

Structură Programarea Aplicațiilor de Simulare

1 Curs & Laborator

1.1 Curs

În cadrul cursului se vor aborda concepte teoretice întâlnite des în programarea aplicațiilor de simulare și în dezvoltarea jocurilor video. De exemplu:

- Optimizări low level
- Logica jocurilor & Entity Component Systems
- Transformări & randare
- Tehnici de baza de randare
- Tehnici avansate de randare
- Detecția coliziunilor & partiționare spațială
- Fizică & Animație
- Programare pe GPU
- AI în jocuri

Laboratoarele vor fi alocate pentru a lucra la proiect și întrebări, atât tehnice cât și legate de proiect.

2 Notare

2.1 Descriere

Notarea va fi pe bază de proiect.

Trebuie să realizați o aplicație de simulare sau un joc care folosește concepte studiate în cadrul cursului, sau alte concepte similare.

Pentru proiect puteți folosi absolut orice game engine doriți (*Unity*, *Unreal Engine*, *Cry Engine*, *Godot*, etc..), puteți folosi orice framework-uri low-level doriți (*MonoGame*, *SFML*, *Raylib*, *DirectXTK*, *EnTT*, *Box – 2D*, *Cocos2d – x*, *Bullet Physics*, etc...) sau puteți chiar să scrieți totul de la 0 sau să folosiți un engine custom creat de voi (*Windows API*, *OpenGL*, *DirectX*, *Vulkan*, etc...)

Pentru proiectele care folosesc framework-uri low-level sau care sunt scrise de la 0, pe lângă funcționalitățile aplicației de simulare, va fi apreciat și efortul depus în plus.

Proiectul trebuie realizat în echipă, echipele putând avea 5-10 persoane. Puteți lucra și individual sau în echipe mai restrânse, dar dacă alegeți asta, va fi nevoie să anunțați înainte.

Puteți folosi lucrarea de licență pe post de proiect în cazul în care aceasta este în temă cu obiectivele

cursului. Puteți de asemenea să studiați un paper care descrie o tehnică avansată folosită în aplicațiile de simulare și să o implementați corect în locul proiectului. Dacă ați realizat deja proiecte care respectă cerințele acestei materii, le puteți prezenta.

2.2 Cerințe proiect

Cerințele minime pe care trebuie să le îndeplinească aplicația (pentru nota 5):

- Aplicația trebuie să ruleze
- Interfață grafică
- Să aibă cel puțin o funcționalitate a unei aplicații de simulare
- Github Repository & README
- Prezentarea aplicației

Pentru o notă mai bună, este nevoie ca aplicația să conțină mai multe funcționalități specifice aplicațiilor de simulare, sau să implementeze tehnici avansate.

Funcționalități orientative ale unui proiect de nota 10:

- Fizici - În aplicație, obiectele au comportament fizic. Acestea se comportă natural și nu apar glitch-uri. În funcție de specificul aplicației, poate este necesar să se implementeze funcționalități fizice noi pe lângă ce oferă engine-ul (de exemplu fizici de înot sau de control al unei rachete). Dacă se aplică proiectului, implementarea unui sistem de partiționare spațială este bine-venită. Folosirea de animații/sisteme de particule este de asemenea bine-venită.
- Grafică - În aplicație trebuie să existe elemente grafice custom. Acestea pot fi de exemplu: post-procesări complexe, sisteme diferite de iluminare, deformări ale obiectelor folosind shadere, tessellare, etc..
- Inteligență Artificială - Aplicația implementează cel puțin o tehnică de Inteligență Artificială. Aceasta poate fi absolut orice: State Machines, Arbori de Decizie, căutări (A^* , Alpha-Beta...), Algoritmi Genetici, Machine Learning, etc..

2.3 Exemple de proiecte:

- Joc de curse
- Mașină care se conduce singură
- Joc de aterizat pe planete folosind o rachetă
- Simulare și generare a unui sistem solar
- Explorarea oceanului într-un submarin
- Joc cu fractali 3D similar cu Marble Marcher
- Mini Game Engine cu Deferred Renderer
- Joc în care legile fizicii sunt alterate
- Simulator de zbor
- Custom Minecraft
- Simularea fenomenelor naturale asupra unui teren
- Joc similar cu Portal

- Joc similar cu Superliminal
- Oraș generat procedural
- Nori volumetrici
- Simularea apei
- Generare procedurală de vegetație
- Roller-coaster VR
- Joc similar cu Auto Chess/ Dota Underlords impotriva unor boți
- Shooter in care timpul este alterat
- Joc într-un spațiu non-Euclidian, similar cu Hyperbolica
- Joc 4D
- Simulare de fluide fum
- Simularea oceanului
- Custom DOOM engine
- Custom Out Run engine
- Fractali sincronizati pe muzica
- Orice altceva cu o complexitate similară