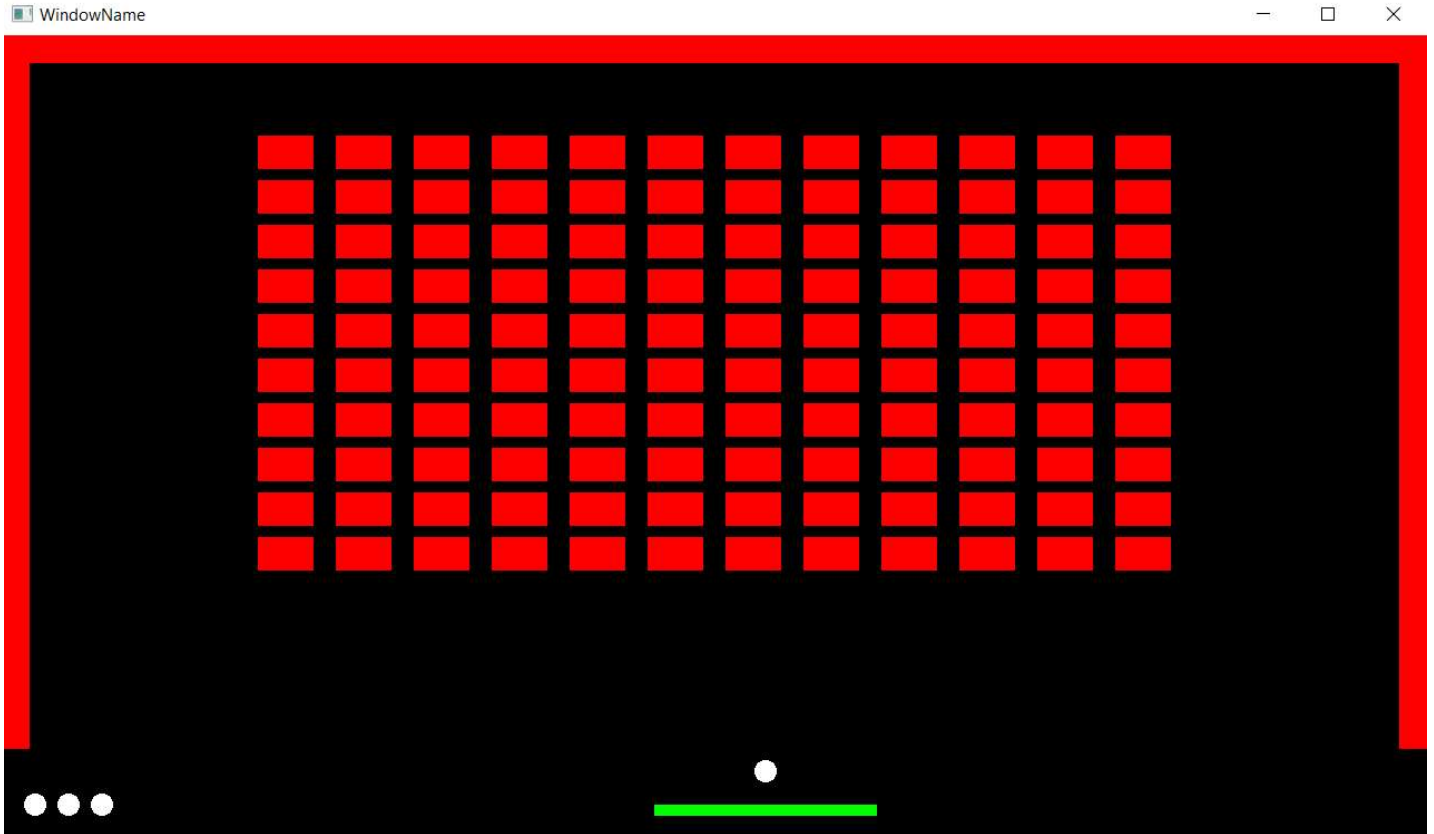


Tema 1

Brick breaker

Scopul temei este realizarea unui joc 2D in care utilizatorul controleaza o platforma. Platforma ghideaza o bila care loveste caramizi.



Obiecte

Scena contine urmatoarele obiecte:

- platforma (un dreptunghi umplut)
- peretii: peretele din stanga, din dreapta si de sus (dreptunghiuri umplute)
- bila (un disc de cerc)
- cel putin 5 x 10 caramizi (dreptunghiuri umplute)
- numarul de vietile ale jucatorului (de exemplu, fiecare viata poate fi reprezentata printr-un disc de cerc, asemanator cu bila)
- powerups (patrate)

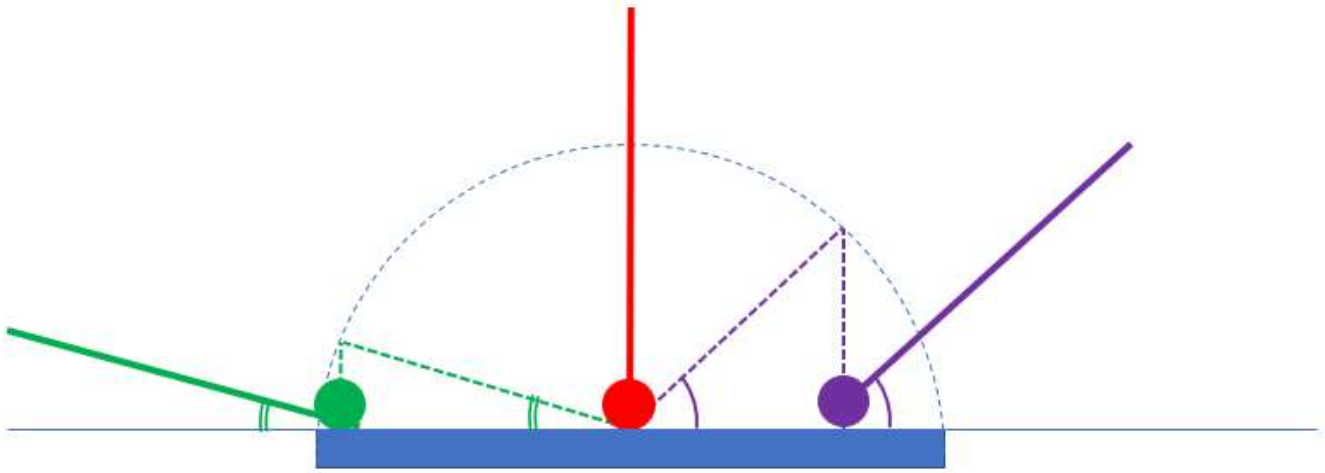
Gameplay

Scopul jocului este distrugerea tuturor caramizilor din scena.

Platforma este controlata de jucator prin deplasarea mouse-ului. La mouse click, bila este trimisa in scena. Bila se misca incontinuu pe directia data de platforma sau prin reflexie, la coliziunea cu peretii si caramizile. Daca bila ajunge in partea de jos a scenei si nu este salvata de platforma, jucatorul pierde o viata (iar numarul de obiecte de pe ecran reprezentand vietile se micsoreaza).

La lansarea bilei in scena, ea este trimisa de-a lungul directiei verticale. La coliziunea cu platforma, i se schimba directia. Daca bila loveste platforma in mijloc, directia de reflexie va fi verticala. Daca bila loveste platforma in partea stanga, atunci directia de reflexie pentru bila va fi inspre stanga platformei. Analog in partea dreapta. In figura de mai jos este explicata reflexia pe platforma pentru bila. Puteti sa va imaginati ca platforma reprezinta axa cosinusului de pe cercul trigonometric iar pozitia unde bila loveste platforma, chiar valoarea cosinusului. Daca bila cade pe mijlocul platformei, atunci $\cosinus = 0$. Daca bila cade in partea

dreapta a platformei, cosinus este între 1 și 0. Dacă bila cade în partea stângă a platformei, cosinus este între 0 și -1. Bila se reflectă după un unghi egal cu arccosinus.



La coliziune cu celelalte elemente din scena (pereti, caramizi), bila se reflectă natural (de exemplu, la coliziune cu un perete vertical, dacă x și y creșteau până în momentul coliziunii cu pași tx , respectiv ty , după coliziune y crește în continuare cu pasul ty , dar x începe să scadă cu pasul tx).

La coliziune cu bila, caramizile dispar treptat (li se aplică o scalare pentru micșorare timp de 1-2 secunde).

La coliziune cu unele caramizi, apar în scena powerups (patrate) care se rotesc în timp ce cad pe verticală. Dacă aceste powerups sunt atinse de platformă, atunci sunt activate anumite bonusuri pentru jucător pentru o anumită perioadă de timp (30 secunde - 1 minut). Exemple de powerups:

- apare și în partea de jos a scenei, sub platformă, un perete; astfel jucătorul nu își pierde o viață dacă bila nu este salvată de platformă
- platforma poate trage cu anumite arme înspre caramizi
- bila poate deveni mai mare
- bila poate deveni mai puternică. Astfel, la coliziunea cu caramizile, bila nu mai realizează reflexie, ci își continuă traseul, distrugând mult mai multe caramizi printr-o singură traversare a scenei
- platforma devine "lipicioasă". Astfel, la coliziunea cu platforma, bila rămâne pe platformă până când este lansată din nou în scenă. Bila ar putea fi trimisă pe o anumită direcție, dată de mouse. Având în vedere că în funcționalitatea de bază, platforma se mișcă după mouse, acest bonus ar putea fi implementat prin două stări: În starea 1 bila se lipește de platformă la poziția din care a venit și platforma se deplasează liberă până la momentul în care se apasă click stângă. Starea 2 durează de la momentul apăsării click-ului până la momentul release-ului. La momentul primei apăsări platforma se blochează la poziția în care se afla. În timpul hold-ului mouse-ul se mișcă liber fără să afecteze platforma, iar la release bila pleacă în direcția poziției mouse-ului.

Funcționalități obligatorii

Barem orientativ pentru realizarea funcționalităților (din 100 puncte):

- desenare scenă (25 puncte)
- control platformă cu mouse-ul (5 puncte)
- detectie coliziuni bila-platformă, bila-pereti, platformă-powerups (5 puncte)
- detectie coliziuni bila-caramizi (10 puncte)
- deplasare bila în scenă la lansare (5 puncte)
- deplasare prin reflexie la coliziunea cu platformă (10 puncte)
- deplasare prin reflexie la coliziunea cu peretii și caramizile (15 puncte)
- animație dispariție caramizi și animație rotire și cadere powerups (10 puncte)
- gestiune pierdere vieți (5 puncte)
- implementarea unui powerup la alegere (10 puncte)

Bonusuri

Pentru bonusuri se accepta orice aduce realism scenei sau creste imersiunea jocului. Exemple de bonusuri:

- implementarea altor powerups in afara de cea din functionalitatile obligatorii (fie powerups descrie mai sus, fie altele propuse de voi)
- realizarea mai multor niveluri care cresc in dificultate (de exemplu, inserarea in scena a unor caramizi mai puternice, care nu se distrug la o singura coliziune cu bila)

Demo joc

Aici aveti un demo (implementat in framework-ul de laborator) pentru tema, care va va ajuta in intelegerea enuntului temei.

Intrebari si raspunsuri

Pentru intrebari vom folosi forumurile de pe moodle.

Notare

Baremul este orientativ. Fiecare asistent are o anumita libertate in evaluarea temelor (de exemplu, sa dea punctaj partial pentru implementarea incompleta a unei functionalitati sau sa scada pentru hard coding). Acelasi lucru este valabil atat pentru functionalitatile obligatorii, cat si pentru bonusuri.

Tema trebuie incarcata pe moodle. Pentru a fi punctata, tema trebuie prezentata la laborator. Vor exista laboratoare speciale de prezentare a temelor (care vor fi anuntate).

Indicatii suplimentare

Tema va fi implementata in OpenGL si C++. Este indicat sa folositi framework-ul si Visual Studio.

Pentru implementarea temei, in folderul Source/Laboratoare/ puteti crea un nou folder, de exemplu Tema1, cu fisierele Tema1.cpp si Tema1.h (pentru implementare POO, este indicat sa aveti si alte fisiere). Pentru a vedea fisierele nou create in Visual Studio in Solution Explorer, apasati click dreapta pe filtrul Laboratoare si selectati Add→New Filter. Dupa ce creati un nou filtru, de exemplu Tema1, dati click dreapta si selectati Add→Existing Item. Astfel adaugati toate fisierele din folderul nou creat. In fisierul LabList.h trebuie adaugata si calea catre header-ul temei. De exemplu: #include <Laboratoare/Tema1/Tema1.h>

Arhivarea proiectului

- in mod normal arhiva trebuie sa contina toate resursele necesare compilarii si rularii
- inainte de a face arhiva asigurati-va ca ati dat clean la proiect
 - click dreapta pe proiect in **Solution Explorer** → **Clean Solution**, sau
 - stergeti folderul **/Visual Studio/obj**
- stergeti fisierul **/Visual Studio/Framework EGC.sdf** (in caz ca exista)
- stergeti fisierul **/Visual Studio/Framework EGC.VC.db** (in caz ca exista)
- stergeti folderul **/x64** sau **/x86** (in caz ca exista)
 - executabilul final este generat in folderul **/x86** sau **/x64** la finalul link-editarii in functie de arhitectura aleasa la compilare (32/64 biti)
- in cazul in care arhiva tot depaseste limita de 20MB (nu ar trebui), puteti sa stergeti si folderul **/libs** sau **/Resources** intrucat se pot adauga la testare. Nu este recomandat sa faceti acest lucru intrucat ingreuneaza mult testarea in cazul in care versiunea curenta a librariilor/resurselor difera de versiunea utilizata la momentul scrierii temei.

Deadline tema

4 noiembrie ora 23:55