

## Tema 2 - World of Tanks

---

- **Responsabili:** Anca Băluțoiu, Robert Caragicu, Florin Iancu, Vlad Novetschi
- **Lansare:** 20 noiembrie 2023
- **Termen de predare:** 15 decembrie 2023, ora 23:59
- **Regulament:** Regulament general
- **Notă:** Orice informație ce nu a fost acoperită în acest document este la latitudinea voastră!

În cadrul temei 2, veți avea de implementat un joc similar cu World of Tanks, însă single player.

### Reguli generale de joc

Jocul se desfășoară pe un plan orizontal ce conține tancuri și clădiri. Scopul jocului este ca tancul jucătorului să distrugă cât mai multe tancuri inamice într-un timp limită (cel puțin 1 minut). Pentru fiecare tanc distrus jucătorul primește puncte. Tancurile se deplasează înainte, înapoi și își pot roti corpul. Pentru a ținti după inamici turela tancului poate fi rotită (cerință avansată). Proiectilele sunt lansate din vârful tunului și se deplasează rectiliniu și uniform pentru o durată de timp până când lovesc clădiri sau tancuri. Fiecare tanc rezistă la un număr de minim 3 lovituri de proiectil. Cu cât sunt mai avariate tancurile cu atât vor fi desenate mai deformate și mai înnegrite. Odată scurs timpul de joc, scena va ingheța; doar rotația turelei și a tunului vor mai fi posibile.

Tancurile inamice se vor mișca pe hartă fie pe un traseu predefinit (cerință de bază) fie pe un traseu aleator (cerință avansată). Tancurile inamice își rotesc turela în direcția jucătorului în scopul de a trage proiectile înspre acesta.

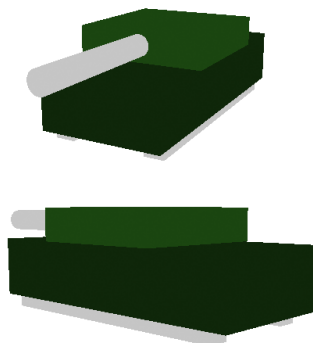
Un tanc complet avariat este imobil.

Jocul se termină fie când limita de timp este atinsă, fie când tancul jucătorului este distrus. În cazul în care timpul a expirat, în consolă se va afișa scorul. Scorul acumulat se consideră doar dacă timpul de joc a expirat. Dacă tancul jucătorului este distrus, atunci se afișează în consolă un mesaj corespunzător.

### Detalii de implementare

#### Construcție elemente vizuale

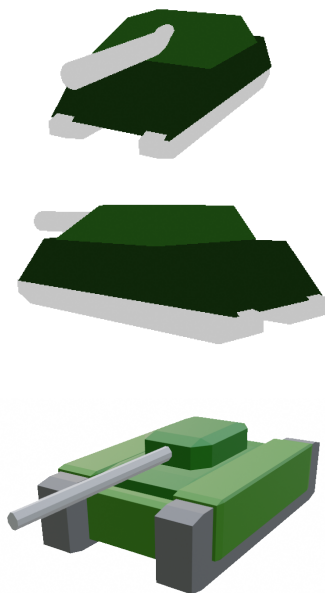
Tanc de bază



Un tanc este format dintr-un corp, turelă, tun și șenile. Corpul este paralelipipedul cel mare. Atașat de corp este turela formată tot dintr-un paralelipiped. Turela este atașată de corp și se rotește stânga dreapta după

metoda descrisă în capitolul Rotație Turelă Tanc. Atașat de turelă este un tun format dintr-un cilindru. Atașat de corp se găsesc două paralelipipede care reprezintă șenilele tancului. Tancul va fi desenat în 3 culori pentru a distinge elementele sale. Astfel corpul vor avea o culoare, turela altă culoare iar tunul și șenilele altă culoare. Tunul și șenilele au aceeași culoare.

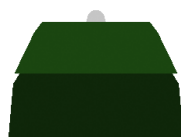
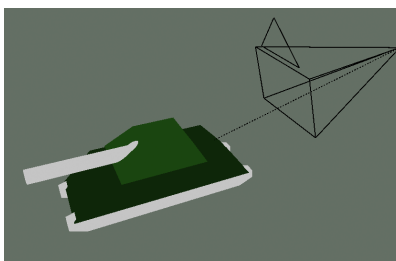
Tanc avansat



O cerință avansată este ca tancul să aibă forme mai complexe. Aveți 2 exemple date în imaginile de mai sus. Puteți importa modele create prin programe software cât timp se păstrează separate componentele tancului (corpul, turela, tunul și șenilele). Aceste componente vor fi desenate cu aceeași culori ca în capitolul *Tanc de bază*.

**Dacă rezolvați această cerință se consideră rezolvată și cerința cu tancul de bază.**

Poziționare Cameră Third Person

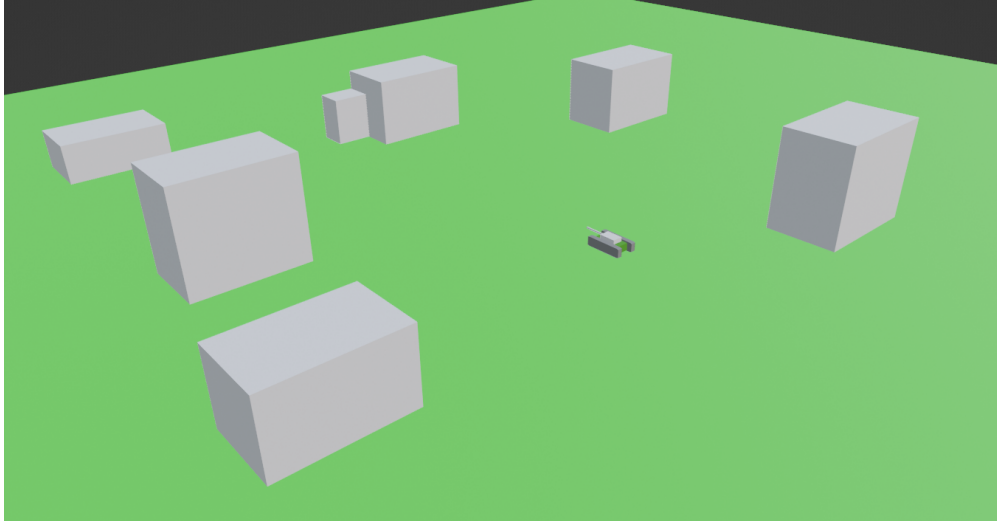


Camera va fi poziționată în spatele tancului așa cum apare în figura de sus (camera este ilustrată cu piramida neagră). Mai exact, camera este atașată de corpul tancului la o anumită distanță și se uită înainte. Astfel camera se deplasează după cum se deplasează corpul tancului. Când corpul tancului se rotește camera va orbita în jurul corpului pentru a fi în continuare în spatele camerei. Vederea camerei este în figura de jos.

Mediul de joc

Mediul de joc este format dintr-o suprafață plană ce reprezintă pământul.

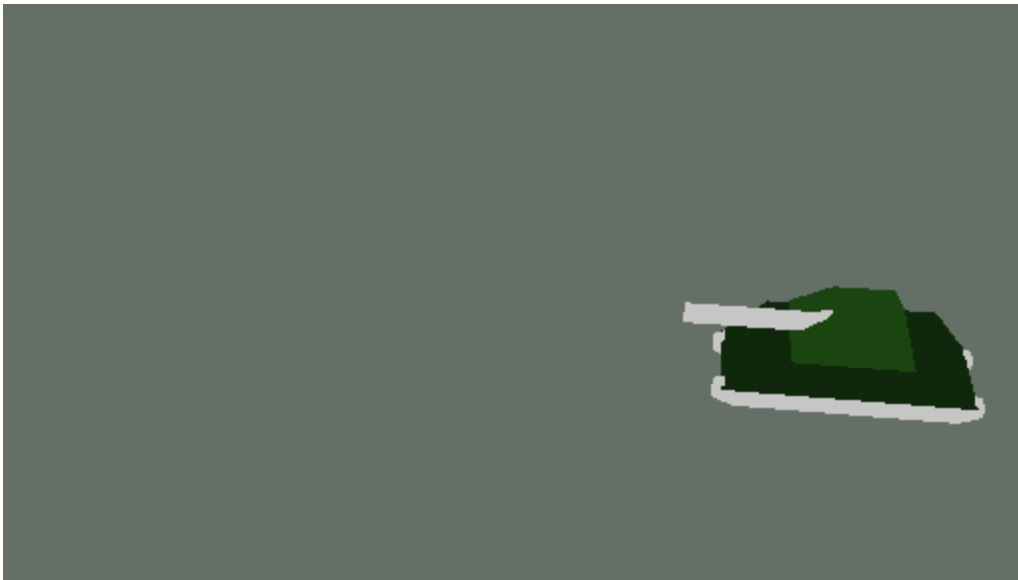
Clădiri (cerință avansată)



Pe suprafața de joc, vor fi amplasate aleator clădiri reprezentate prin paralelipede aliniate cu axele. Clădirile se pot intersecta. Atunci când vor fi create tancul jucătorului și tancurile inamice acestea nu trebuie să se intersecteze cu clădirile. Clădirile vor fi colorate diferit de restul elementelor pentru a fi distinse.

Controlul tancului

Mișcare tanc



Pentru a mișca tancul jucătorul controlează mișcarea relativ la poziția și rotația tancului (în loc de mișcare față de perspectiva camerei). Acest mod de control se numește "controluri de tanc" [[https://en.wikipedia.org/wiki/Tank\\_controls](https://en.wikipedia.org/wiki/Tank_controls)]. 😊

Astfel:

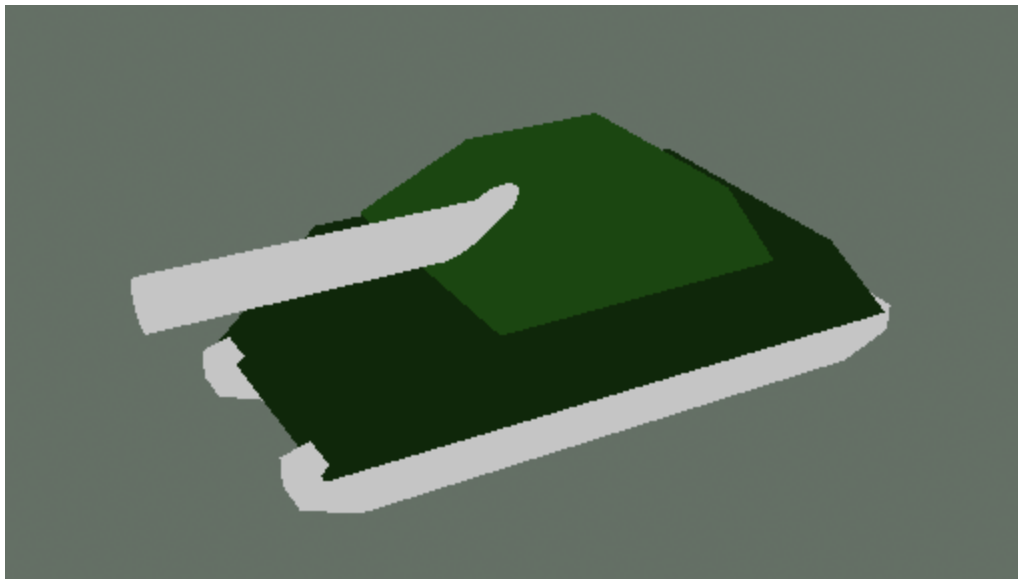
- Cât timp este apăsată tasta W tancul se va deplasa în direcția indicată de corpul său.
- Cât timp este apăsată tasta S tancul se va deplasa în direcția opusă față de cea indicată de corpul său.
- Cât timp este apăsată tasta A tancul se rotește la stânga.
- Cât timp este apăsată tasta D tancul se rotește la dreapta.

Vitezele de deplasare și rotație sunt la alegerea voastră.

### Rotație cameră (cerință avansată)

Camera poate fi rotită folosind click dreapta și mișcând mouse-ul similar cu rotațiile third person din laboratorul 5. Atunci când nu este apăsat click dreapta mișcările mouse-ului vor conduce la rotația turelei descrisă în capitolul *Rotație Turelă Tanc*.

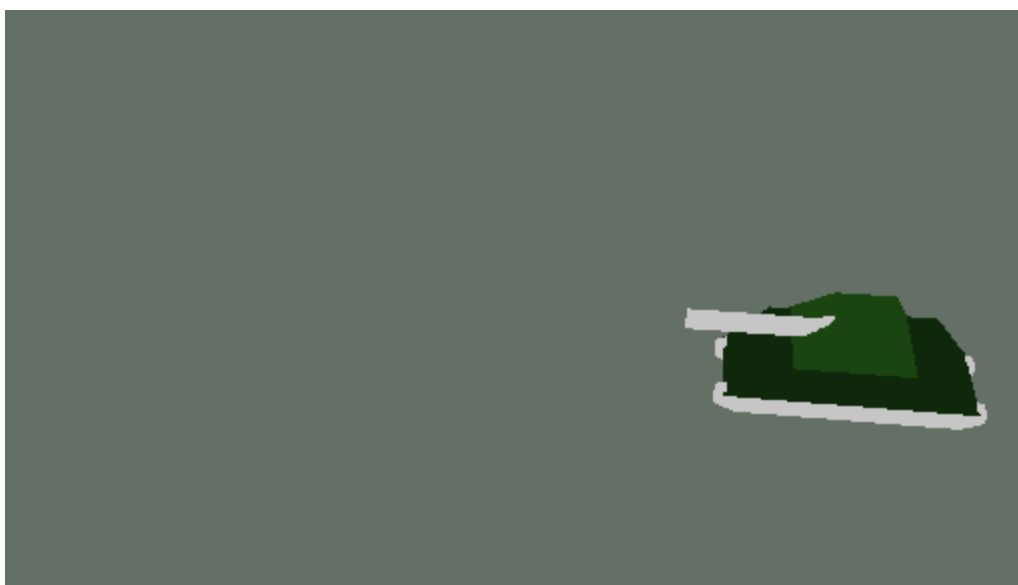
### Rotație Turelă tanc (cerință avansată)



Turela se va roti stânga dreapta prin mișcarea mouse-ului în stânga și în dreapta. Rotația este în jurul centrului propriu al turelei pe planul local xoz.

**Atenție! Poziția camerei nu se modifică când se rotește turela pentru că aceasta este atașată de corp.**

### Tras cu tunul (cerință de bază)



Tancul trage cu proiectile atunci când utilizatorul dă click stânga. Proiectilele sunt sfere negre, încep din vârful tunului și au mișcare rectilinie și uniformă după cum este prezentat în animație. Direcția de deplasare este în direcția indicată de tunul tancului când s-a tras proiectilul. Proiectilul va fi șters atunci când se întâmplă cel puțin una dintre situații:

- Trece un anumit timp după tragere
- Proiectilul lovește tancuri

- Proiectilul lovește clădiri

Viteza inițială a proiectilului și timpul de viață al proiectilului sunt la alegerea voastră. Fiecare tanc nu poate trage proiectile la un interval mai mic de 1 secundă.

## Inamicii

Tancurile inamice vor avea alte culori decat cel al jucătorului.

Mișcarea inamicilor

### Cerință de bază

Tancurile se vor mișca înainte și înapoi pe un traseu predefinit (de-a lungul unei linii, într-un cerc etc).

### Cerință avansată

Inamicii vor fi amplasati aleator pe harta (fie pe toata suprafata, fie într-o anumită zona de început - la libera alegere), si se vor plimba aleator.

Puteti face orice fel de miscare doriti, atat timp cat aceasta este naturală și continuă (inamicii nu se teleporteaza, nu se invart instantaneu etc)

O posibila astfel de implementare este printr-un automat de stări rudimentar. Mai precis, tancul poate sa fie în diverse stări (mișcare înainte sau înapoi, rotire pe loc stanga sau dreapta) și la fiecare cateva secunde, tancul își va schimba starea (de exemplu, va merge inainte 4 secunde, apoi se va roti spre stânga timp de 2 secunde, va începe iar să meargă înainte cateva secunde). Un exemplu de cod este disponibil aici: [C++] gcc 12.2.0 - Wandbox [<https://wandbox.org/permlink/y9DkfVMLZrhFhg4z>]

Cum atacă Inamicii? (cerință avansată)

Atunci cand jucătorul este destul de aproape de un inamic, turela inamicului se va învârti către jucător si va trage spre acesta.

## Coliziuni

Pentru coliziuni obiectele vor fi approximate astfel:

- Tancurile cu sfere
- Proiectilele cu sfere
- Clădirile ca paralelipede aliniate cu axele

Mai multe informatii despre coliziuni si cum se pot implementa in 3D:

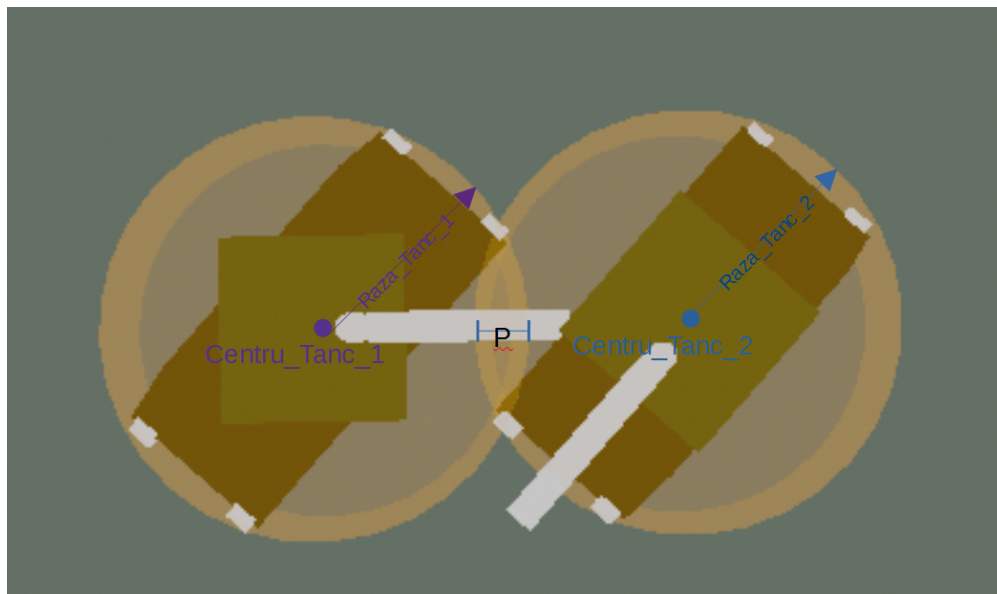
- [https://developer.mozilla.org/en-US/docs/Games/Techniques/3D\\_collision\\_detection](https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection)  
[[https://developer.mozilla.org/en-US/docs/Games/Techniques/3D\\_collision\\_detection](https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection)]

### Coliziunea Proiectil - Tanc

Atunci cand o coliziune dintre un tanc și un proiectil este detectata, proiectilul va disparea si tancului ii se va scadea HP-ul.

### Coliziunea Tanc - Tanc

Când 2 tancuri intră în coliziune, acestea vor fi deplasate astfel încât să nu mai fie în coliziune. Metoda de a calcula deplasarea necesară este următoarea:



Daca  $Distanța\_Intre\_Tancuri < Raza\_Tanc\_1 + Raza\_Tanc\_2$ , atunci acestea s-au ciocnit.

Si se suprapun cu o distanta egala cu  $|P| = (Raza\_Tanc\_1 + Raza\_Tanc\_2 - Distanța\_Intre\_Tancuri)$

pentru a evita suprapunerea lor, trebuie sa mutam tancurile în direcții opuse astfel încât distanța dintre acestea sa creasca cu  $|P|$ .

Directia in care trebuie sa le deplasam poate fi calculata astfel:

```
Dif = ( Centru_Tanc_2 - Centru_Tanc_1 )
P = |P| * glm::Normalize( Dif )
Centru_Tanc_1 += P * -0.5f;
Centru_Tanc_2 += P * 0.5f;
```

### Coliziunea Tanc - Clădire

Pentru coliziunea tanc clădire, va fi necesar un calcul similar cu cel prezentat la coliziunile tanc-tanc (calculat deplasamentul necesar unui tanc pentru a îl scoate din clădire).

## Efecte vizuale

### Deformare formă inamic

Obiectele ce reprezintă tancurile inamice (cu sau fără turelă, la libera alegere) se vor deforma în funcție de gradul de avarie al fiecăruia (HP); acesta va fi trimis către vertex shader, acolo unde se va realiza deformarea.

Deformarea va fi o funcție matematică (la libera alegere) ce va primi gradul de avarie (un număr) și coordonatele locale ale unui vertex (un vector), si va calcula un deplasament (un vector) care va fi aplicat coordonatelor locale ale respectivului vertex (înainte de aplicarea matricilor MVP).

Operația va fi deterministă, adică pentru aceeași pereche <grad de avarie, coordonate locale vertex> se va calcula întotdeauna același deplasament. Motivul este de a evita fluctuații nenaturale de la un cadru la altul.

### Modificare Culoare Inamic

Obiectele ce reprezintă tancurile inamice (cu sau fără turelă, la libera alegere) își vor schimba culoarea în funcție de gradul de avarie al fiecăruia. Schimbarea culorii se va realiza în fragment shader.

Schimbarea culorii va fi o funcție matematică (la libera alegere) ce va primi gradul de avarie (un număr) și culoarea inițială (un vector), și va calcula culoarea nouă.

Operația va fi deterministă, adică pentru aceeași pereche <grad de avarie, culoare inițială> se va calcula întotdeauna același deplasament. Motivul este de a evita fluctuații nenaturale de la un cadru la altul.

## Barem

### Funcționalități de Bază (150p)

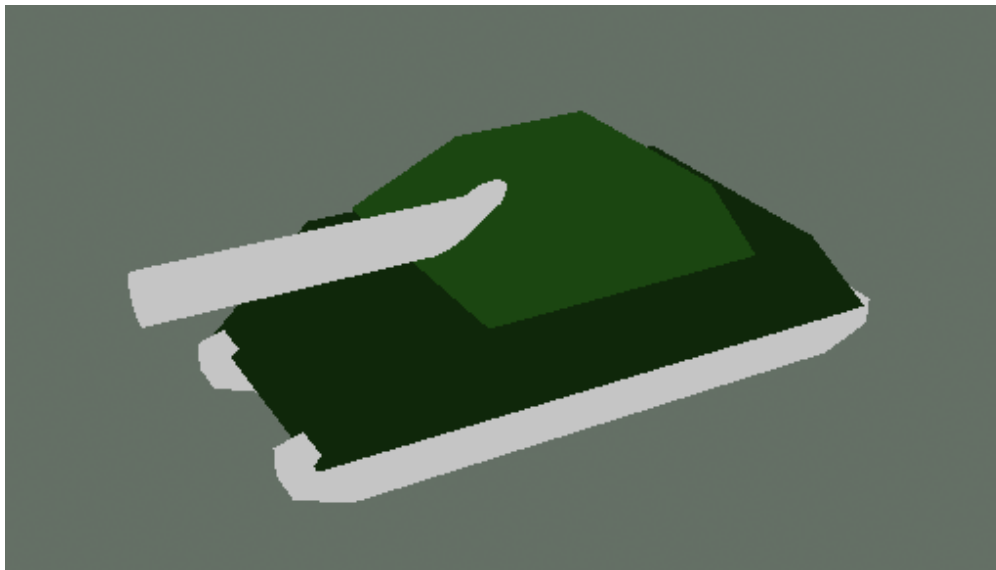
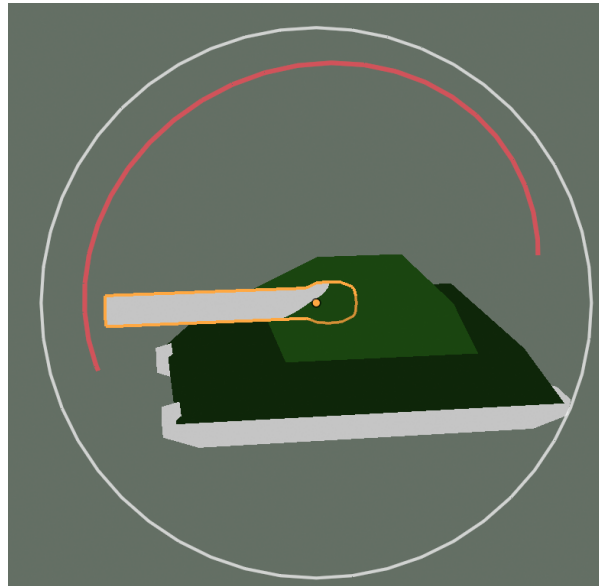
- Construcție tanc din forme simple 20p
- Control tanc 50p
  - Deplasare tanc prin tastele WASD 35p
  - Tras cu tunul 15p
- Dispariția proiectilului după un timp sau la coliziune cu un inamic 10p
- Mișcarea inamicilor în scenă pe o traiectorie definită 15p
- Sfârșitul jocului după cel puțin un minut 5p
- Coliziuni 25p
  - Proiectil - inamic 15p
  - Tanc - tanc 10p
- Avarie inamici 25p
  - Trimis către shader HP-ul tancului 5p
  - Deformare formă tanc în vertex shader în funcție de HP 10p
  - Modificare culoare tanc în fragment shader în funcție de HP 10p

### Funcționalități Avansate (75p)

- Construcție tanc din forme mai complexe 10p
- Rotația camerei 5p
- Rotație turelă tanc 15p
- Atacul jucătorului de către inamici 15p
- Mișcarea inamicilor în scenă pe o traiectorie aleatoare 10p
- Desenare clădiri 5p
- Coliziuni 15p
  - Tanc - clădire 10p
  - Proiectil - clădire 5p

## Exemple de Funcționalități Bonus

### Rotația tunului tancului



Tunul se va roti pentru a ținti în sus și jos prin mișcarea mouse-ului în sus și jos. Rotația este în jurul pivotului indicat de figură pe planul local zoy sau xoy (depinde care direcție este considerată cea din față). Pivotul este punctul galben și planul de rotație este cel ce conține cercul roșu. În animație sunt prezentate cum lucrează cele două rotații pentru ca tancul să poată ținti oriunde.

Traietorie parabolă a proiectilului





Proiectilul va avea mișcare balistică după cum este prezentat în animație. Se aplică legea gravitației. Astfel viteza inițială a proiectilului este direcția tunului înmulțită cu o constantă. În fiecare cadru algoritmul următor descrie mișcarea proiectilului:

- $viteza.y -= gravitație * deltaTime$
- $poziție += viteza * deltaTime$

În acest caz proiectilul nu va mai fi șters după o perioadă de timp. În schimb proiectilul va fi șters atunci când ajunge sub planul pământului ( $poziție.y < 0$ ) **Dacă se implementează mișcarea balistică se consideră rezolvată cerința de bază cu mișcarea rectilinie și uniformă.**

## Minimap

Va exista un viewport aditional, așezat în unul din colțurile ferestrei, care va conține o proiecție ortografică a hărții, văzută de sus. Jucătorul se va menține centrat în mijlocul viewport-ului.

## Alte exemple de bonusuri

- Tancurile inamice se deplasează înspre jucător atunci când îl atacă
- Meniuri de început joc și de sfârșit joc
- Scorul și mesajul pentru distrugerea propriului tanc sunt afișate în joc, nu în consolă
- Jucătorul poate trage mai multe tipuri de proiectile (ex: proiectile explozive care atunci cand lovesc ceva, explodeaza si afectează toate tancurile dintr-o anumita raza, chiar daca nu le-au nimerit in mod direct.)
- Power ups colectabile din scenă (de exemplu, invincibilitate pentru 5 secunde sau refacerea unei părți din HP)
- Pentru a îngreuna jocul, la un anumit interval de timp apar noi inamici noi, căzând din cer la poziții aleatoare
- Mai multe modele pentru tancurile inamice, cu comportamente diferite.
- Colizuni cu cutii care nu sunt aliniate cu axele
- Suprafata de joc sa fie non-plata (sa aveti poduri, relief, etc)

## Întrebări și Răspunsuri

Pentru întrebări vom folosi forumurile de pe moodle. Orice nu este menționat în temă este la latitudinea fiecărui student!

## Notare

Baremul este orientativ. Fiecare asistent are o anumită libertate în evaluarea temelor (de exemplu, să dea punctaj parțial pentru implementarea incompletă a unei funcționalități sau să scadă pentru hard coding). Același lucru este valabil atât pentru funcționalitățile obligatorii, cât și pentru bonusuri.

Tema trebuie încărcată pe moodle. Pentru a fi punctată, tema trebuie prezentată la laborator. Vor exista laboratoare speciale de prezentare a temelor (care vor fi anunțate).

## Indicații Suplimentare

Tema va fi implementată în OpenGL și C++. Este indicat să folosiți framework-ul și Visual Studio.

Pentru implementarea temei, în folderul **src/lab\_m1** puteți crea un nou folder, de exemplu **Tema2**, cu fișierele **Tema2.cpp** și **Tema2.h** (pentru implementare POO, este indicat să aveți și alte fișiere). Pentru a vedea fișierele nou create în Visual Studio în Solution Explorer, apăsați click dreapta pe filtrul **lab\_m1** și selectați **Add→New Filter**. După ce creați un nou filtru, de exemplu **Tema2**, dați click dreapta și selectați **Add→Existing Item**. Astfel adăugați toate fișierele din folderul nou creat. În fișierul **lab\_list.h** trebuie adăugată și calea către header-ul temei. De exemplu: **#include "lab\_m1/Tema2/Tema2.h"**

## Arhivarea Proiectului

- În mod normal arhiva trebuie să conțină toate resursele necesare compilării și rulării
- Înainte de a face arhiva asigurați-vă că ați curățat proiectul Visual Studio:
  - Click dreapta pe proiect în **Solution Explorer** → **Clean Solution**
  - Ștergeți folderul **/build/.vs** (dacă nu îl vedeți, **este posibil să fie ascuns**)
- În cazul în care arhiva tot depășește limita de 50MB (nu ar trebui), puteți să ștergeți și folderul **/deps** sau **/assets** întrucât se pot adăuga la testare. Nu este recomandat să faceți acest lucru întrucât îngreunează mult testarea în cazul în care versiunea curentă a bibliotecilor/resurselor diferă de versiunea utilizată la momentul scrierii temei.

egc/teme/2023/02.txt · Last modified: 2023/12/04 13:47 by victor.asavei