**Mario**

**-DOCUMENTAŢIE-**

**PROIECT REALIZAT DE:**

**Anghel Silvia**

**Cosa Raissa**

**Mitici Alexandru**

**1.Descriere**

Implementarea unui joc de tip Mario în Visual Studio singleplayer. Jucătorul este reprezentat de un pătrat de culoare verde, iar acesta trebuie să ajungă la finalul jocului fără a fi omorât de inamicii reprezentaţi de pătratele de culoare roşie şi neagră şi să adune cât mai multe puncte.

Pe parcursul jocului acesta întâlneşte inamicii pe care îi poate omorî dacă sare deasupra lor sau de care poate fi omorât dacă intră în contact cu ei în alt mod. Scorul este contorizat în partea stanga a ecranului.

De asemenea, există nişte blocuri speciale reprezentate prin semnul întrebării care oferă puncte şi pot ascunde nişte elemente ajutătoare ce îi conferă jucătorului o viaţă în plus sau pot fi nişte simple cărămizi. În cazul în care jucătorul are o viaţă în plus, dar este atins de un inamic, acesta rămâne cu o singură viaţă şi poate continua jocul.

Finalul jocului este marcat de un steag, atunci când jucătorul atinge steagul jocul e terminat, iar jucătorul e câştigător şi are şi rezultatul final al punctelor. Un mesaj de tip pop-up va aparea pe ecran care va anunţa jucătorul că a câştigat. Dacă acesta rămâne fără vieţi, mesajul îl va alerta că a pierdut jocul.

Pentru a controla jucatorul se pot folosi sagetile sau tastele WASD. Pentru a sari se foloseste tasta Space, sageata Sus sau tasta W.

**2.User stories**

1. *“Ca părinte, vreau să fie un joc educativ, astfel încât să ştiu că copilul meu nu are parte de jocuri care nu sunt potrivite pentru vârsta lui.”*
2. *“Ca adult, vreau să fie un joc care să îmi amintească de copilarie, astfel încât să mă bucur că încă se mai ţine cont de jocurile vechi.”*
3. *“Ca student, vreau un joc scurt, astfel încât să nu îmi pierd mult timp şi să pot să studiez.”*
4. *“Ca elev, vreau să pot ţine evidenţa scorului, astfel încât când îmi invit prietenii să putem realiza un clasament.”*
5. *“Ca gamer, vreau să pot colecţiona bonusuri, astfel încât să reuşesc să dobor toţi inamicii şi să îmi perfecţionez rezultatul.”*
6. *“Ca prieten, vreau un joc simplu în care să pot să mă deplasez în mai multe sensuri în caz de am ratat vreun inamic, astfel încât să pot depăşi scorul prietenilor mei.”*
7. *”Ca gamer, vreau să ştiu scorul în timpul jocului, astfel încât să fiu conştient dacă mi-am depăşit celelalte scoruri din timpul jocului.”*
8. *“Ca gamer, vreau un joc 2D old-school, deoarece în ziua de astăzi majoritatea jocurilor publicate sunt 3D.”*
9. *“Ca părinte, vreau un joc simplu şi scurt pe care îl pot juca atunci când am puţin timp liber.”*
10. *“Ca entuziast al jocurilor video, vreau un joc vechi, clasic, implementat într-o manieră diferită ce serveşte drept o alternativă bună.”*

**3.Backlog creation**

**Design de nivel:**

* + Modelarea unei arene de joc
  + Amplasarea platformelor
  + Amplasarea elementelor de grafică de nivel
  + Crearea de obiecte fizice

**Elemente de fizică:**

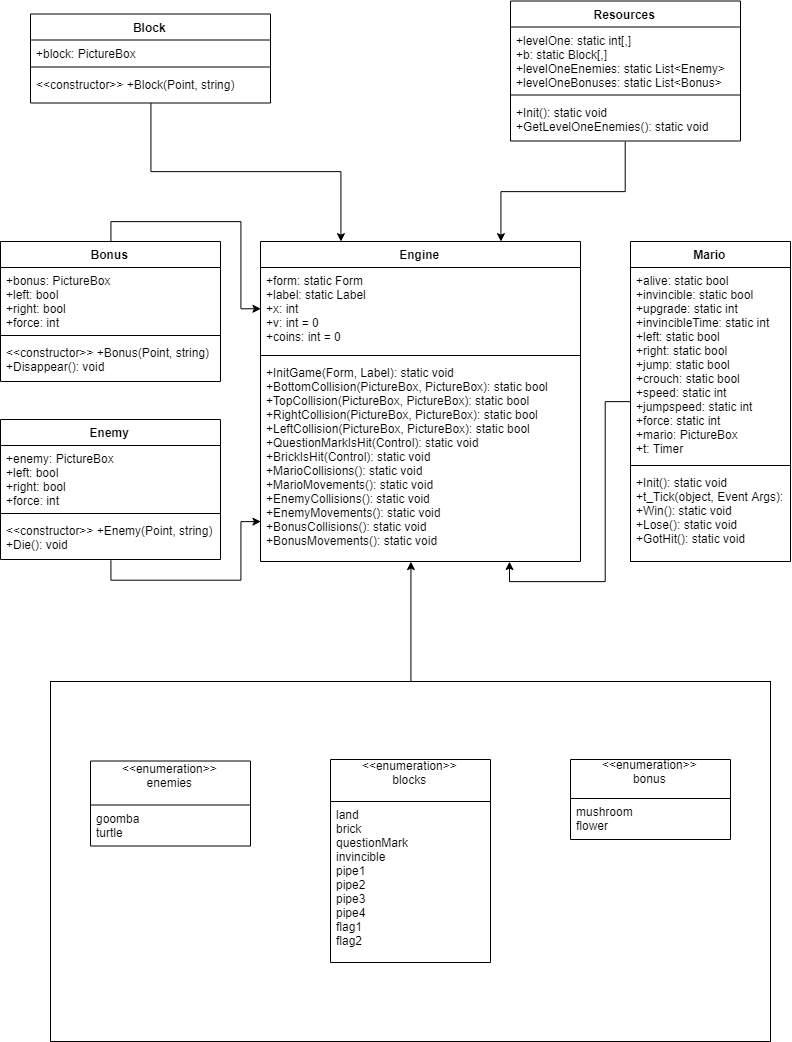
* + Implementarea de coliziuni

**UI(User Interface):**

* + Implemetare HUD
  + Asignare butoane

**Design de grafică:**

* + Desenat grafică background
  + Desenat caractere
  + Desenat platforme şi alte elemente

**4. Diagrama UML**

**5. Arhitectură. Cerinţe de sistem minimale**

|  |  |
| --- | --- |
| **Processor** | 1 GHz |
| **RAM** | 512 MB |
| **Disk space (minimum)** |  |
| 32-bit | 4.5 GB |
| 64-bit | 4.5 GB |
| **Operating System** | Windows 10 |

Deoarece jocul foloseste .NET Framework 4.6 maşina pe care rulează jocul trebuie să aibă minim următoarele proprietăţi[[1]](#footnote-1):

**6. Refactoring si code styling**

Organizare: Fiecare clasă are un fişier propriu ce conţine variabilele şi metodele necesare.

Indentare: Am folosit stilul Allman[[2]](#footnote-2) pentru acolade, iar pentru spacing am folosit taburi. Am eliminat majoritatea liniilor goale, care erau inutile, îmbunătăţind astfel cantitatea codului ce poate fi vazut pe ecran, dar şi calitatea, deoarece codul arată mai organizat în acest mod.

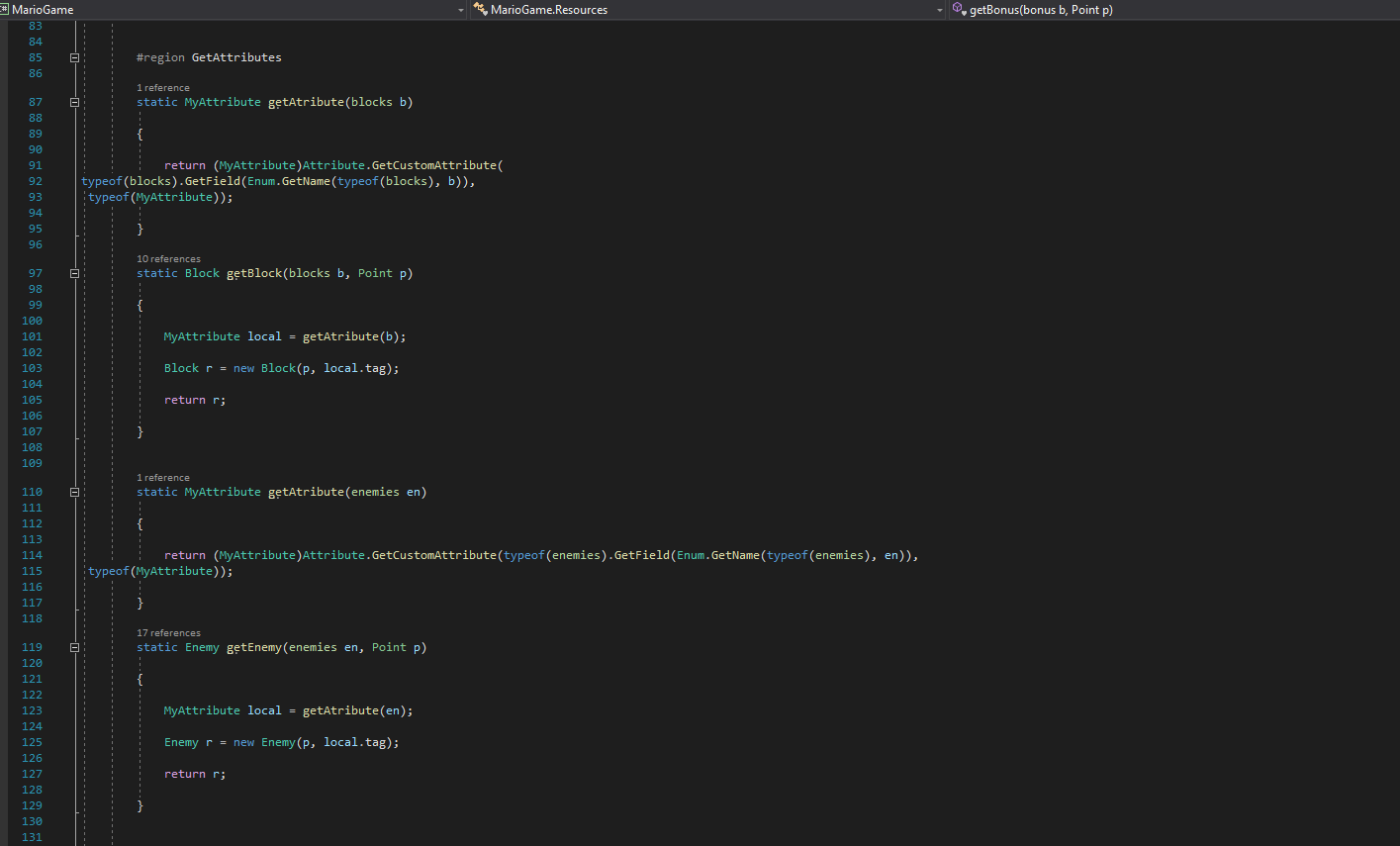
Convenţii de numire a variabilelor şi a metodelor: Am folosit o combinaţie de camelCase şi PascalCase, astfel că variabilele au fost denumite folosind camelCase, iar metodele folosind PascalCase.

Declararea variabilelor din clase a avut loc la începutul clasei, iar variabilele de tip static au avut prioritate. În continuarea variabilelor au fost implementate metodele necesare claselor.

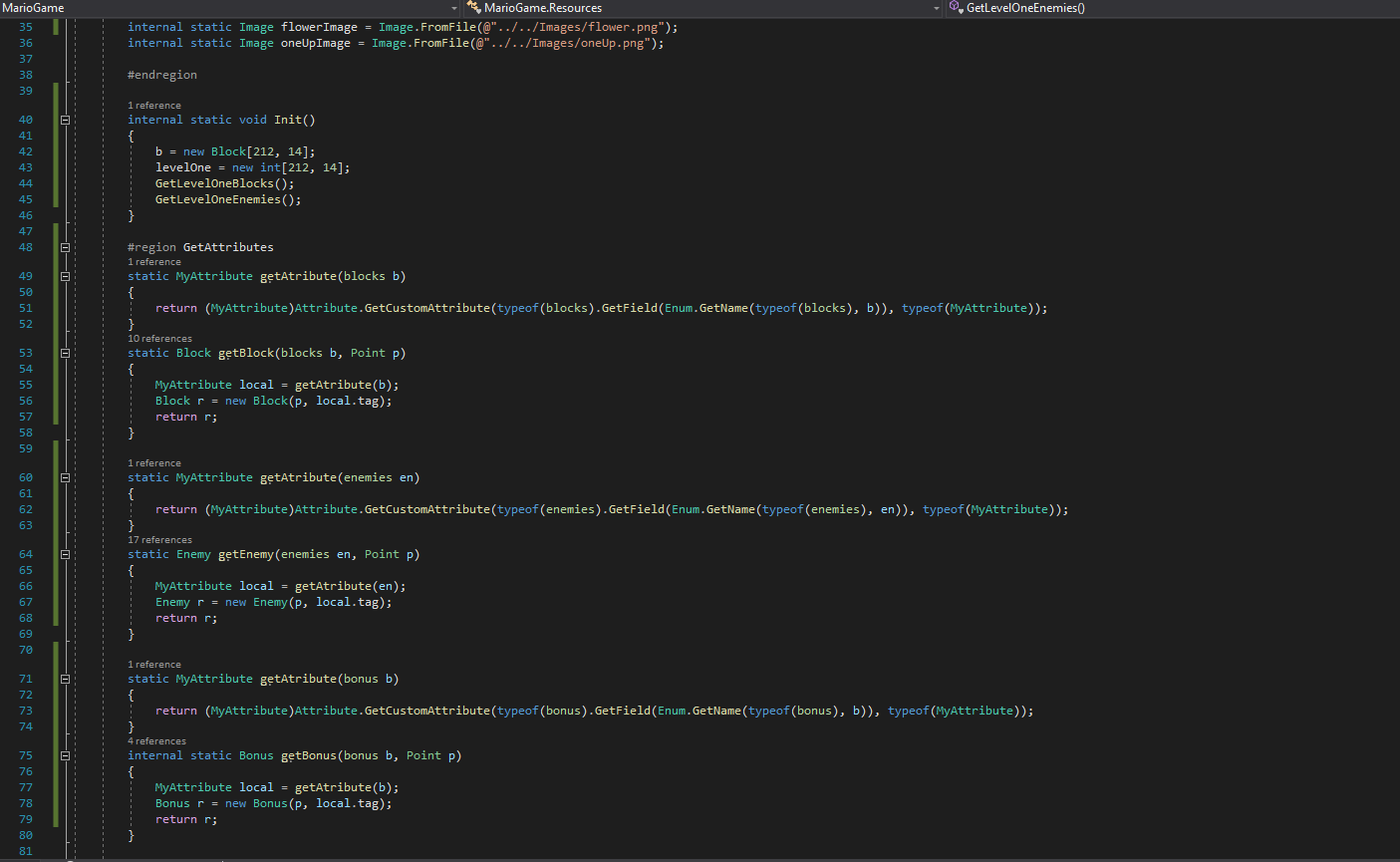
De asemenea, lungimea liniilor de cod este mică spre medie, cu câteva excepţii. Împreună cu celelalte îmbunătăţiri de mai sus, codul este lizibil.

În continuare sunt două exemple de code refactoring.

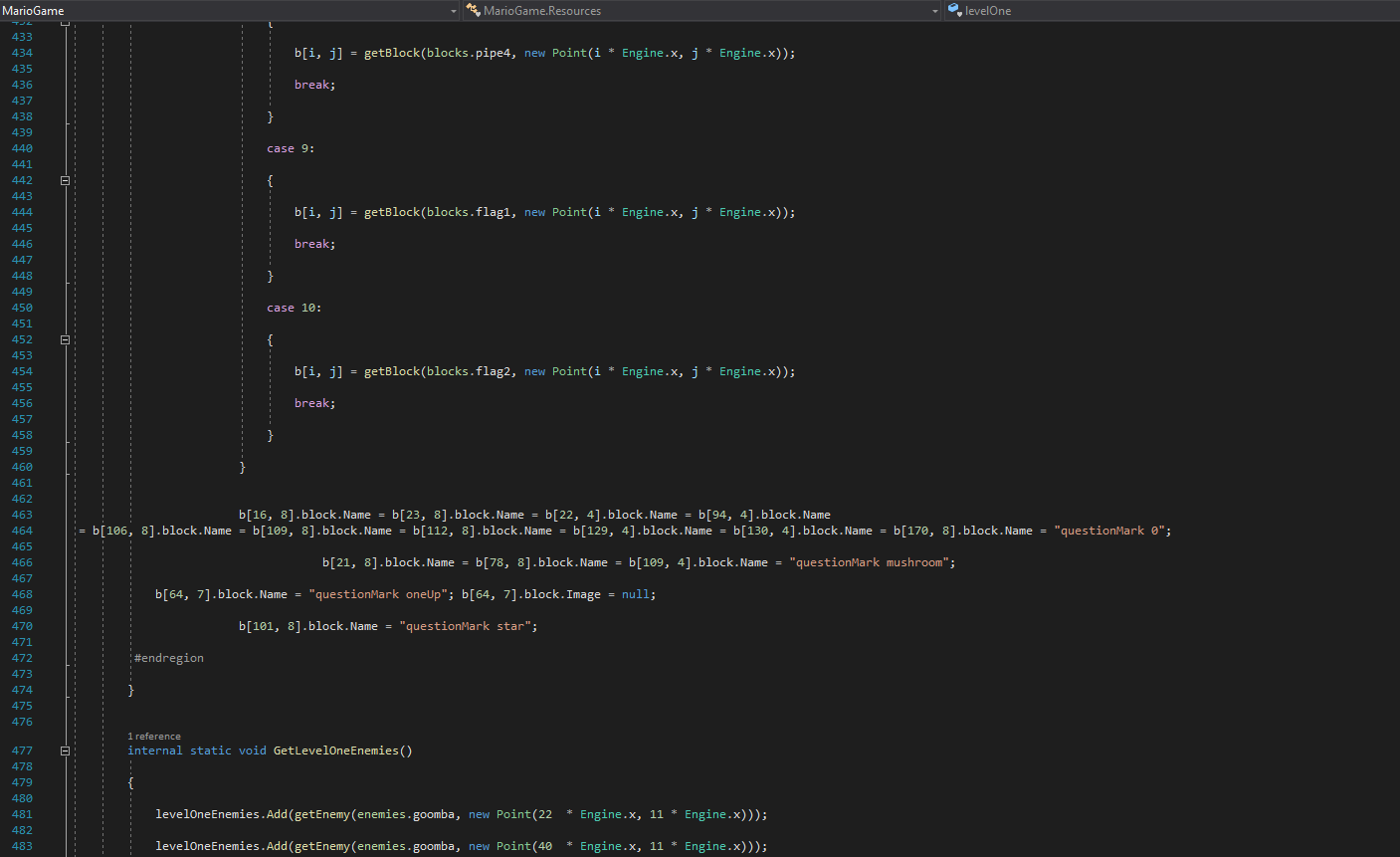
Exemplul 1: Before

****

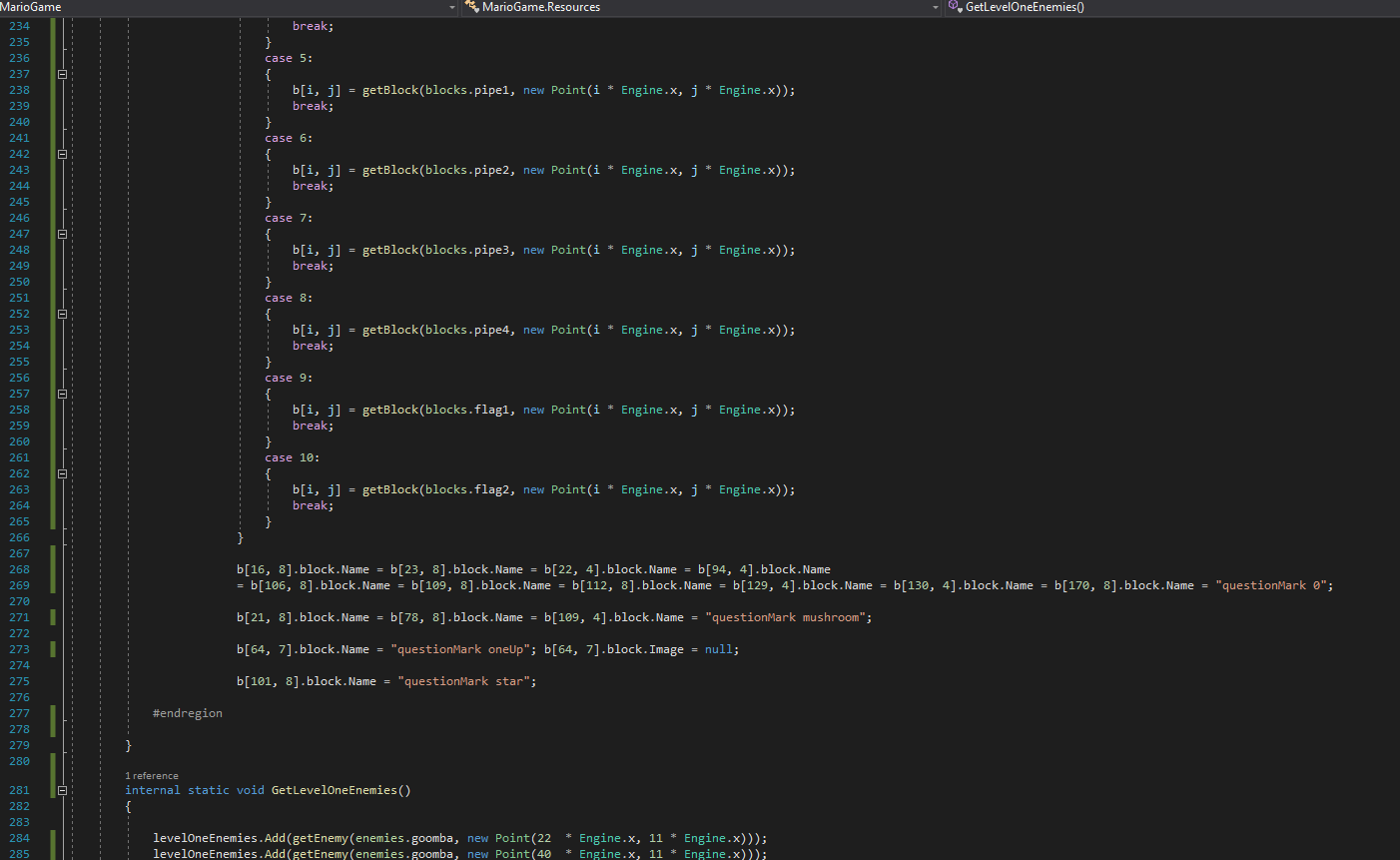
Exemplul 1: After

****

Exemplul 2: Before

****

Exemplul 2: After

****

1. Preluate de pe <https://docs.microsoft.com/en-us/dotnet/framework/get-started/system-requirements> [↑](#footnote-ref-1)
2. <https://en.wikipedia.org/wiki/Indentation_style#Allman_style> [↑](#footnote-ref-2)