaj več časa.

Literatura

- [1] James D. Foley, Richard L. Phillips, John F. Hughes, Andries van Dam, and Steven K. Feiner. Introduction to Computer Graphics. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.
- [2] Johannes Meng, Marios Papas, Ralf Habel, Carsten Dachsbaecher, Steve Marschner, Markus Gross, and Wojciech Jarosz. Multi-scale modeling and rendering of granular materials. ACM Transactions on Graphics (Proceedings of SIGGRAPH), 34(4), July 2015.

² <https://github.com/lo-th/Oimo.js/>