

Crazy Driver

Termen de predare: 17 noiembrie 2013, ora 23:55.

Nota: **Orice informatie ce nu a fost acoperita in acest document este la latitudinea voastra!**

1. Descriere cerinte

Utilizand framework-ul oferit la laborator (cel din laboratorul 3 - fara shadere), trebuie sa implementati un joc 3D wireframe de curse auto cu obstacole, intr-o varianta simplificata. Urmatoarele exemple pentru astfel de jocuri sunt complet orientative; puteti folosi ce culori si forme doriti, atat timp cat rezultatul final e in ton cu tema.

- [Exemplul 1](#)
- [Exemplul 2](#)
- [Exemplul 3](#)
- [Exemplul 4](#)

2. Jocul

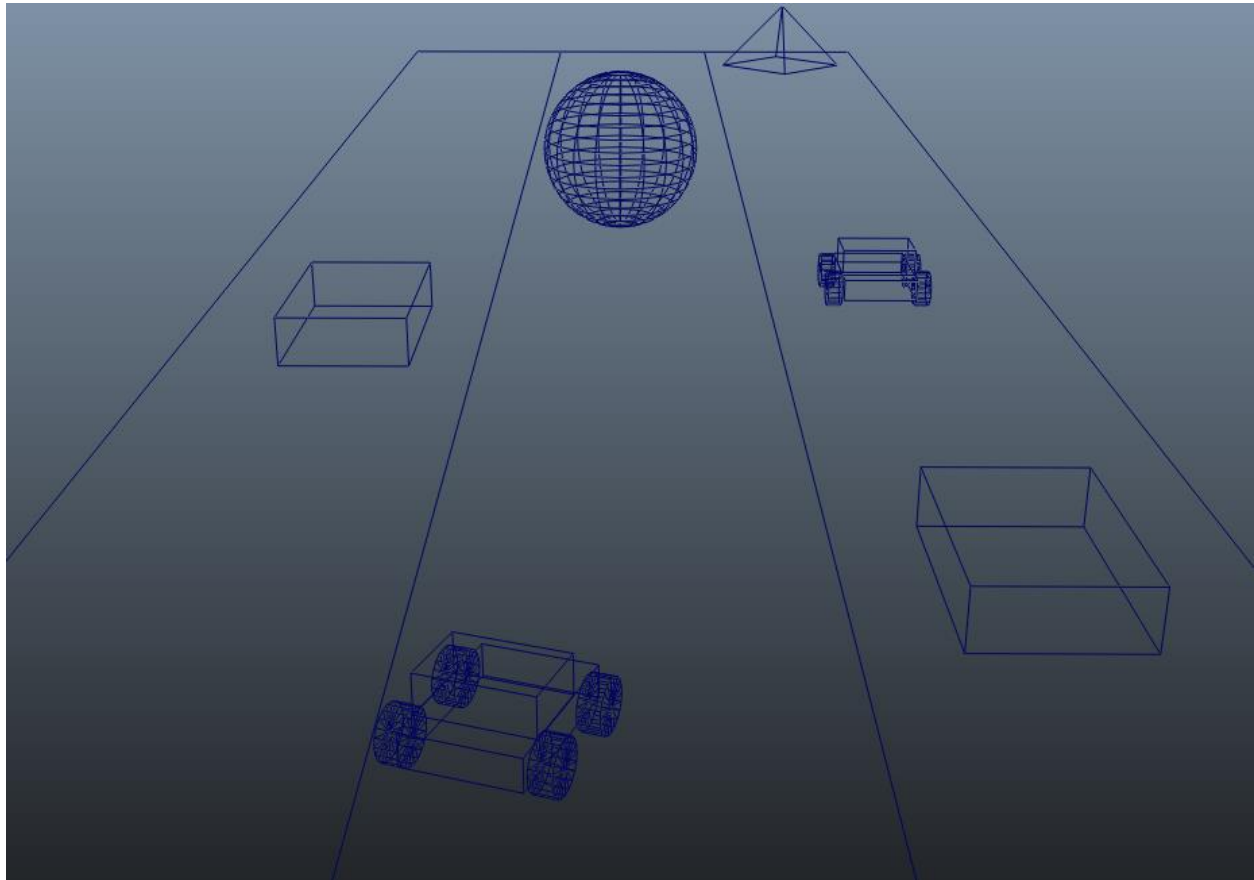
Terenul de joc este suprafata pe care se desfasoara jocul si este reprezentata printr-o sosea dreapta cu mai multe benzi (cu alte cuvinte, nu veti avea de implementat curbe pe aceasta sosea). Separarea benzilor va trebui sa fie marcata, de exemplu prin linii intrerupte.

Jucatorul este reprezentat in acest joc printr-o masina foarte simpla. Aceasta va fi alcatuita din cel putin un paralelipiped si trebuie sa fie usor identificabila fata de celelalte elemente ale jocului. Masina se va conduce prin intermediul tastelor *Sus*, *Jos*, *Dreapta*, *Stanga*. Efectul vizual al schimbarii unei benzi prin intermediul tastelor *Dreapta* sau *Stanga* ar trebui sa fie similar cu cel din *Exemplul 1* de mai sus – botul masinii trebuie sa se roteasca. Tastele *Sus* si *Jos* vor fi folosite pentru accelerare, respectiv franare.

Obstacolele sunt reprezentate de diverse entitati pozitionate pe benzile de circulatie : obiecte fixe (simulate prin forme geometrice simple precum paralelipede, piramide, sfere, conuri, cilindri etc.) sau mobile (alte masini, mergand fiecare cu o viteza constanta proprie). Obiectele mobile (masinile) nu isi vor schimba niciodata banda de circulatie si nici viteza. Generarea obstacolelor este la latitudinea voastra, o solutie putand fi un algoritm de generare aleator. Un obiect asignat unei benzi nu trebuie sa depaseasca in latime banda (cu alte cuvinte, nu va trebui sa considerati cazul in care un obiect ocupa simultan mai multe benzi). De asemenea, este tot la latitudinea voastra gestionarea situatiei in care un obstacol mobil se intalneste cu un obstacol fix de pe aceeaasi banda.

Figura de mai jos prezinta o schita orientativa a cum ar putea arata jocul (realizata in Autodesk Maya – fundalul nu trebuie sa fie gradient). Masina nu trebuie sa fie atat de complexa, iar diferitele obiecte ce compun masina (primitive 3D sau poligoane) pot fi generate folosind

framework-ul de laborator. Aceeasi observatie se aplica si obstacolelor. Astfel, puteti realiza jocul numai din paralelipipede daca doriti, dar pentru punctaj maxim se recomanda construirea a cel putin unui alt tip de primitiva, ca de ex. o sfera.



Scopul jocului este de a parcurge o anumita distanta pe sosea, acumuland puncte prin evitarea obstacolelor, respectiv pierzand puncte pentru obiectele cu care jucatorul intra in coliziune. Va fi permis un numar maxim de coliziuni, dupa care jocul va fi pierdut. Pentru testarea mai usoara a coliziunii, puteti folosi volume incadratoare (*bounding spheres* sau *axis-aligned bounding boxes*) pentru jucator si obstacole. Numarul de puncte acordate / luate per obstacol si numarul maxim de coliziuni permise sunt la latitudinea voastra.

Scorul va fi afisat permanent in partea de sus a ecranului si va cuprinde distanta ramasa, timpul petrecut pana la momentul respectiv, punctajul curent si numarul de coliziuni ramase pana la pierderea jocului.

3. Bonusuri

Se pot obtine bonusuri pentru implementarea unor elemente suplimentare (nu neaparat toate):

- aparitia ocazionala a unor obstacole-bonus (power-ups - coliziunea cu ele avand efecte benefice, precum obtinerea unor puncte suplimentare)
- modele geometrice mai complexe pentru jucator si celelalte masini

- generarea dinamica a soselei (eventual chiar si cu curbe si/sau dealuri) - aici se pleaca de la observatia ca in orice moment din timpul jocului, jucatorul vede o portune foarte mica din lungimea totala a soselei, insa soseaua se deseneaza complet la fiecare cadru ; acest algoritm nu este optim, o alternativa mult mai buna (cea propusa in acest bonus) este de a considera drumul format din mai multe portiuni de lungime fixa numite « tile-uri » (spre exemplu, un drum de 10 km poate fi considerat ca format din 10 portuni identice a cate 1 km fiecare) ; la fiecare cadru, se va desena doar tile-ul curent si, eventual, cel urmator. In acest mod, efectul vizual va fi acelasi insa efortul de desenare va fi redus.
- diferite animatii si efecte aplicate masinii jucatorului si/sau obstacolelor
- implementarea unui sistem de consum de carburant (consumul va fi mic la viteza constanta si franare, dar va fi mare la acceleratie) ; jucatorul va porni cu o anumita cantitate de carburant si va trebui sa termine traseul cu aceasta, in caz contrar pierzand jocul
- poligoane pline si/sau shading – in cel de-al doilea caz obiectele din joc fiind iluminate fie folosind pipeline-ul fix OpenGL (in conjunctie cu primul framework de laborator), fie folosind shadere (framework-ul introdus in laboratorul 4) ; in primul caz trebuie avut grija ca formele in perspectiva sa fie inca usor de recunoscut.

Punctajele de la bonusuri vor fi determinate de asistentul ce va verifica tema in functie de complexitatea si calitatea acestora.

Succes!

Responsabili: Andrei Trandafir, Mihai Francu