Cuprins

[Capitolul 1: Consideraţii generale 4](#_Toc37157023)

[1.1 Structura lucrării 4](#_Toc37157024)

[1.2 Prezentarea aplicaţiei 4](#_Toc37157025)

[1.3 Resurse folosite 4](#_Toc37157026)

[Capitolul 2: Dezvoltarea aplicaţiei la nivelul programarii orientate pe obiect 5](#_Toc37157027)

[2.1 Clasa Engine 5](#_Toc37157028)

[2.2 Clasa Fleet 7](#_Toc37157029)

[2.3 Clasa Planet 7](#_Toc37157030)

[2.4 Clasa Player 7](#_Toc37157031)

[2.5 Clasa Ship 7](#_Toc37157032)

[Capitolul 3: Dezvoltarea aplicaţiei din punct de vedere al programării vizuale 8](#_Toc37157033)

[3.1 Formularul de start 8](#_Toc37157034)

[3.2 Formularul principal 8](#_Toc37157035)

[3.3 Formularul de luptă 8](#_Toc37157036)

[3.4 Formularul de ajutor 8](#_Toc37157037)

[Capitolul 4: Utilizarea aplicaţiei 8](#_Toc37157038)

# Capitolul 1: Consideraţii generale

* 1. Structura lucrării
  2. Prezentarea aplicaţiei
  3. Resurse folosite

# Capitolul 2: Dezvoltarea aplicaţiei la nivelul programarii orientate pe obiect

## 2.1 Clasa Engine

Această clasă are specificatorii de acces static şi public pentru a avea un caracter global. Fiecare metodă şi proprietate definită aici poate fi accesată în mod direct de toate componentele proiectului. În această clasă, de regulă, se implementează funcţii matematice, variabile globale, metode de import a resurselor ş.a.m.d.

Pentru aplicaţia curentă, am folosit următoarea definiţie a clasei Engine:

|  |
| --- |
| public static class Engine  {  public static Random rnd = new Random();  public static List<Planet> planets = new List<Planet>();  public static List<Player> players = new List<Player>();  public static Player crtPlayer;  public static int crtPlayerIDX = 0;  public static int[,] ma;  public static int n;  public static float zoomX = 1, zoomY = 1;  public static Color backColor = Color.FromArgb(ş20, 20, 50);  public static PictureBox display;  public static Graphics grp;  public static Bitmap bmp;  public static void InitGraph(PictureBox p) {...}  public static void InitDemo() {...}  public static void SelectNextPlayer() {...}  public static void Load(string fileName, string fileName2) {...}  public static void DrawMap() {...}  public static void RefreshMap() {...}  public static void BattleWave(Fleet a, Fleet b) {...}  public static void BattleCycle(Fleet a, Fleet b) {...}  public static void Battle(List<Fleet> A, List<Fleet> B) {...}  public static void RemoveDestroyedShips(Fleet A) {...}  public static void RemoveDestroyedFleets(List<Fleet> A) {...}  } |

* public static Random rnd = new Random();
* public static List<Planet> planets = new List<Planet>();
* public static List<Player> players = new List<Player>();
* public static Player crtPlayer;
* public static int crtPlayerIDX = 0;
* public static int[,] ma;
* public static int n;
* public static float zoomX = 1, zoomY = 1;
* public static Color backColor = Color.FromArgb(ş20, 20, 50);
* public static PictureBox display;
* public static Graphics grp;
* public static Bitmap bmp;
* public static void InitGraph(PictureBox p) {...}
* public static void InitDemo() {...}
* public static void SelectNextPlayer() {...}
* public static void Load(string fileName, string fileName2) {...}
* public static void DrawMap() {...}
* public static void RefreshMap() {...}
* public static void BattleWave(Fleet a, Fleet b) {...}

Această metodă se foloseşte la rezolvarea conflictului între două obiecte de tipul flotă. Metoda de rezolvare este inspirată din jocurile strategice de profil. Am considerat, aşa cum se vede în definiţia clasei Ship, 3 metode de atac şi 3 metode defensive.

|  |
| --- |
| public static void BattleWave(Fleet a, Fleet b)  {  foreach(Ship ship in a.ships)  {  int target = rnd.Next(b.ships.Count);  float attack = ship.beam - b.ships[target].shield;  if (attack < 0)  attack = 0;  attack += ship.gun - b.ships[target].armor;  if (attack < 0)  attack = 0;  attack += ship.missile - b.ships[target].pointDef;  if (attack < 0)  attack = 0;  if (attack > 0)  b.ships[target].damage += attack;  if (b.ships[target].damage >= b.ships[target].size)  b.ships[target].destroyed = true;  }  } |

* public static void BattleCycle(Fleet a, Fleet b) {...}
* public static void Battle(List<Fleet> A, List<Fleet> B) {...}
* public static void RemoveDestroyedShips(Fleet A) {...}
* public static void RemoveDestroyedFleets(List<Fleet> A) {...}

## 2.2 Clasa Fleet

## 2.3 Clasa Planet

## 2.4 Clasa Player

## 2.5 Clasa Ship

# Capitolul 3: Dezvoltarea aplicaţiei din punct de vedere al programării vizuale

## 3.1 Formularul de start

## 3.2 Formularul principal

## 3.3 Formularul de luptă

## 3.4 Formularul de ajutor

# Capitolul 4: Utilizarea aplicaţiei

**1.4 Resurse folosite**

Aplicația dispune de resurse text sub forma fișierelor \*.map din directorul aplicației \Resources\Maps.

Un fișier de tip map conține definiția unei constelații sub forma unui graf unde, în prima parte, sunt introduse coordonatele planetelor și date despre acestea (vezi figura 1.3), iar în a doua parte, avem legăturile dintre noduri.

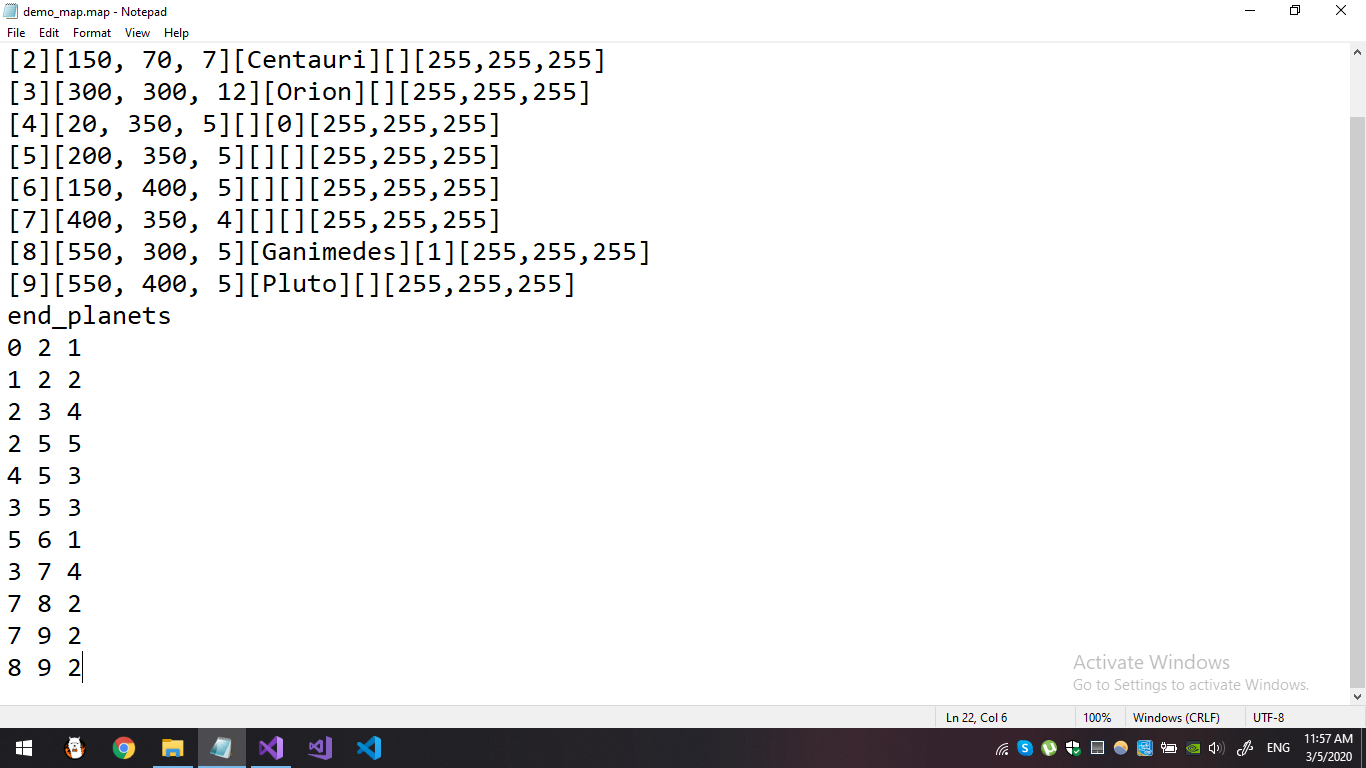


Fig. 1.3 Fișierul de resurse Demo.map