# Abstract

Această documentație își propune să prezinte succint modul de funcționare al jocului video intitulat „Orizonturi cenușii" din punct de vedere tehnic.

Vom începe prin a ne familiariza cu acest joc printr-o scurtă descriere. „Orizonturi cenușii" este un joc 2-dimensional ce îmbină stilurile jocurilor de strategie, a celor de împușcături (mai precis, shoot-em-up) în care jucătorul se află în controlul unui soldat ce trebuie să își apere teritoriul aflat sub asediu de către inamici. Acest joc se axează principal pe lupta dintre tancuri, împreună și cu alte vehicule puse la dispoziția jucătorului.

# Prezentarea jocului

„Orizonturi cenușii” prezintă o realitate crudă, dată de existența perpetuă a războaielor de-a lungul istoriei întregii omeniri. Intenția acestui joc nu este de a promova războiul, chiar din contră. Prin intermediul efectelor vizuale, a situației prezentate în joc și a deducțiilor jucătorului se realizează o atitudine negativă față de ororile produse de către război.

Jocul este unul bidimensional, dar ce folosește grafică modernă, sunete reale și diverse alte elemente pentru a oferi o notă de realism. În mod normal, astfel de jocuri se pot limita la o anumită grilă de poziționare, ce împiedică fluiditatea în mișcare sau a efectelor grafice. „Orizonturi cenușii" folosește o grilă la nivel de pixel, ce permite o fluiditate maximală în cele două dimensiuni.

## Geneza

„Orizonturi cenușii” a început drept un joc anonim ce își propunea să reproducă un joc de arcadă extrem de popular în anii ’80, intitulat „Battle City”. Acesta urma să fie o demonstrație practică a implementării programării orientate pe obiecte într-un mod cât mai pur, și ușor de interpretat.

În timp ce proiectul prindea amploare, s-a ajuns în mod natural la concluzia că ar fi mult mai indicat extinderea respectivei idei promovate de jocul de arcadă și aducerea lui la standarde moderne.

Astfel s-a născut „Orizonturi cenușii”, joc ce avea să fie supus unui număr extrem de mare de modificări pe parcursul dezvoltării, și evoluării naturale, a acestuia.

## Gameplay

Scopul jocului este dat de către misiuni, ce constituie firul poveștii al acestui joc. Fiecare misiune este formată din diverse obiective, ce antrenează jucătorul și îl pune în situații ce necesită concentrare, experiență și o strategie bine definită.

La începutul fiecărui nivel, jucătorul este reprezentat de un soldat. Acesta poate fi controlat folosind tastele standard (WASD, săgeți), rotirea realizându-se cu ajutorul mouse-ului; jucătorul are posibilitatea de a se folosi de anumite vehicule precum mașini (un bun exemplu este mașina ce poate plasa mine), tancuri (principalul element al acestui joc), etc.

## Elemente tehnice caracteristice

Câmpul de luptă (denumit ulterior „hartă”) permite o dimensiune care se poate extinde în afara rezoluției jucătorului. Harta poate avea, teoretic, o dimensiune infinită (limitată doar de hardware). Harta poate conține un număr nelimitat de obiecte (precum mașini, tancuri, soldați, ziduri, etc.), atât timp cât procesorul (actualizarea stării obiectelor poate fi intensivă) și memoria permit.

Orice obiect aparținând hărții poate avea o dimensiune arbitrară, independentă de celelalte. De asemenea, poziționarea se face asfel încât jucătorul are parte de o experiență cât mai fină. Exista, de asemenea, și posibilitatea unui obiect de a se roti liber, de la 0 pana la 360 de grade in planul bidimensional în care se află.

# Tehnologia folosită

Pentru a fi posibilă realizarea acestui joc, au fost folosite un număr de tehnologii. Urmează o listă a acestora, împreună cu o scurtă descriere a lor:

1. Limbajul de programare — „Orizonturi cenușii” a fost realizat în limbajul de programare Microsoft C# (versiunea 5); acesta este un puternic și bine-cunoscut limbaj ce permite utilizarea paradigmei programării orientate pe obiecte (engl. OOP) implementată cu succes în dezvoltarea jocului. Limbajul C# se bazează pe tehnologia Microsoft .NET Framework, ce a fost concepută cu scopul de a oferi posibilitate de a crea software la un nivel înalt (cât mai abstract de implementarea propriu-zisă) și o mare portabilitate între sisteme de operare și diverse sisteme de operare (C# este capabil de a rula pe toate platformele de dispozitive majore cu ajutorul diverselor implementări precum Mono).
2. Versiunea de Microsoft .NET Framework utilizată aici este varianta portabilă menționată anterior, Mono versiunea 4.0. Implementarea multiplatformă Mono este compatibilă cu cea canonică, rulând doar pe Windows, a Microsoftului. Acest fapt, împreună cu practicile de portabilitate adoptate, i-ar permite teoretic jocului să ruleze pe o varietate de dispozitive precum Android, Windows 8 (Modern UI), Windows Phone, iOS, Xbox, OUYA, etc.
3. Drept platformă de desenare grafică a fost aleasă biblioteca MonoGame; aceasta este o reimplementare open-source și portabilă a Microsoft Xna Framework. MonoGame facilitează desenarea grafică atât în plan bidimensional, cât și tridimensional, oferind de asemenea și capabilități de intrare/ieșire. Principala funcționalitate de desenare folosită aici este cea pur 2-dimensională, bazată pe imagini-componente (engl. sprites), dată de clasa SpriteBatch.
4. Pentru administrarea automată a desenării diverselor stări ale jocului (meniu principal, setări, jocul propriu-zis, interfața Heads-Up Display) s-a folosit GameStateManager; un set de trei clase create de către Microsoft pentru a servi drept un exemplu complet funcțional a stărilor unui joc.
5. Deoarece geometria implicată în poziționarea absolută, a dimensiunilor arbitrare și a rotirii libere a obiectelor aflate pe câmpul de luptă este suficient de dificilă, a fost folosită o bibliotecă pentru MonoGame denumită RotatedRectangle; aceasta implementează clasele ce reprezintă un dreptunghi rotit în radieni, împreună cu metoda de verificare a intersecției între două astfel de dreptunghiuri.
6. Din punct de vedere al inteligenței artificiale, determinarea celei mai potrivite căi pentru a ajunge la un punct oarecare pe hartă devine de o complexitate ridicată. Luând în considerare situația de față, cea mai eficientă metodă este de a folosi poligoane de navigare (engl. Navigation Meshes). Această tehnologie este de obicei folosită în medii tridimensionale, dar poate funcționa și în medii bidimensionale (coordonata Z este întotdeauna 0). Biblioteca ce permite acest comportament se intitulează SharpNav.

# Structura internă a proiectului

## „Soluția” și proiectele aferente

Jocul, ca întreg, este reprezentat de către un set de proiecte C# (engl. solution). Proiectele aferente acestui set sunt:

1. Proiectul partajat „GreyHorizons" (engl. Shared Project) se include în toate proiectelor derivate specifice unei platforme; acesta constituie practic întreaga logică și funcționalitate a jocului;
2. Proiectul „GreyHorizons.Content", care este de fapt un proiect ce nu necesită compilare; acesta conține toate fișierele de conținut aferente jocului (texturi, sunete, fonturi, etc.), aflate atât în original, cât și într-o formă ce poate fi citită de către MonoGame (compilate în .xnb și alte extensii de către MonoGame Content Pipeline).
3. Proiectul „GrayHorizons.Windows.DirectX” reprezintă implementarea propriu-zisă a jocului pe platforma Windows, folosind biblioteca grafică specifică — DirectX (folosind SharpDX).
4. Proiectul „GrayHorizons.Windows.OpenGL” reprezintă varianta OpenGL a jocului pentru platforma Windows.

Modularitatea conferită de această structură a proiectelor permite portarea cu ușurință a acestui joc pe un număr mare de alte platforme (Xbox, Windows 8, etc.).