

## Tema 1 - Tank Wars

- **Responsabili:** Vlad Novetschi, Eduard-Andrei Radu, Andrei Voicu, Cristian Lambru
- **Lansare:** 28 octombrie 2024
- **Termen de predare:** 17 noiembrie 2024, ora 23:59
- **Regulament:** [Regulament general](#)
- **Notă:** Orice informație ce nu a fost acoperită în acest document este la latitudinea voastră!

În cadrul temei 1, veți implementa un joc inspirat de „[Tank Wars](#)”, dar cu un gameplay mai interactiv, renunțând la mecanica turn-based. Puteți viziona un mic demo realizat pe baza framework-ului de laborator, care ilustrează cerințele jocului și prezintă un aspect vizual minimal ["video"](#).

De asemenea, puteți descărca demo-ul [aici](#) pentru a-l testa și a înțelege mai clar comportamentul de joc.

### Detalii de implementare

În continuare, vom prezenta o abordare de implementare pe care v-o recomandăm *cu căldură* :).

Este important de menționat faptul că abordarea descrisă mai jos nu este obligatorie. Aveți libertatea să alegeți modalitatea de implementare, cu observația că **rezultatul vizual și comportamentul de joc trebuie să includă toate elementele conținute în demo**.

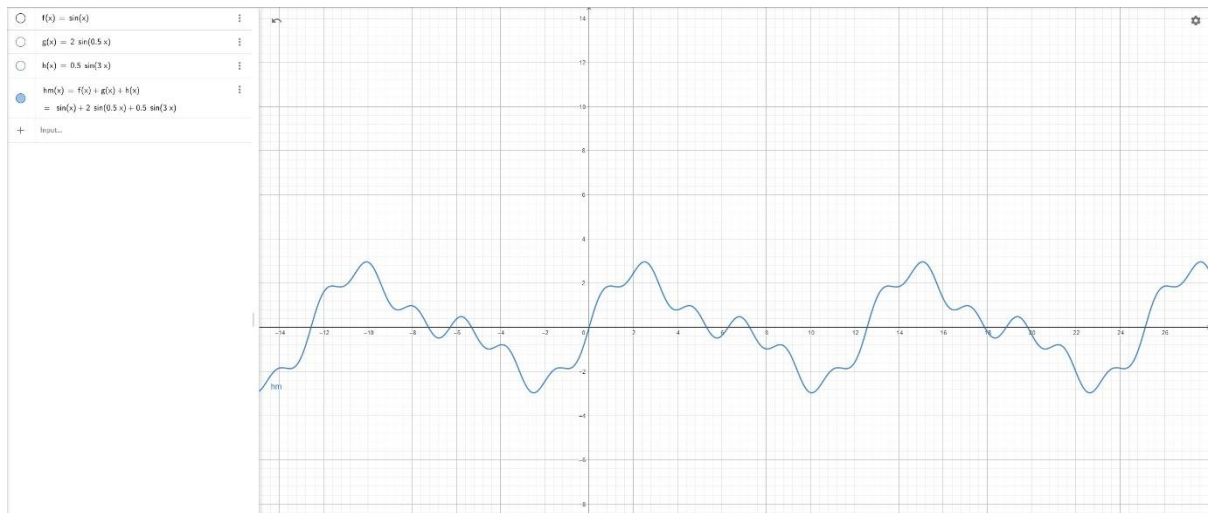
### Construcție teren

Terenul va fi construit prin operația de eșantionare a unei funcții pe un interval ales de voi. Funcția folosită trebuie să fie complexă pentru a duce la un teren interesant, dar să și arate suficient de apropiat de un teren plauzibil. Recomandarea noastră este suprapunerea mai multor sinusoide și/sau parabole de diferite amplitudini și frecvențe:

$$f(x)=A1*\sin(\Omega1*t)+A2*\sin(\Omega2*t)+...f(x)=A1*\sin(\Omega1*t)+A2*\sin(\Omega2*t)+...$$

În formula de mai sus, perechea  $(A_i, \Omega_i)$  reprezintă un vector de redimensionare a codomeniului funcției sinus și  $t$  reprezintă valoarea în care se eșantionează. Cu parametrul  $A_i$ , se poate controla amplitudinea și cu  $\Omega_i$ , lungimea perioadei și implicit frecvența.

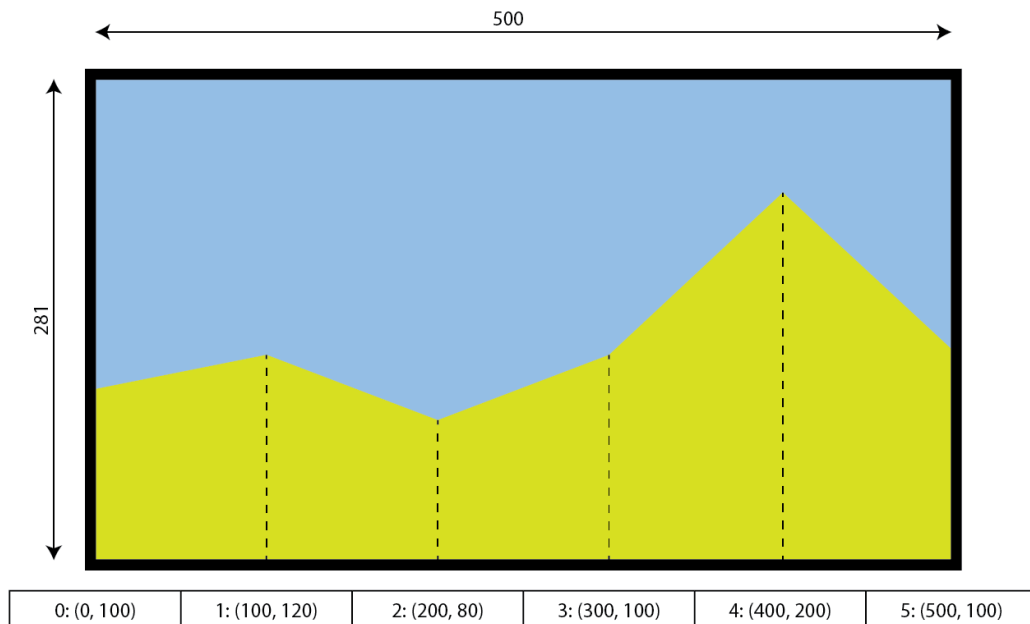
În figura de mai jos, avem o posibilă funcție ce poate servi ca sursă de eșantionare pentru terenul implementat.



## Stocare teren

Terenul poate fi stocat într-un vector de tip hartă de înălțime („height map” în limba engleză), unde fiecare element din vector reprezintă înălțimea unui punct din teren. Vom considera că punctele vor fi spațiate egal între ele.

În figura de mai jos, avem o reprezentare a unei hărți de înălțime cu 6 elemente și modul în care acesta ar arăta desenat într-un spațiu logic de dimensiune 500×281.



În figura de mai sus, perechile din tabloul din partea de jos a imaginii reprezintă coordonatele punctelor de pe suprafața terenului, respectiv perechi (x, y) definite în spațiul logic.

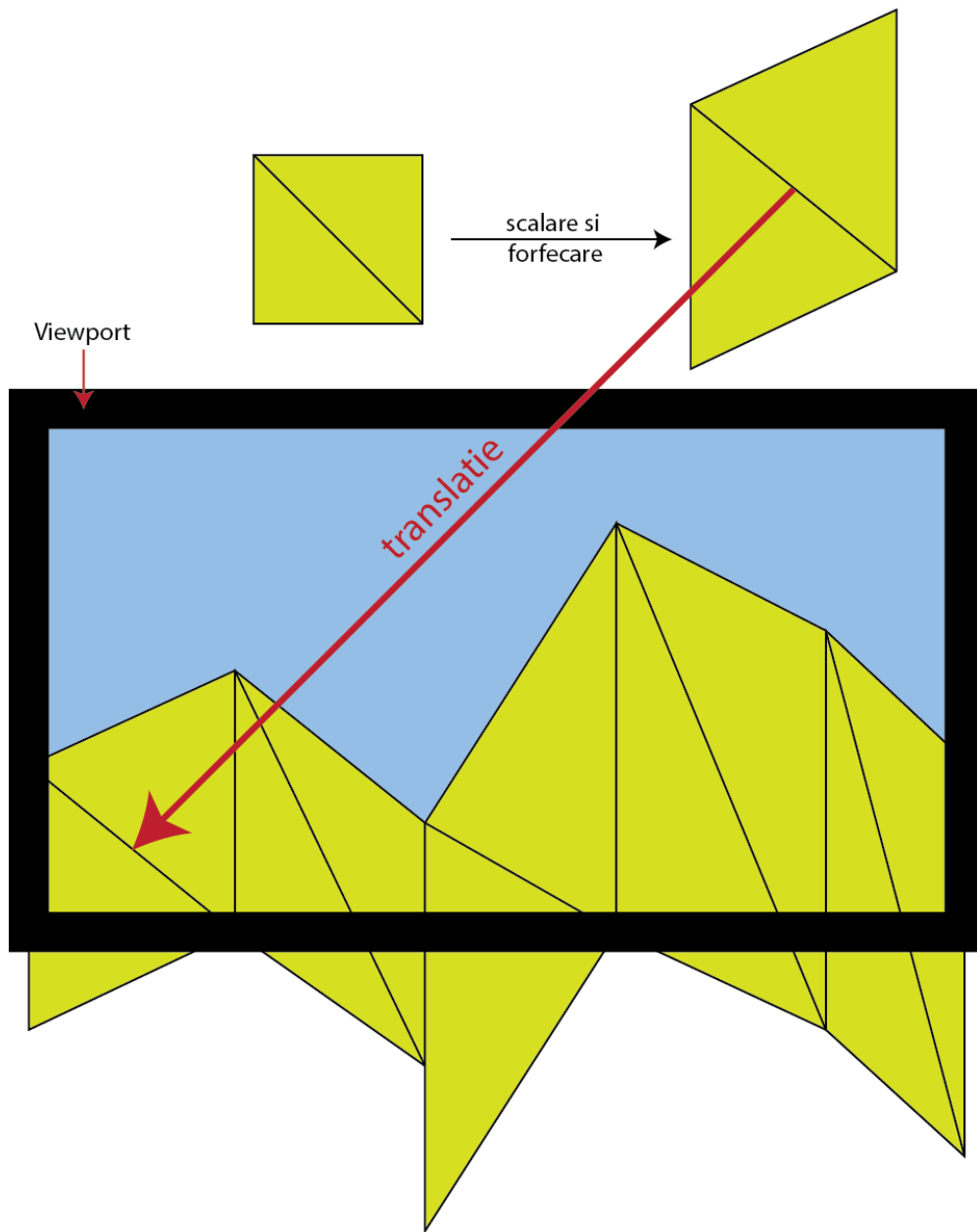
Vom considera faptul că spațiul logic are aceeași dimensiune cu poarta de afișare și cu fereastra. În această temă, nu este necesară utilizarea transformării fereastră-poartă, dar ea poate fi utilizată în anumite contexte pentru bonus. Vedeți exemple de bonus mai jos.

În situația în care spațiul logic are dimensiunea de  $1280 \times 720$ , voi ar trebui să aveți o cantitate mult mai mare de puncte ( $\sim 1$  pentru 10 unități de spațiu), față de exemplul din imaginea de mai sus.

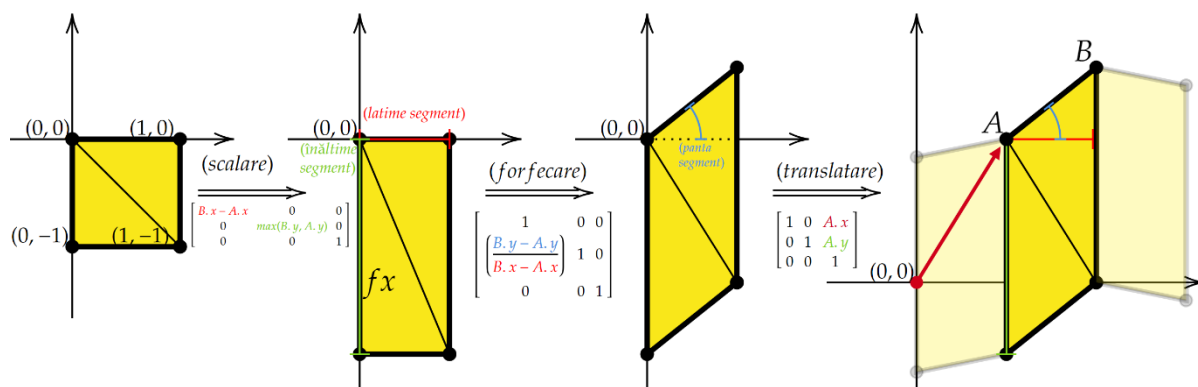
Geometrie și desenare teren

### **Metoda 1**

Terenul poate fi desenat folosind un pătrat care este rasterizat pentru fiecare 2 puncte adiacente din harta de înălțime, aplicându-i-se transformările necesare pentru a avea laturile laterale orientate vertical, latura de sus suprapusă cu segmentul care unește cele 2 puncte și latura de jos undeva sub ecran. Transformarea necesară pentru acest lucru este, pe lângă translație și scalare, forfecarea:

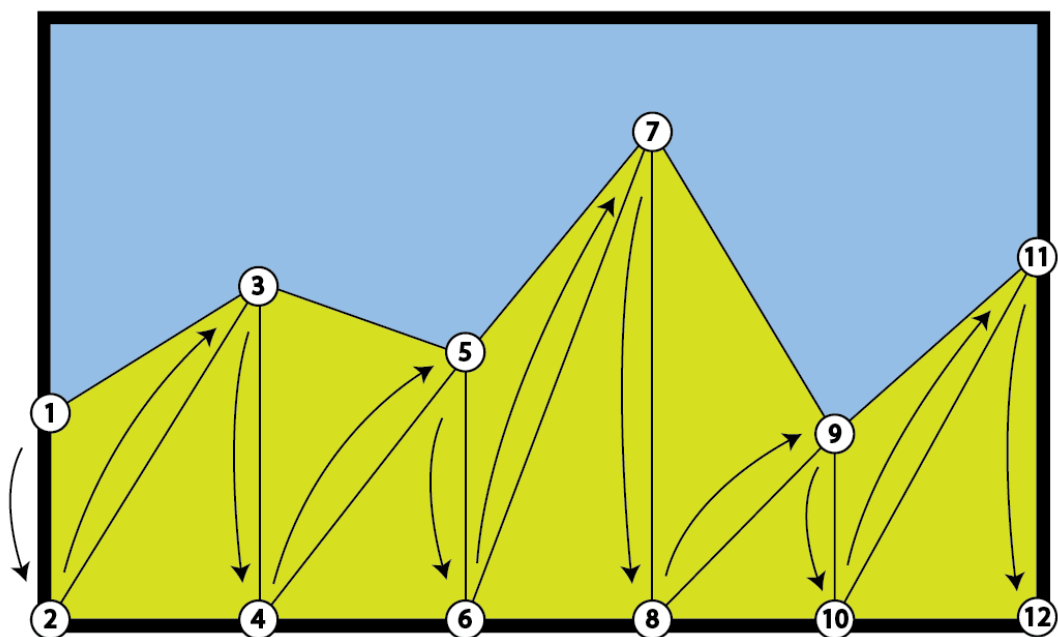


În figura de mai jos, sunt descrise în detaliu transformările necesare acestei metode:



## Metoda 2

O metodă alternativă de desenare a terenului utilizează crearea unui model ce conține geometria inițială a terenului și desenarea acestui model în fiecare cadru. Această abordare poate fi implementată prin utilizarea unei topologii de tip „triangle strip” ca în figura de mai jos:



În figura de mai sus, se consideră inactivă optimizarea de tip „face culling”.

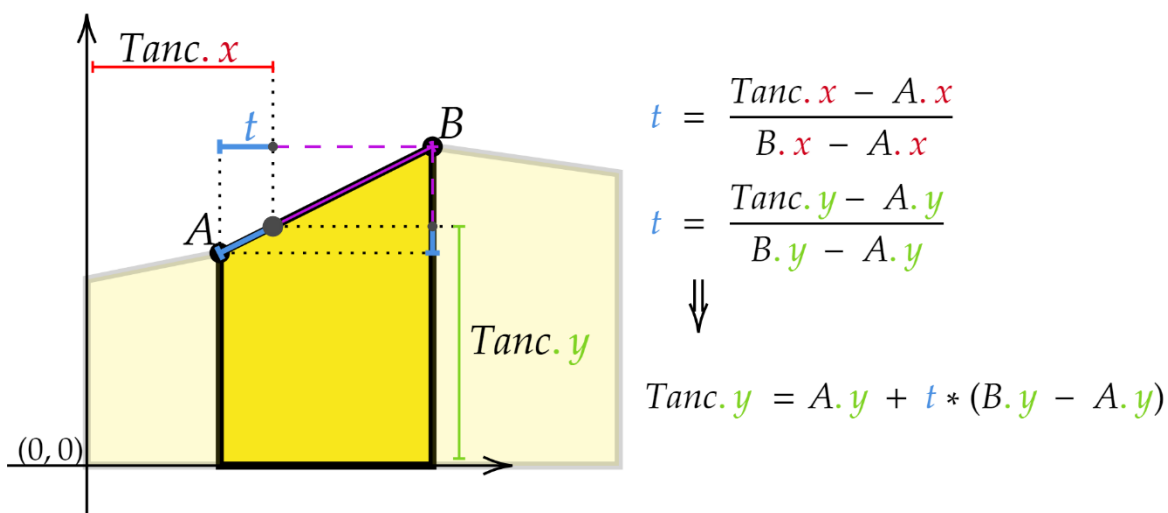
Pentru această metodă, modelul trebuie recreat la finalul cadrelor în care geometria terenului este deformată de cele 2 scenarii posibile:

1. Intersecția unui proiectil cu suprafața terenului;

2. Situația în care are loc procesul de „alunecare de teren” în puncte de pe suprafață.

Plasare/orientare tanc pe teren

În imaginea de mai jos, se poate observa abordarea de calcul a poziției y a unui tanc, în situația în care cunoaștem componenta x a coordonatei lui. În secțiunea „Deplasare tanc”, mai jos, se descrie abordarea recomandată de noi pentru deplasarea stânga-dreapta a unui tanc. Acest proces determină cunoașterea componentei x a coordonatei tancului. Pentru plasarea lui pe teren, este necesară doar calcularea componentei y, după cum este detaliat în imaginea de mai jos:



Tancurile se vor orienta automat în funcție de panta terenului pe care se află. Rotația acestora se va ajusta pentru a fi perfect aliniată cu înclinația suprafeței, asigurând astfel o poziționare realistă, indiferent de forma terenului generat.

Orientarea tancului se realizează printr-o rotație a geometriei. Unghiul de rotație a tancului se poate calcula după cum urmează:

$$\text{unghi} = \arctan(yV \rightarrow xV \rightarrow) \quad V \rightarrow = B - A \quad \text{unghi} = \arctan(yV \rightarrow xV \rightarrow) \quad V \rightarrow = B - A$$

Vom folosi acest unghi în transformarea rotației tancului. Pentru a funcționa fără alte transformări ajutătoare de translație, tancul trebuie construit cu centrul laturei de jos în originea sistemului de coordonate. Altfel, coordonata acestui centru trebuie calculată, iar rotația se va face mai întâi translatând acel punct în origine, aplicând rotația și translatând înapoi.

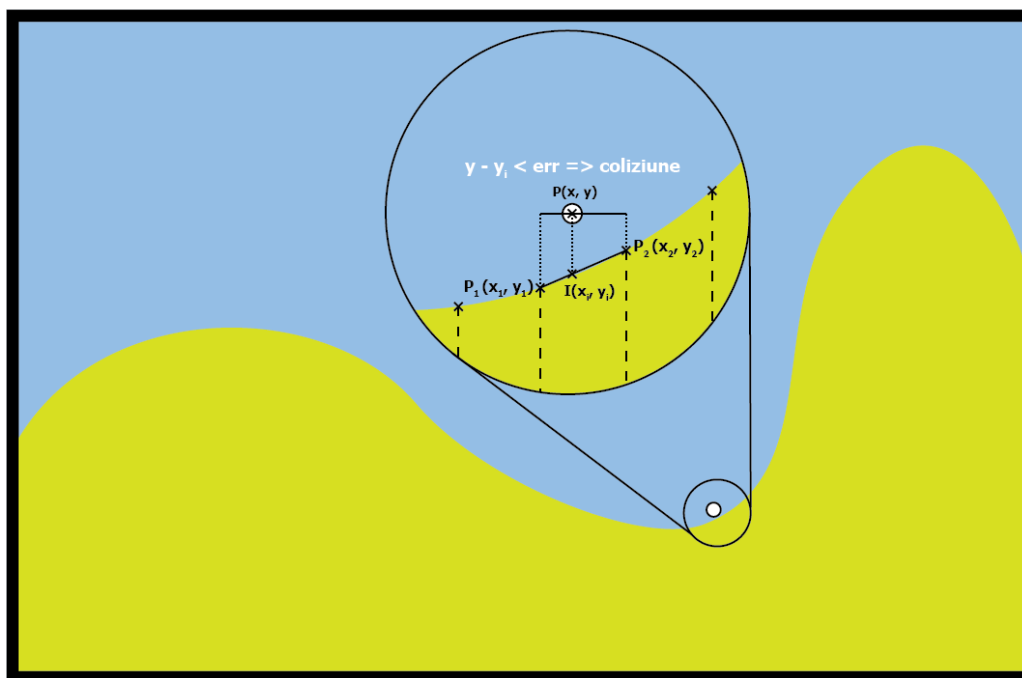
Utilizați atan2() pentru calcularea arctangentei.

Detecție coliziune proiectil-teren

O coliziune are loc dacă un proiectil se intersectează cu suprafața terenului. Pentru simplitate, putem considera că o coliziune are loc atunci când proiectilul se află la o distanță mai mică de un prag (ales de voi) de punctul aflat la aceeași coordonată x

de pe teren. Dacă acest punct nu se află direct în harta de înălțime, el poate fi aflat prin metoda descrisă mai sus, utilizată pentru calcularea poziției unui tanc pe teren.

După cum se poate vedea în imaginea de mai jos, pe baza componentei  $x$  a coordonatei  $P$  a proiectilului, se poate calcula componenta  $y$  a proiecției proiectilului pe teren. Se consideră că proiectilul intersectează terenul, când coordonata proiecției pe teren se află la o distanță față de poziția propriu-zisă a proiectilului ce este mai mică decât un prag ales de voi.



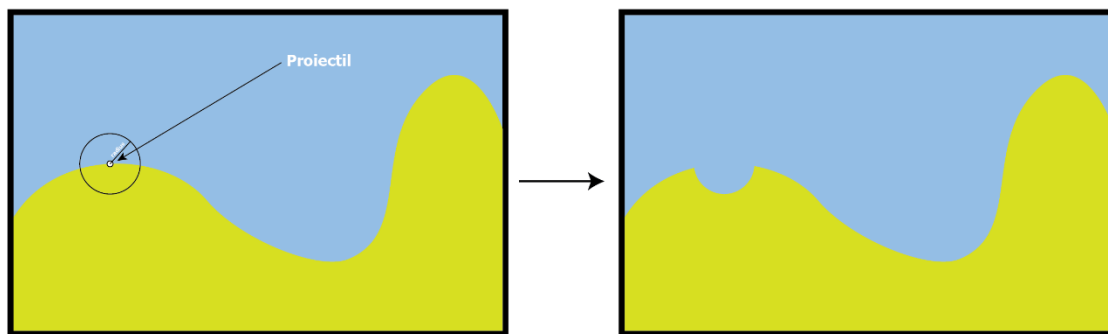
Putem afla  $t$  - coeficientul de interpolare ca fiind procentul pe care îl reprezintă  $(x - x_1)$  din  $(x_2 - x_1)$ . Folosind acest coeficient, putem afla coordonatele lui  $I$  prin interpolarea între punctele  $P_1$  și  $P_2$ . Coliziunea se întâmplă când  $(x - x_i)$  scade sub un prag, care poate fi chiar 0.

$$t = \frac{x - x_1}{x_2 - x_1} \quad I = f(P_1, P_2, t) \quad f(a, b, t) = a * (1 - t) + b * t$$

În formula de mai sus,  $f$  este o interpolare liniară.

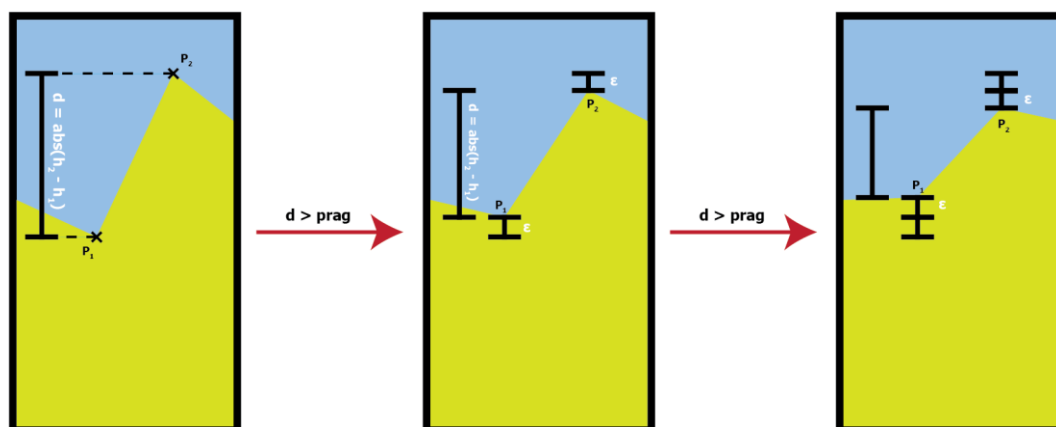
Deformare teren

În caz de coliziune, terenul va fi deformat. Se va utiliza un parametru  $\text{radius}$ , care reprezintă raza de efect a exploziei proiectilului. Efectul exploziei trebuie să fie coborârea nivelurilor din harta de înălțime, în zonele afectate, astfel încât terenul să ajungă în afara razei de efect  $\text{radius}$ .



### Animație de „alunecare de teren”

În caz că 2 puncte alăturate din harta de înălțime au o diferență de înălțime  $dd$  mai mare de un prag  $\text{prag}$ , se va face un „transfer” de înălțime  $\epsilon$  de la punctul cu înălțime mai mare la celălalt. Acest lucru se repetă în fiecare cadru până când  $dd$  este sub  $\text{prag}$ :



Efectul rezultat ar trebui să fie similar cu cel din animația de mai jos, simulând o alunecare de teren:



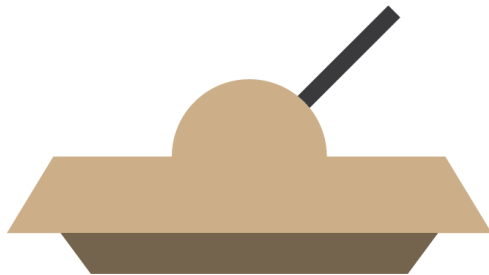


Valoarea  $\epsilon$  utilizată în procesul de „transfer” de înălțime trebuie să țină cont de timpul ultimului cadru, respectiv de valoarea deltaTime.

Valoarea pragului trebuie să fie aleasă astfel încât animația de „alunecare de teren” să fie realizată de-a lungul mai multor cadre. Altfel spus, animația trebuie să se poată observa vizual și nu trebuie să se realizeze instant, deoarece nu s-ar mai numi o animație.

### Construcție tancuri

Tancurile sunt formate din două trapeze suprapuse, orientate în sens opus, ce creează baza vehiculului. Deasupra, un arc de cerc conturează partea superioară a turelei, iar pe aceasta este plasat un dreptunghi care reprezintă țeava tunului, prin care tancul va trage proiectile. Fiecare dintre cele două tancuri va fi desenat într-o culoare distinctă, pentru a le diferenția clar. O imagine a acestui element poate fi văzută mai jos.



Proiectilul va fi reprezentat printr-un disc de cerc.

### Control tancuri

Deplasarea tancurilor se va efectua atunci când sunt menținute apăsate tastele A/D sau săgețile stânga/dreapta ( $\leftarrow/\rightarrow$ ), în funcție de tanc.

### Control direcție turelă

Turela tancurilor se va roti de-a lungul semicercului acestora atunci când sunt menținute apăsate tastele W/S sau săgețile sus/jos ( $\uparrow/\downarrow$ ), în funcție de tanc.

### Deplasare tanc

Deplasarea tancurilor va ține cont de conturul terenului, astfel încât acestea să poată urca și coborî pantele fără a pătrunde în suprafață. Viteza de deplasare poate fi constantă, iar tancurile se vor adapta automat la forma terenului, aliniindu-se perfect cu înclinațiile acestuia.

O metodă simplă de implementare constă în deplasarea tancului pe coordonata X, conform input-ului, iar poziționarea sa pe coordonata Y va fi calculată în funcție de înălțimea terenului în punctul respectiv.

## Lansare proiectil

Lansarea proiectilului se va realiza atunci când jucătorul apasă Spacebar, respectiv Enter, în funcție de tanc. Proiectilul va fi reprezentat grafic printr-un cerc și va fi lansat din vârful turelei, în direcția în care aceasta este orientată. Viteza și traiectoria proiectilului vor depinde de unghiul turelei și de forța aplicată la lansare, simulând efectul gravitației pentru a crea o mișcare realistă. Proiectilul va interacționa cu terenul și cu celelalte tancuri, provocând daune în momentul impactului.

Pentru deplasarea proiectilului, vom considera la lansare un vector cu același unghi ca și tunul tancului și magnitudine aleasă de voi. Pe parcursul deplasării proiectilului, componenta x a vectorului nu se va modifica, iar componenta y va scădea din ce în ce mai repede, cu o accelerație `g` constantă (valoarea ei nu trebuie să fie neapărat 9.8 ca la accelerația gravitațională, ci o valoare pe care o considerați potrivită în contextul jocului pe care îl implementați).

Inițial, vectorul de deplasare a proiectilului este următorul:

$\vec{v} = [\cos(\text{unghiTun}) \sin(\text{unghiTun})] \cdot \text{magnitudine}$   
 $\vec{v} \rightarrow = [\cos(\text{unghiTun}) \sin(\text{unghiTun})] \cdot \text{magnitudine}$

La fiecare cadru trebuie să realizați următoarul proces:

$P = P + \vec{v} \cdot \Delta t$   
 $\vec{v} = \vec{v} - \vec{g} \cdot \Delta t$

În formula de mai sus, P reprezintă poziția proiectilului la un anumit cadru.

## Interfața cu utilizatorul

### Bara de viață

Ambele tancuri vor avea deasupra o bară de viață, reprezentată printr-un dreptunghi wireframe (neumplut), peste care va fi suprapus un alt dreptunghi umplut. Bara de viață va scădea atunci când unul dintre proiectile lovește tancul.

### Afișaj traiectorie

Ambele tancuri vor avea desenată o previzualizare a traiectoriei proiectilului înainte de a trage, bazată pe unghiul turelei și viteza inițială a proiectilului. Aceasta le va permite jucătorilor să ajusteze tunul pentru a îmbunătăți precizia asupra țintei. Traectoria va fi reprezentată printr-o serie de segmente de linie care urmează curba balistică a proiectilului, ținând cont de gravitație. Vor fi calculate puncte pe această traiectorie, care vor fi conectate între ele pentru a forma linia continuă a previzualizării.

### Dectecție coliziune proiectil-tanc

Cele două tancuri vor suferi daune atunci când un proiectil le lovește. Pentru detectarea coliziunilor, se vor folosi cercuri ca forme de coliziune: atât proiectilul, cât și tancurile vor fi încadrate în cercuri cu o rază specifică. Coliziunea va fi determinată

prin verificarea intersecției dintre cercul asociat proiectilului și cercurile asociate tancurilor, astfel încât impactul să fie corect înregistrat.

Barem

Funcționalități de bază (150 puncte)

- Sistem teren deformabil (90p total)
  - Desenare teren 50p
  - Deplasare și orientare corectă a tancurilor pe teren (40p total)
    - Deplasare 20p
    - Orientare 20p
- Construcție elemente scenă (30p total)
  - Desenare geometrie tancuri 20p
  - Desenare geometrie proiectil 10p
- Sistem proiectil (30p total)
  - Lansare și deplasare proiectil - traiectorie fixă 20p
  - Verificare intersecție proiectil - tanc inamic 10p
    - Sistem puncte de viață, un tanc dispare din scenă după ce este lovit de câteva ori

**Atenție!** Punctajul pentru rezolvarea cerințelor specifice realizării terenului se acordă doar în situația în care forma terenului este obținută prin compunerea a cel puțin 2 curbe. Nerespectarea acestei restricții atrage după sine anularea punctajului aferent tuturor cerințelor pentru teren.

Modelele descrise în enunț (tancul, proiectilul, terenul, bara de viață, traiectoria) trebuie construite / generate programatic, **NU** importate.

Realizarea sistemului de lansare și deplasare a unui proiectil se punctează și în situația în care nu implementați controlul traiectoriei și afișajul de ghidaj. Singura observație este să realizați deplasarea proiectilului pe o traiectorie fixă sub forma unei curbe de tip parabolă.

Verificarea intersecției dintre un proiectil și tancul inamic se punctează doar în situația în care se realizează și un sistem de puncte de viață pentru tanc, în care, după ce este lovit de un număr de proiectile, acesta dispare. Această observație este valabilă pentru ambele tancuri.

Funcționalități avansate (75p total)

- Control traiectorie proiectil 10p

- Desenare interfață cu utilizatorul (20p total)
  - Desenare și animație bară de viață dinamică, în partea de sus a tancurilor 10p
  - Desenare ghidaj traiectorie proiectil 10p
- Animație de alunecare a terenului de-a lungul mai multor cadre 15p
- Verificare intersecție proiectil - teren 10p
- Deformare teren la poziția intersecției cu un proiectil 20p

#### Exemple de funcționalități bonus

- nivele în plus (ex: nave pe un ocean, rovere pe marte, etc)
- Animații distrugere tanc
- Animații de explozie la impactul proiectilului
- Alte efecte vizuale (nori, praf, etc)
- Obiecte interactive sau de decor împrastiate prin nivel
- Camera shake
- mai multe tipuri de proiectile (ex care să se rostogolească, sau să adauge teren, ghidate, etc)
- mai multe tipuri de tancuri (ex compuse din mai multe elemente animate)
- sistem upgrade-uri (raza mai mare proiectile, mai mult hp etc)
- mod singleplayer contra AI.
- țintire cu mouse-ul.
- mod sidescroller cu mai mulți inamici
- reprezentare mai avansată a terenului (care să permită de exemplu tuneluri)
- alte mecanici care să facă gameplay-ul mai interesant (elemente rogue-like, turn-based, inventar, etc)
- Deplasare tanc pe suprafața terenului care să țină cont de forma și lungimea acestuia din urmă
- deplasare tancului să țină cont de forma terenului (ex: să meargă mai încet / să alunece când e pe o porțiune abruptă)
- orice altă funcționalitate interesantă/creativă la care va gândiți

#### 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 **Tema1**, cu fișierele **Tema1.cpp** și **Tema1.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 **Tema1**, 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/Tema1/Tema1.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.