Elemente de Grafică pe Calculator - Tema 3

NICULA Bogdan Daniel 334CB Atletism

1. Cerinta

Implementati o cursa de atletism cu cel putin 3 culoare, un obiect controlat de user si 2 adversari.

Pentru fiecare "concurent" va exista cate o camera FPS. De asemenea vom avea o camera TPS care sa urmareasca player-ul si o camera FPS libera.

2. Utilizare

2.1 Fisiere

Se foloseste fisierul speed.txt.

In el se introduc viteza player-ului si vitezele celor 2 adversari cate una pe linie in ordinea(viteza_player, viteza_adv_1, viteza_adv_2). Pentru viteze mai mici de 0 se va folosi o valoare prestabilita(1).

Se recomanda viteze intre 0.5 si 3; pentru valori mai mari durata jocului va fi prea scurta pentru a observa ceva.

2.2 Consola

-

2.3 Input Tastatura

Taste:

- [UP ARROW | DOWN ARROW | LEFT ARROW | RIGHT ARROW]
 - o Permit deplasarea inainte, inapoi, stanga, dreapta a player-ului
- [P]
 - o Permite stoparea temporare a jocului.
- [Y|U]
 - Permit modificarea camerelor(6 optiuni posibile)
- [l]
 - Afisare wireframe(nu e recomandat...)

- [F1|F2]
 - o toggle FullScreen/NormalScreen

Comenzi in cazul camerei 3(camera FPS cu imaginea de ansamblu a stadionului)

- [W|S; A|D; R|F]
 - o translatare upword; right; forward
- [Q|E]
 - o rotire Oy
- [Z|C]
 - o rotire Oz
- [T|G]
 - rotire Ox
- [O]
 - o resetare camera FPS libera

3. Implementare

IDE: Microsoft Visual Studio 2012

Sistem de operare: Microsoft Windows 7

Framework: Cel oferit la laborator(laboratoarul 6 - 3 shadere).

->viteze concurenti

Vitezele concurentilor se citesc din speed.txt, unde se gasesc scrise cate una pe linie. Valorile recomandate sunt intre 0.5 si 3. In cazul in care introduceti valori <=0 ele vor fi setate implicit 1. Vitezele celor 2 adversari nu se pot modifica pe parcurs. Viteza jucatorului este constanta, insa poate scadea daca acesta iese de pe pista(teren verde).

->reprezentare:

-concurenti: Acestia sunt "desenati" prin intermediul functiei **void drawCompetitor(...).** Toti sunt reprezentati ca un "android" in culorile rosu(player), mov(adv_1), galben(adv_2) folosindu-se mesha "android logo.obj" din fisierul resurse. (NOTA: Se pot folosi, la alegere si meshe-le songoku, van, buddha sau bunny aflate in ./resources)

-pista: Pista este alcatuita din 3 benzi de alergare si un teren verde care o inconjoara. "Descrierea" ei se afla in fisierul "track.obj".

Afisarea pistei se face cu functia drawTrack(...).

NOTA: Deoarece pista este alcatuita din sute de triunghiuri mici, la rotatii FPS ale camerei libere este posibil ca ea sa para segmentata.

-steag: (./resurse/flag.obj) Steagul este alcatuit dintr-o retea de 200 de puncte dispuse pe 10 linii si 20 de coloane. Afisarea sa se realizeaza in cadrul functiei drawFlag(...). In cadrul shader-ului de vertecsi avem un caz special gandit pentru steag(flag_mode = 1) in care fiecare punct al steagului este translatat pe Oz pe baza unei functii sinusoidale care tine cont de coordonatele Ox si Oy ale punctului.

-camera laterala: (./resurse/camera.obj) Camera laterala este afisata in permanent cu directia obiectivului tangenta la conturul pistei.

->limitare miscare:

Avem 2 tipuri: limitare partiala si limitare totala. La iesirea de pe pista jucatorul este incetinit(se injumatateste viteza), dar el se poate deplasa. Totusi inaintarea sa este oprita la o departare mai mare de 20 de unitati de pista.

Acest comportament este gestionat de functiile: fieldContains(...) si totalyOut(...).

->camere

In total avem 6 camere: cate o camera FPS pentru fiecare concurent, o camera TPS laterala pentru player, camera FPS panoramica libera, si o camera TPS in spatele player-ului. Camerele se pot schimba prin intermediul tastelor Y si U. In cazul camerei FPS libere, aceasta poate fi mutata cu tastele: W,A,S,D,R,F si rotita cu Z, C, T, G, Q, E.

Caz special: Camera TPS Laterala

Aceasta este permanent reprezentata printr-o camera foto profesionista. Ea urmareste mereu player-ul. Deplasarea ei se bazeaza pe deplasarea pe Oz a player-ului.

Pe portiunile drepte camera se deplaseaza exact cat se deplaseaza player-ul pe Oz. Pe portiunile curbate ea se roteste in jurul centrului curbei(semicercului) astfel incat centrul playerului, centrul camerei si centrul semicercului sa fie permanent coliniare.

Pentru a descrie aceasta ne folosim de Teorema lui Thales:

coord_Camera = coord_Player * distanta(centru, Camera) / distanta(centru, Player);

Acest comportament este descris in functia: moveSideCamera(float deplasareOz);

->Miscare adversari

Pe portiunea dreapta adversarii se deplaseaza cu o viteza constanta citita din fisier. Pe portiunea circulara ei se rotesc cu k * speed la fiecare iteratie, unde k este o constanta determinata empiric care depinde de departarea fata de centrul semicercului. La iesirea din curba se efectueaza o mica corectie asupra pozitiei pe Ox a adversabilului insesizabila la viteze normale.(<10)

->Shadere

vertex_shader

Am specificat un comportament special(data de flag-ul: flag_mode)pentru reprezentarea steagului. Coordonatele Oz ale steagului vor depinde de o functie sinusoidala ce primeste ca parametri coordonatele pe Ox si Oy ale punctului si un unghi incrementat la fiecare iteratie. Aceasta functie va crea un efect de "fluturare" a steagului.

fragment_shader

Avem 2 comportamente posibile intre care alegem pe baza flag-ului value. In primul caz (value!=1) colorarea se face in functie de vectorul normala, in al doilea caz(value==1) ea se face in procent de 80% in functie de o culoare primita din exterior si 20% in functie de vectorul normala.

3.1 Despre functionarea programului

>la pornire

Playerul apare pe banda stanga, iar adversarii sai pe celelalte 2 benzi.

> functionare pe parcurs

Ciocnirile nu au niciun efect asupra sa. Scopul sau este sa faca tura cat mai repede. Jocul poate fi oricand pauzat prin tasta 'P'. Ceilalti playeri se deplaseaza cu viteza constanta si fara schimbarea benzii

>conditii incheiere

O data ce un jucator a realizat o tura completa jocul s-a incheiat si se trece la scena victoriei. In aceasta jucatorul invingator este afisat impreuna cu drapelul sau(in culoarea proprie).

4. Testare

Jocul a fost testat cu viteze diferite(se recomanda valori intre 0.5 si 3), cu castigatori diferiti.

Testarea sa a fost facuta pe Windows 7 si s-au incercat toate cazurile speciale.

5. Probleme aparute

La rotirea camerei FPS panoramice libere este posibil ca pista sa aiba un aspect sters/fragmentat deoarece ea este reprezentata bidimensional si este compusa din foarte multe triunghiuri de dimensiuni mici.

6. Continutul Arhivei

main.cpp

Sursa principala a aplicatiei. Aici este descris tot comportamentul jocului.

"./shadere/": shader_vertex.glsl shader_fragment.glsl

Shadere folosite pentru "desenarea" obiectelor din joc.

"./resurse/" track.obj, camera.obj, flag.obj, /android/android logo.obj

Meshe folosite pentru a reprezenta pista, camera steagul si competitorii.

Meshe-le track.obj si flag.obj au fost generate de mine, restul au fost gasite pe internet.

speed.txt

Fisier in care sunt reprezentate, pe cate o linie, vitezele jucatorilor.

README

acest fisier

7. Functionalitati

Programul contine un set de baza de functionalitati cerute in enuntul temei(existenta celor 5 camere, formatul jocului, scena de victorie) la care am adaugat si niste functionalitati pentru a imbunatatii experienta jucatorului.

Lista functionalitati

Functionalitati Standard (ca in enunt)

- >Miscare player(smooth) intre benzi
- >Camere FPS competitori, TPS pentru player si FPS libera
- >Reprezentarea camerei laterale printr-un obiect
- >Formatul jocului: castiga "alergatorul" cu cea mai rapida tura
- >Scena finala cu afisarea steagului jucatorului castigator.
- >Citirea vitezelor jucatorilor din fisierul speed.txt

Functionalitati Suplimentare

Camera TPS in spatele jucatorului(mai realista).

Colectie de meshe in folderul resurse din care se pot alege pentru joc.

Scena realista.

Optiunea de oprire temporara a jocului.

Scena finala grandioasa.