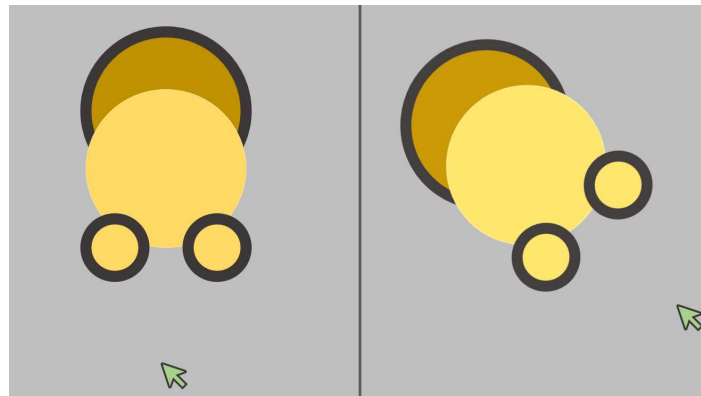


Tema 1 - Survival Shooter

În cadrul temei 1 se va implementa un survival shooter simplu cu camera top-down în care jucătorul trebuie să distrugă cât mai mulți inamici, înainte să moară.

Jucătorul

Jucătorul este reprezentat de câteva primitive geometrice (minimum 3), poziționate sugestiv, astfel încât direcția în care privește jucătorul să fie clară. Un exemplu este cel din figura de mai jos, în partea stângă, în care jucătorul privește în jos. Direcția în care privește utilizatorul este dată de cursorul mouse-ului. Acest lucru se poate observa de asemenea în figura de mai jos, unde sunt prezentate două orientări diferite ale personajului, în funcție de poziția cursorului mouse-ului. Acesta se deplasează pe axele X și Y (sus, stanga, jos, dreapta), cu tastele W, A, S, D.

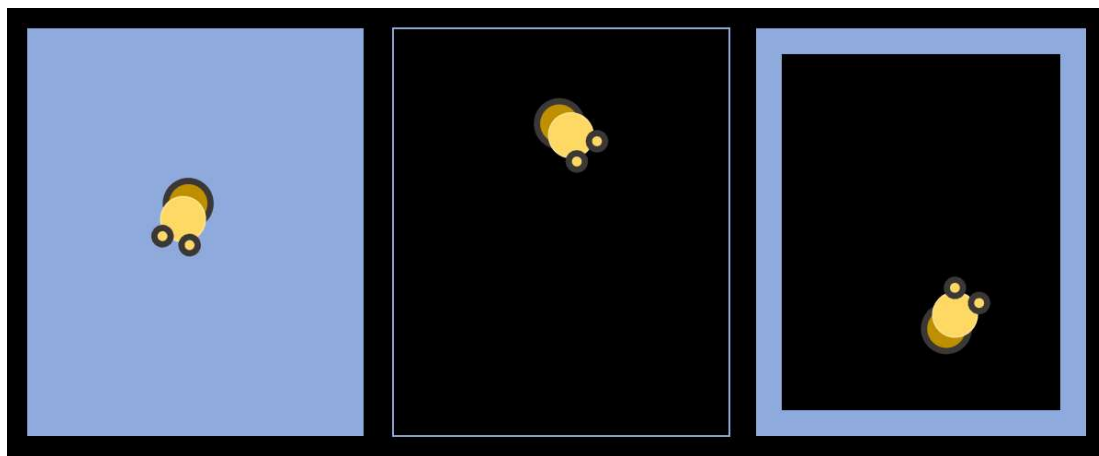


Harta

Spatiul de joc este un dreptunghi. Figura de mai jos prezintă moduri în care se poate reprezenta harta:

- dreptunghi solid (un obiect desenat în spatele tuturor elementelor din joc) - stanga
- dreptunghi wireframe - centru
- prin desenarea peretilor spatiului - dreptunghiuri solide - dreapta

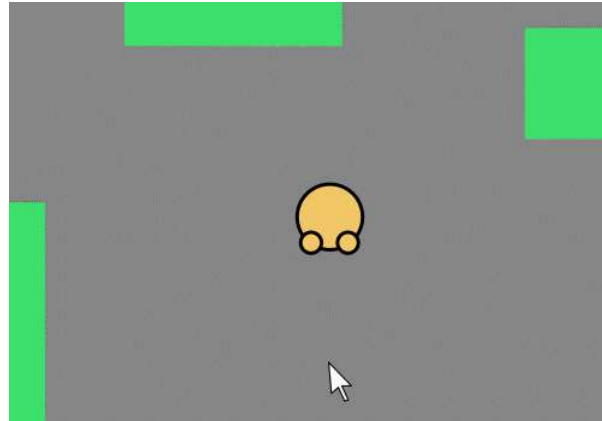
Jucătorul nu poate să iasă din harta. Harta trebuie să fie mai mare decât fereastra de afișare.



Pe harta vor fi și alte obstacole, reprezentate prin dreptunghiuri scalate diferit, prin care jucătorul nu poate să treacă.

Proiectilele

Jucatorul poate sa traga cu proiectile reprezentate prin dreptunghiuri negre (la click stanga cu mouse-ul). Proiectilele vor fi lansate din pozitia jucatorului, in directia in care jucatorul priveste (data de cursorul mouse-ului). Traectoria acestora va liniara, iar viteza de deplasare, constanta. Ele vor fi de asemenea rotite in functie de directia de tragere. In momentul coliziunii cu un inamic, proiectilul va fi distrus (va disparea). De asemenea, inamicul va disparea, iar jucatorul va castiga un numar de puncte. Proiectilele nu pot trece prin obstacole. Astfel, la intalnirea unui obstacol sau al unui perete al hartii, proiectilele dispar. Arma jucatorului are o distanta maxima de tragere, astfel ca dupa parcurgerea unei anumite distante, un proiectil dispara, chiar daca nu a intalnit un inamic. Arma jucatorului are si un "fire rate" maxim. Astfel, jucatorul va putea lansa proiectile in mod succesiv dupa o regula de timp (de exemplu, nu va putea trage un proiectil decat o data la minim 0.5 secunde). Animatia de mai jos prezinta un exemplu in care jucatorul trage cu proiectile.



Camera

Camera va urmări în permanență jucatorul (jucatorul va fi desenat tot timpul în centrul ferestrei de afișare). În video-ul de mai jos puteți vedea un exemplu de hartă, în care spațiul logic se deplasează împreună cu jucatorul. Animatia aceasta nu prezintă ce se vede pe ecran (în fereastra de afișare). În fereastra de afișare se va vedea numai ce este inclus în viewport.



Inamicii

Inamicii se reprezintă tot prin forme geometrice simple, dar cu alte culori/primitive, astfel încât să se diferențieze față de jucător. De exemplu, dacă jucătorul este format din cercuri crem, asemenea figurii, inamicii pot fi reprezentați prin aranjamente de patrate roșii. Inamicii se vor crea încontinuu de-a lungul jocului, fie câte unul pe rând (de exemplu, să apară un inamic nou la 3 secunde), fie în valuri (de exemplu, să apară 5 inamici noi la 10 secunde). Fiecare inamic are o viteză diferită de deplasare. Pozițiile unde apar inamicii și vitezele acestora se pot stabili aleator sau se pot calcula după diferite reguli, dar nu trebuie să fie

hardcodate. Inamicii dispar cand sunt loviti de un proiectil al jucatorului. Inamicii trebuie sa urmareasca incontinuu jucatorul. Inamicii pot trece prin obstacole.

Gameplay

Scopul jocului este ca jucatorul sa supravietuiasca inamicilor cat mai mult timp. Jucatorul porneste de la un nivel maxim de viata (health). La fiecare coliziune cu un inamic, acest nivel scade. Daca nivelul scade la 0, atunci jucatorul moare si jocul se termina. Healthbar-ul se va prezenta pe ecran (orizontal, in partea de sus, sau vertical, in partea dreapta) prin doua dreptunghiuri:

- un dreptunghi wireframe, care reprezinta nivelul maxim de viata
- un dreptunghi solid, in interiorul celui wireframe, care reprezinta nivelul curent

Scorul va creste la eliminarea inamicilor. Scorul se va afisa periodic in consola sau se va ilustra pe ecran prin doua dreptunghiuri, similar cu healthbar-ul (scorebar).

Detalii de implementare

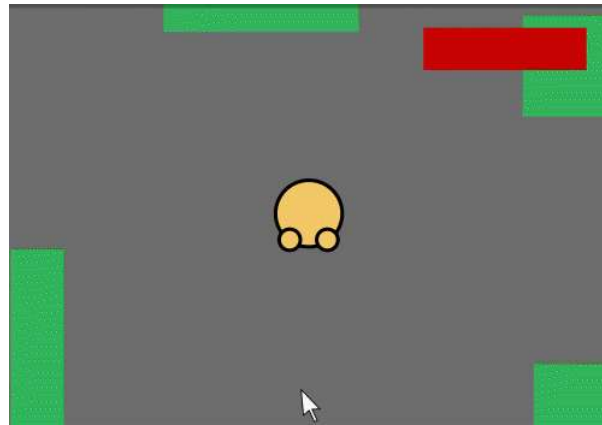
- Crearea mediului inconjurator trebuie sa se realizeze in asa fel incat consumul de memorie si timpul de redare sa fie optime. Nu creati incontinuu obiecte care sa reprezinte inamici si proiectile care apar si dispar din spatiul de desenare!!!! O implementare eleganta este sa creati un singur obiect in functia init(), iar in functia Update() sa dati comanda de desenare pentru acel obiect de mai multe ori, de fiecare data la alta pozitie si cu alt factor de scalare.
- Toate animatiile trebuie sa fie independente de timpul de procesare al unui cadru.
- Veti avea de implementat urmatoarele tipuri de coliziuni
 - jucator - inamic
 - jucator - perete harta
 - jucator - obstacol
 - proiectil - inamic
 - proiectil - perete harta
 - proiectil - obstacol
 - inamic - perete harta

Mai multe informatii despre ce reprezinta coliziunea si cum se poate implementa:

- https://developer.mozilla.org/en-US/docs/Games/Techniques/2D_collision_detection
[https://developer.mozilla.org/en-US/docs/Games/Techniques/2D_collision_detection]
- <https://learnopengl.com/In-Practice/2D-Game/Collisions/Collision-detection>
[<https://learnopengl.com/In-Practice/2D-Game/Collisions/Collision-detection>]
- <https://www.youtube.com/watch?v=aTbw71EpamY> [<https://www.youtube.com/watch?v=aTbw71EpamY>]
- https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection
[https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_collision_detection]

Mentione: in cazul acestei teme, se accepta coliziuni simple (cerc-cerc si cerc-dreptunghi aliniat cu axele).

Animatia de mai jos prezinta un exemplu de fereastra de afisare a jocului, in care jucatorul trage in inamici (reprezentati cu rosu). La coliziunea cu unul dintre inamici, healthbar-ul este modificat. La intalnirea unui pick-up, nivelul de viata este crescut din nou (aceasta cerinta nu este obligatorie). In animatie se poate observa ca jucatorul este in permanenta in centrul ferestrei de afisare si ca acesta nu poate trece dincolo de limitele hartii.



Bonus

- Traiectorii complexe pentru inamici - acestia continua sa urmareasca jucatorul, dar incearca sa se fereasca de proiectile
- Afisarea unui minimap
- Crearea unui tip de inamic mai complex - care ataca jucatorul cu proiectile
- Scena mult mai complexa
- Diverse pick-up-uri (ex: element care creste nivelul de viata, element care mareste proiectilele pentru o anumita perioada, element de rapid fire mode in care arma isi modifica fire rate-ul astfel incat sa traga o data la minim 0.2 secunde, element care transforma arma intr-un shotgun astfel incat sa se lanseze mai multe proiectile la o tragere, etc)

Exemplu (doar pentru orientare): https://www.youtube.com/watch?v=vcezerC5fDk&ab_channel=TasselsPlays [https://www.youtube.com/watch?v=vcezerC5fDk&ab_channel=TasselsPlays]

Functionalitati obligatorii

Barem orientativ pentru realizarea functionalitatilor (din 150 puncte):

- Desenare jucator - 15p
- Miscare si orientare jucator - 10p
- Harta (desen harta, desen obstacole) - 15p
- Camera - 25p
- Desenare inamici - 15p
- Miscare si orientare inamici - 5p
- Proiectile (tragere si desenare) - 20p
- Coliziuni - 25p
 - jucator - inamic (5p)
 - jucator - perete harta (5p)
 - jucator - obstacol (2.5p)
 - proiectil - inamic (5p)
 - proiectil - perete harta (2.5p)
 - proiectil - obstacol (2.5p)
 - inamic - perete harta (2.5p)
- Scor si healthbar (functionalitate si UI) - 20p

Intrebari si raspunsuri

Pentru intrebari vom folosi forumurile de pe moodle. Orice nu este mentionat in tema este la latitudinea fiecarui student!

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 **src/lab_m1** 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 lab_m1 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 lab_list.h trebuie adaugata si calea catre header-ul temei. De exemplu: `#include "lab_m1/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 curatat proiectul Visual Studio:
 - click dreapta pe proiect in **Solution Explorer** → **Clean Solution**
 - si stergeti folderul **/build/.vs** (daca nu il vedeti, **este posibil sa fie ascuns**)
- SAU stergeti complet folderul **/build**
- in cazul in care arhiva tot depaseste limita de 50MB (nu ar trebui), puteti sa stergeti si folderul **/deps** sau **/assets** 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

14 noiembrie ora 23:59

Responsabili tema

- Chris Luntraru
- Miruna Chiricu
- Andrei Lapusteanu
- Anca Morar

egc/teme/2021/01.txt · Last modified: 2021/11/14 15:41 by anca.morar