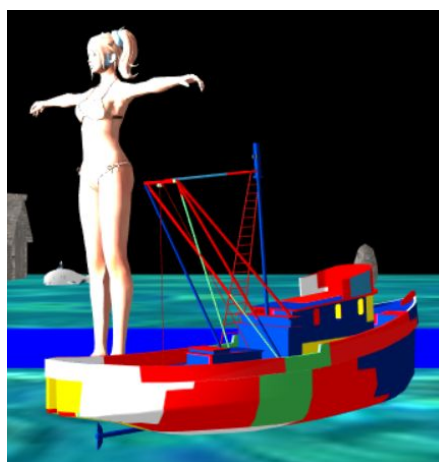




Univerza v Ljubljani
Fakulteta *za računalništvo*
in informatiko

VIDIMO SE NA MORJU

(seminarska naloga pri računalniški grafiki)



Avtorja: **Anže Luzar in Tomaž Nemanič**

Ljubljana, 2019

KAZALO VSEBINE

1. UVOD	2
2. ZASTAVLJENI CILJI	3
3. STRUKTURA PROJEKTA	3
4. PREDSTAVITEV OKOLJA	3
5. POSTAVITEV KAMERE	4
6. OBJEKTI	4
7. IZDELAVA SEMINARJA V WebGL-u	5
8. DODATNA GLASBA	6
9. REALIZACIJA ZAZNAVANJA TRKOV	7
10. NAVODILA ZA IGRO	7
11. ZAKLJUČEK	8

1. UVOD

Najina seminarska naloga pri računalniški grafiki ima naslov “SE VIDIMO NA MORJU”, saj naju je morje vedno navduševalo. Želela sva narediti igro, ki bi imela možnost plovbe z ladjo po morju.

2. ZASTAVLJENI CILJI

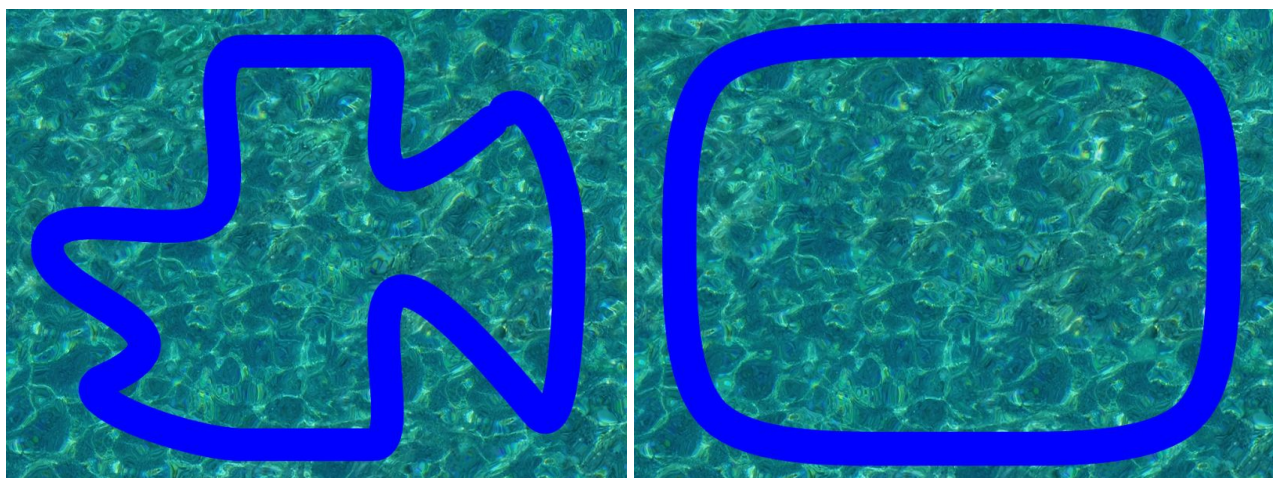
Zastavila sva si cilj, da bova naredila projekt z glavno idejo prevažanja ljudi, in sicer z ladjo. Svet z morjem in otoki bo obarvan in osvetljen ter mogoče tudi osenčen (morda bi v sam projekt dodala tudi menjavanje dneva in noči, da bi igralec ločil razliko med plovbo podnevi in ponoči). Pri prikazu morja bova uporabila elemente taselacije (mozaičenja), da bo videz morske vode naraven. Ladje s potniki bo možno premikati (translirati) skozi vodo do otokov, kjer bodo na ladje vstopili gostje in muzikantje, ki bodo z bučno glasbo poskrbeli za pravo vzdušje na plovbi, kar bo tudi dodano v samo igro kot zvočni efekt. Planirava, da bi ob igranju dodala dejanske zvoke različnih sproščujočih pesmi. V igro bo možno vstopiti z različnimi plovili, ki jih bova implementirala (katamarani, barke, gliserji in možnost vožnje z smučmi na vodi), poleg tega pa bo igra vsebovala več zaporednih nalog, ki bodo nagrajene z dodatnimi razširitvami glede na uspešnost prevoza potnikov v pravem času na cilj. V igri bi bilo potrebno opraviti nalogo z ladjo v zvezi s prevozom po morju (npr. tovara, ljudi itd.), na poti pa bi dodala še razne ovire (npr. čeri, vetrovne razmere itd.). Pri tem bova poskušala dodati tudi upoštevanja ovir (collision detection) pri tem, ko se ladja npr. zaleti v čer. Ideja projekta je, da se igralci lahko naučijo nekaj o ravnanjih pri plovbi na morju, hkrati pa tudi sprostitev ob dobri glasbi, ki bo vključena v projekt. Morda bo sama igra zamišljena kot neka zgodba, kjer bodo vključeni določeni liki, s katerimi se bo lahko igralec poosebil. Kodo za projekt bova napisala v WebGL-u.

3. STRUKTURA PROJEKTA

Najin projekt vsebuje več imenikov, ki so potrebni za delovanje same igre in sicer imava več html datotek, ki služijo prikazu zaporednih strani, iz katerih sestoji najin seminar. Imava uvodno stran (index.html), stran za igro (game.html), končno stran (end-win.html ali end-lose.html), mapeditor.html, ki služi možnosti prilagajanja morske poti, po kateri pluje ladje, navodila.html z navodili. Imava datoteke s slogi css skoraj za vsako izmed strani. Slednje se nahajajo v imeniku styles. V imeniku assets imava objekte, ki so shranjeni v obliki obj in mtl ter še json, poleg tega pa so tu še texture morja, škatel, rib, žensk, ladij itd. Imenik audio vsebuje ogg datoteke z večinoma morskimi in dalmatinskimi pesmimi, ki med samo igro dajejo vzpodbudo igralcu. V imeniku skript je jedro igre - skripte napisane v javascriptu. Glavna skripta je script.js, ki je povezana z ostalimi skriptami za objekte (npr. shark.js in boat.js).

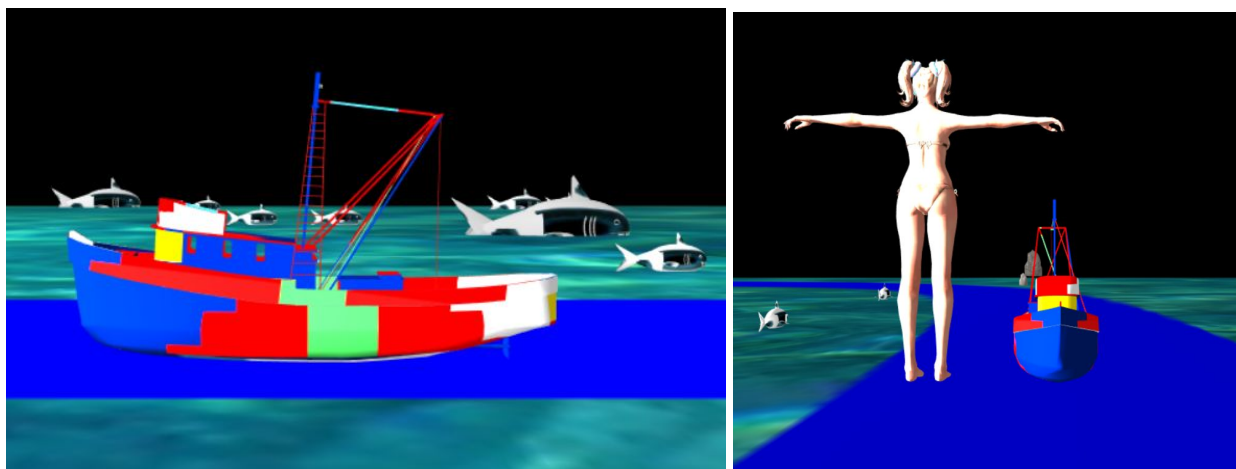
4. PREDSTAVITEV OKOLJA

Okolje sva postavila v morje v obliki kvadrata, progo pa kot neko dirkalno stezo, po kateri naj bi ladja potovala. Da igra ne bi bila preveč enostavna sva dodala ženske, ki igralca zavajajo, da se ustavi. Tudi ribe so tu, da bi zapolnile samočo v pustem morju. Zakaj sva dodala še grad? Enostavno zato, ker sva gledala na celoto in simetričnost. Zadaž pa se skriva skrivnostna zgodba, ki pravi, da se v gradu skriva poseben zaklad, ki bi zopet zavedel igralca, vendar zakladi oziroma škatle so le na progi. Imava tudi skale in čeri različnih velikosti, ki ponazarjajo lažne ovire oziroma privid.



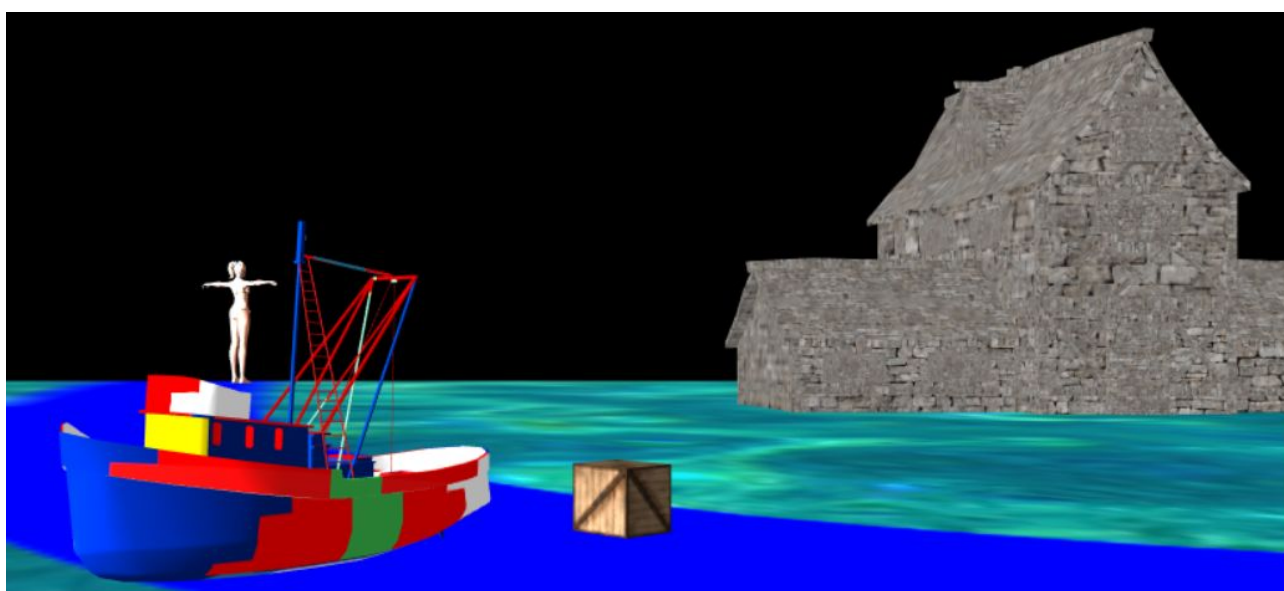
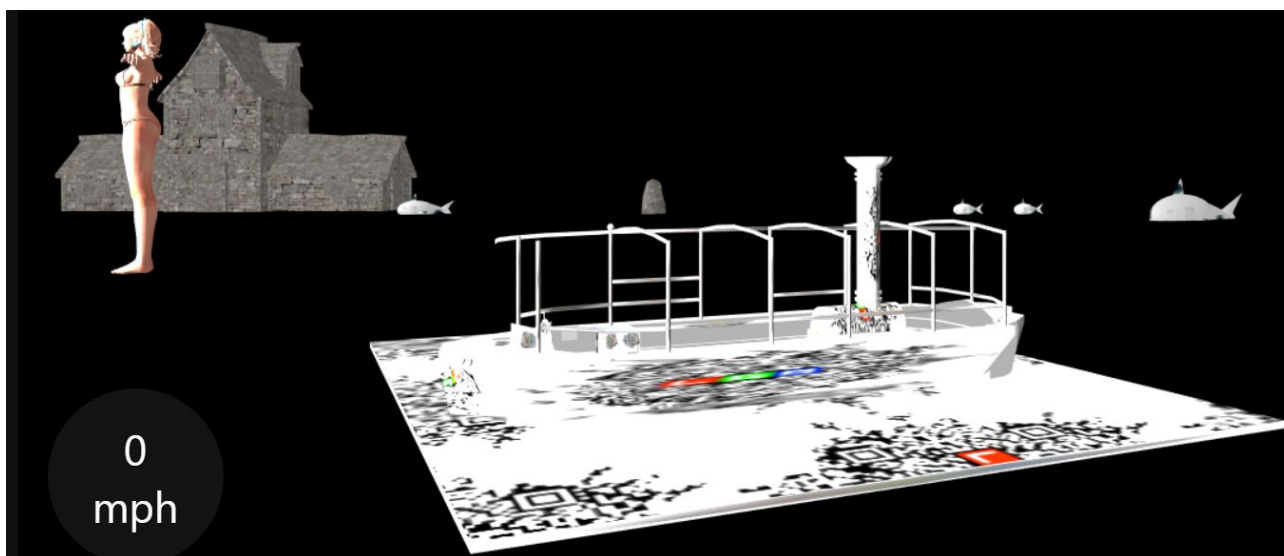
5. POSTAVITEV KAMERE

Pogled sva nastavila kot 3.osebni, torej kamera spremlja ladjo kod potuje ter prikazuje celotno ladjo z okolico. Ta oblika se nama je zdela najprimernejša, saj omogoča lažje igranje in preglednost.



6. OBJEKTI

Najprej sva si izbrala objekte, ki sva jih kasneje postavili na sceno. Pregledala sva različne modele ladij in plovil, ki jih bi lahko uporabila (predvsem sva iskala po clara.io, prenesla pa sva tudi program blender za izris in pretvorbo 3d modelov. Tako sva izbrala 3d modele za ladjo, morske pse, hišo in skale ter ženske. Prenesla sva 3d modele in tako dobila format mtl in obj, ki sva ga pretvorila še v json. Najina igra je napolnjena z veliko objekti. To se nama je zdelo ključno zaradi pestrosti izgleda igre in boljšega doživetja. Glavni objekt je ladja, ki potuje po morju in ob pobiranju zakladov spreminja svojo obliko in ne. Tukaj so tudi ženske, ki zavajajo igralca, da se ustavi in tako nepravčasno zaključi 3 kroge, torej izgubi. Kot del morja sva dodala tudi ribe. Sredi proge je grad, ki je daje celoto in simetričnost teksturi.



7. IZDELAVA SEMINARJA V WebGL-u

Za osnovo izdelave seminarja sva vzela skripto iz vaje imenovano Doom. Nato sva začela pisati skripte, ki bi omogočale to, kar sva želela. Sprva sva definirala spremenljivke (za platno, kontekst gl, uro, števec krogov, objekte, bufferje, različne matrike - za pogled/kamero itd, texture, položaje na platnu, checkpointe in ostale pomožne spremenljivke). Spisala sva koordinate na katerih so checkpointi, funkcije za operacije z matrikami (pop in push), pretvorbo stopinj v radiane, inicializacijo WebGL-a, naložila texture in objete s funkcijo load (ko sva nalagala objekte sva podala kot pot do json datoteke XMLHttpRequest-u, pri teksturah pa pot do texture). Za vsak objekt imava svojo js datoteko, ki vsebuje razred z imenom tega objekta. Tem razredom sva dodala attribute (npr. velikost) ter funkcijo draw, ki objekt nariše na podani lokaciji v argumentih in še funkcijo handleLoaded, ki po prejetju json podatkov naloži podatke v ustrezne bufferje. Pri ladji sva imela nekoliko več atributov in funkcij, saj ima ladja še svojo hitrost, pospešek, velikost ter enako funkcijo draw in handleLoaded ter še animate, ki ustrezno posodobi koordinate. V funkciji drawWorld nariševa svet, ki prebere svoje koordinate iz tekstovne datoteke v assets - to je world.txt. Do izrisa objektov pride v funkciji drawScene, kjer izriševa ladjo, ženske, skale, grad itd. Imava tudi funkcijo animate, ki posodablja časovnik in kot pogleda ter funkcije za upravljanje s tipkovnico

(handleKeys), ki omogočajo, da lahko ladjo premikamo npr. s tipkami A, S, D, W. Tu je še funkcija za preverjanje tega, ali je ladja na progi in

```
class Shark {
  constructor() {
    this.scaleFactor = 1;
  }

  setScale(scaleFactor) {
    this.scaleFactor = scaleFactor;
  }

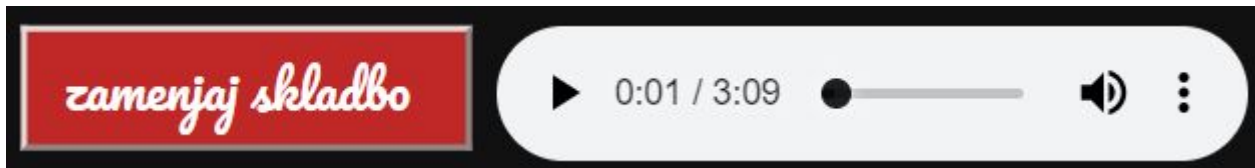
  draw(location, directionAngle) {
    mat4.translate(mvMatrix, location);
    mat4.rotate(mvMatrix, degToRad(directionAngle + 180), [0, 1, 0]);
    // mat4.rotate(mvMatrix, degToRad(180), [0, 0, 1]);
    mat4.scale(mvMatrix, [1 * this.scaleFactor, 1 * this.scaleFactor, 1 * this.scaleFactor]);

    // Activate textures
    gl.activeTexture(gl.TEXTURE0);
    gl.bindTexture(gl.TEXTURE_2D, sharkTexture);
  }
}
```

handleTrack, ki skrbi za progo in checkpointe na njej. Funkcija handleCheckpoints preverja, ali smo že slučajno prevozili tri kroge oz. ali je pretekel čas dveh minut in smo izgubili. Dodala sva še funkcijo za preverjanje tega, ali se plovilo nahaja na checkpointu, saj se v skladu s tem obarva zeleno checkpoint na najini minimapi levo zgoraj, ki prikazuje, kje se ladja nahaja na zemljevidu. Najin program zaženeva s funkcijo start, kjer je med drugim tudi del, v katerem se glede na vrsto pobranega zaklada v switch stavku odločiva, kaj se zgodi z ladjo - ali postane večja/manjša, hitrejša/počasnejša, ali se zamenja tekstura in oblika ladje itd.

8. DODATNA GLASBA

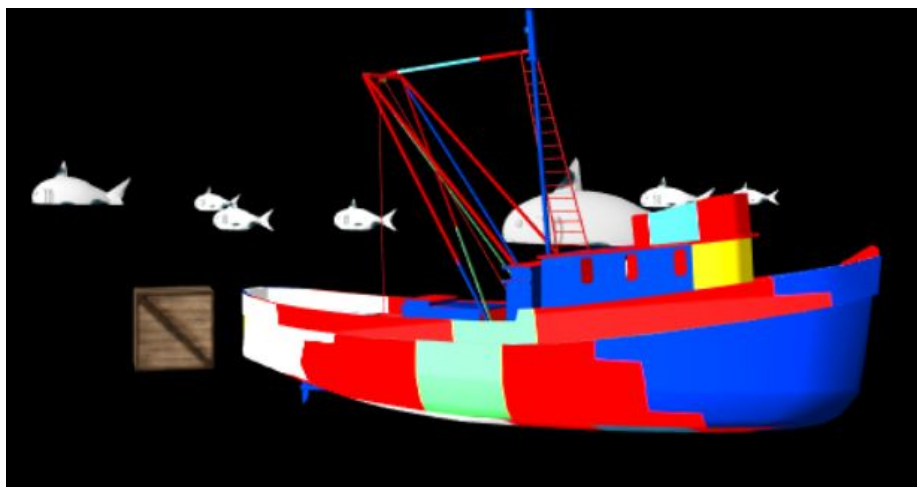
Za še večji užitek ob igranju sva dodala prijetno glasbo. Pesem lahko izbiraš in po želji nastavljaš glasnost. Za tiste, ki raje igrajo v tišini, obstaja tudi možnost pavziranja glasbe.



9. REALIZACIJA ZAZNAVANJA TRKOV

Ker je naše območje omejeno sva morala opraviti še z zaznavanje in realizacijo trkov. Ko pride ladja do roba roba, jo odbije na začetek proge. To sva realizirala s funkcijo handleTrack, kjer preverjava, ali je šla ladja izven koordinat sveta in glede na to ustrezno po potrebi nastavlja nove koordinate. Naredila sva realizacijo pobiranja škatel, ko se ladja zaleti v škatlo, to pa narediva s preprosto funkcijo handleCollision, v kateri izračunava razdaljo med točkama, na katerih je škatla in na katerih je ladja (pri čemer upoštevava le koordinati x in z, saj se y ne spreminja ker smo stalno na plošči) in če je ta razdalja dovolj majhna smo trčili v zaklad in zaklad dobi nove koordinate.

```
// Funkcija, ki reagira na trke
function handleCollision() {
  if (Math.sqrt(Math.pow(xPosition - randomPoint[0], 2) + Math.pow(zPosition - randomPoint[1] - 10, 2)) < 2) {
    randomPoint = getRandomPointOnTrack();
    boxPosition = [randomPoint[0], 1, randomPoint[1]];
    box.generate();
  }
}
```



10. NAVODILA ZA IGRO

Igro začnemo z s klikom na gumb “DVIGNI SIDRO”. Za pogon in pospeševanje stisnemo tipko UP ali W, za zavijanje levo ali desno bodisi LEFT ali A bodisi RIGHT ali D na tipkovnici. Ukaze sva izbirala glede na pogostost uporabljanja v svetu. Cilj igre je, da uspešno prevoziš tri kroge v dveh minutah. Pri tem pa naj te čim manj zavedejo pasti. Pazi tudi na zaklade, ki ti lahko ali povečajo ali zmanjšajo hitrost (pomisli ali se ti splača žrtvovati glede na čas, ki ti preostane). Za to, da igra v popolnosti deluje moramo imeti postavljen webserver (npr XAMPP ali usbwebserver itd.)



11. ZAKLJUČEK

Zaključimo pa lahko, da sva delo opravila uspešno, saj sva več kot pol ciljev izpolnila. Glede na čas, ki sva ga imela na voljo, je projekt dobro uspel, pri tem pa sva se naučila veliko novega in spoznala, da je web-gl zelo kompleksen tako za razumevanje, kot tudi za samo delo.

Anže Luzar (63170183)

Tomaž Nemanič (63170212)