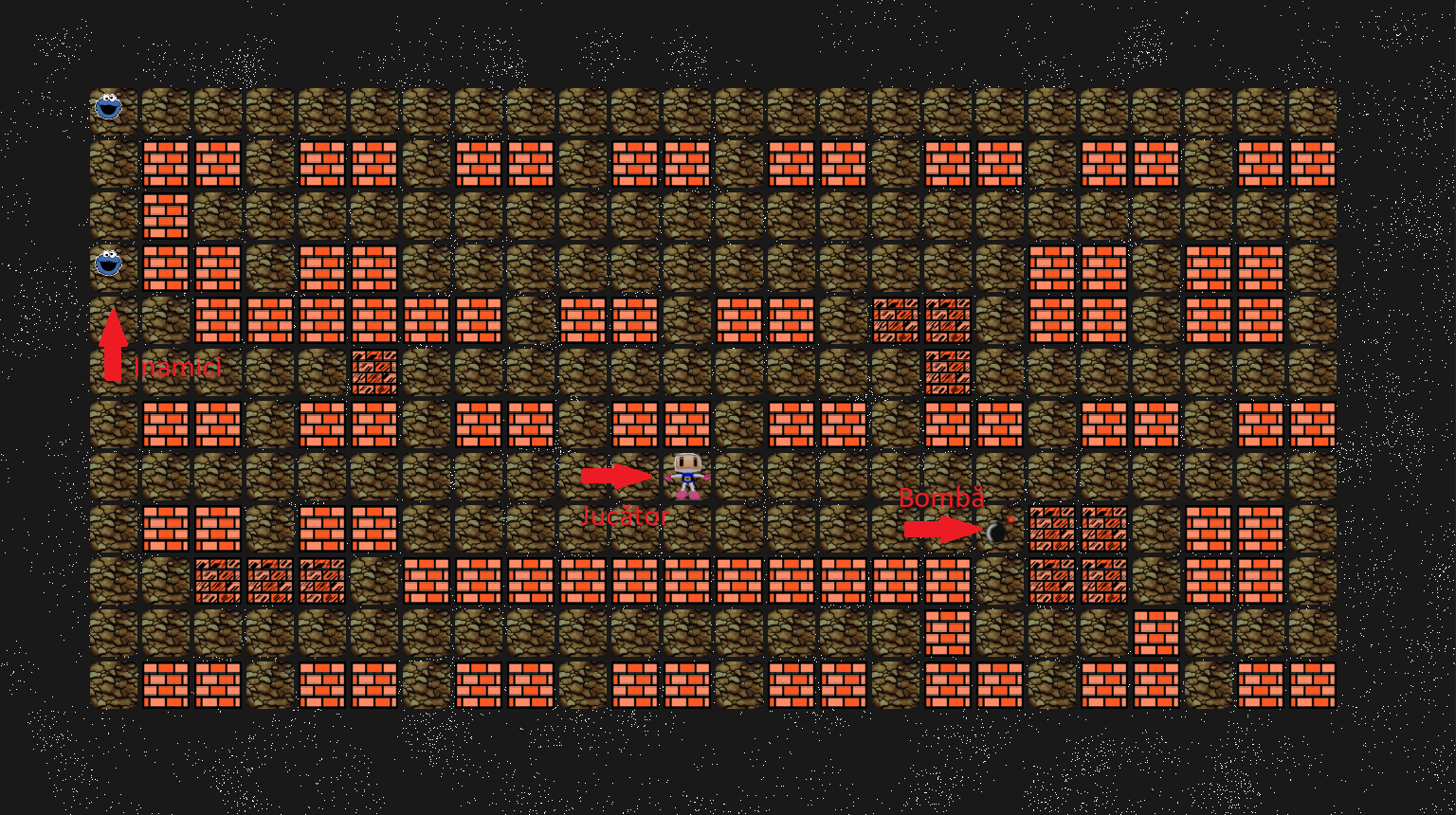
# Bomberman

Echipă:

* Bardaș Denis Adelin
* Brănaci Șerban Mihai
* Chindea Cosmin Mihai

Descriere

Echipa noastră a ales să realizeze jocul Bomberman. Scopul jocului este ca jucătorul să elimine inamicii prezenți pe hartă și să avanseze la nivelul următor. El folosește bombele pentru a învinge inamicii și pentru a sparge blocurile ca să găsească ieșirea.



Harta unui nivel al jocului. Am evidențiat principalele elemente ca jucătorul, bomba și inamicii.

UI

Pentru realizarea acestui proiect am folsit framework-ul MonoGame A red and black logo

Description automatically generated pentru partea de afisare a unei interfețe pentru utilizator. El ne-a ajutat cu organizarea și afișarea elementelor grafice care constituie mediul de joc pentru utilizator.

A screenshot of a computer program

Description automatically generated

*Aici este structura proiectului de timpul MonoGame Desktop Application. În ierarhia de mai jos putem vedea toate utilitarele folosite de noi pentru a afișa mediul de joc și legătura cu partea de logică a proiectului. Framework-ul ne ajută să tratăm intrările de la tastatură ale jucătorului.*



*Sprite-urile folosite pentru afișarea jocului.*

Framework-ul Monogame dispune de funcții și clase speciale pentru afișarea pe ecran a resurselor. Noi am folosit aceste clase și funcții în clasa Bomberman.cs care manageriază jocul din punctul acesta de vedere. Am folosit clase precum GraphicsDeviceManager, SpriteBatch și Vector2. Ca mod de lucru s-au încărcat resursele din foldere și după s-au afișat cu ajutorul funcției Draw().

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

Funcțiile pentru încărcarea resurselor.

Back-End

Pentru partea de back-end (logica) jocul a fost implementat în maniera de lucru OOP.

Design Patterns

Design pattern-urile folosite în proiectul nostrum sunt:

* Observer
* Factory
* Strategy

Factory - acest design pattern a fost folosit la crearea entitățior de Player și Enemy. Acestea moștenesc aceeasi clasă, Entity, și sunt creați de Factory în funcție de nevoi.

A black background with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

Factory Design Pattern



Inițializarea player și enemy

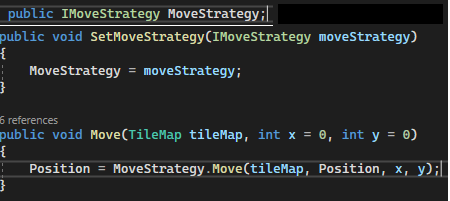
Strategy – acest design pattern a fost folosit pentru separarea tipului de miscare a entităților player și enemy care au restricții diferite pentru deplasare.

A black background with white text

Description automatically generated

A computer screen with text on it

Description automatically generated



Implementarea interfeței IMoveStrategy și a celorlalte două mecanisme

Observer - //vorbit cu Serban

Unit Testing

Pentru Unit Testing am folosit frameworkul MSTests.

Am realizat unit test pentru clasele Tile.cs.

Diagrame

Use Case A diagram of a diagram

Description automatically generated

Activity A diagram of a game

Description automatically generated

State Diagram

A diagram of a game

Description automatically generated