„IKT II.”  
beadandó feladat: Path to Argon

Készítette/Készítették: Simon Soma Benedek  
Osztály: 12.B

Gyakorlatvezető neve: Simon Soma Benedek

2024. január 7.

Tartalom

[Felhasználói dokumentáció 3](#_Toc410323328)

[Feladat 3](#_Toc410323329)

[Futási környezet 3](#_Toc410323330)

[Használat 3](#_Toc410323331)

[A program indítása 3](#_Toc410323332)

[A program bemenete 3](#_Toc410323333)

[A program kimenete 3](#_Toc410323334)

[Minta bemenet és kimenet 4](#_Toc410323335)

[Hibalehetőségek 4](#_Toc410323336)

[Fejlesztői dokumentáció 5](#_Toc410323337)

[Feladat 5](#_Toc410323338)

[Fejlesztői környezet 5](#_Toc410323340)

[Forráskód 6](#_Toc410323341)

[Megoldás 6](#_Toc410323342)

[Programfelépítés 6](#_Toc410323344)

[Függvénystruktúra 7](#_Toc410323345)

[A kód 8](#_Toc410323347)

[Tesztelés 10](#_Toc410323348)

[Érvényes tesztesetek 10](#_Toc410323349)

[Érvénytelen tesztesetek 11](#_Toc410323350)

[Fejlesztési lehetőségek 11](#_Toc410323351)

2. Felhasználói dokumentáció
   1. Feladat
   2. A játékban megadott és elérhető karakter egyedek közül kell választani, mellyel az adott játékosnak 10 napot kell átvészelnie amíg, el nem ér Argon városába. A napok (pályák) folyamán N helyen döntéseket kell hoznia, sikeres pályának mondjuk azt a pályát, amelyben a játékos életereje nem éri el a 0-t, ha viszont igen akkor az sikertelen és azt a napot teljesen előröl kell kezdeni. A játékos legfontosabb tulajdonságai az életerőből, a harci erejéből, a pontjai számából és az aranyai összegéből áll. Mindezen túl, a karakternek rengeteg más funkciójuk van: támadásnál a játékos, tud hárítani és támadni és ha rendelkezik gógyitófőzettel akkor, még gyógyítani is tudja magát. Bronz szinttel rendelkezik, ha a pontokszáma megegyezik vagy meghaladja az 1000 pontot, ezüsttel, ha pontokszáma megegyezik vagy meghaladja 5000 pontot, arannyal, ha megegyezik vagy meghaladja a 10000 pontot. A teljesített napok jutalommal járnak, ezek további fejlesztések a tulajdonságoknak (harcierő, penz, pontszam), emellett a játékos vásárolhat elixirt is.
   3. Futási környezet

LENOVO laptop, exe futtatására alkalmas, 64-bites operációs rendszer (Windows 10). Nem igényel egeret.

* 1. Használat
     1. A program indítása

A program az prog\_beadando\_SS\Konzoljatek\program.css néven található a tömörített állományban. Az Program.css fájl kiválasztásával indítható.

* + 1. A program bemenete

A program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *create* | Egy, új játék generálását meghatározó string változó. |
| **2.** | *valaszt* | A játékos generálását határozzák meg, ahol egyedi vagy automatikus karaktert, kell létrehozni |
| **3.** | *Nev* | A játkos nev adattagjának definiálja. |
| **4.** | *Nem* | A játkos nem adattagjának definiálja. |
| **5.** | *(2x)keppeseg* | A játékos keppeseg adattagjának definiálja. |
| **6.** | *(2x)taktika* | A játékos támadásával kapcsolatos harci taktikát kéreti be. |
| **7.** | *Input* | A játékos gyógyításához kapcsolatos döntést, tárolja el. |
| **8.** | *vasarolsz\_e* | Az elixirek vásárlását, meghatározó változó. |
| **9.** | *(2x)elixirek\_szama* | A megvásárlandó elixirek számát tárolja el (0≤N≤penz\elixir\_ar) |

* + 1. A program kimenete

A standard program első kimenete, [keppeseg] keppeség tulajdonság hiba bemenet jelzése. A második a [nap] napok számának megnevezése. A harmadik, negyedik kimenet a játékos és az ellenség életerejének kiíratása. Az ötödikben a gyógyító elixírek száma jelenik meg. A további kimenetek: a harcok kimenetele, az abból szerzett harcierő, pontszámok, gyógyítások száma, vásárolható elixírek száma, aranyak száma, valamint a legutolsó ezek közül a pontszámok általi szintbéli besorolások.

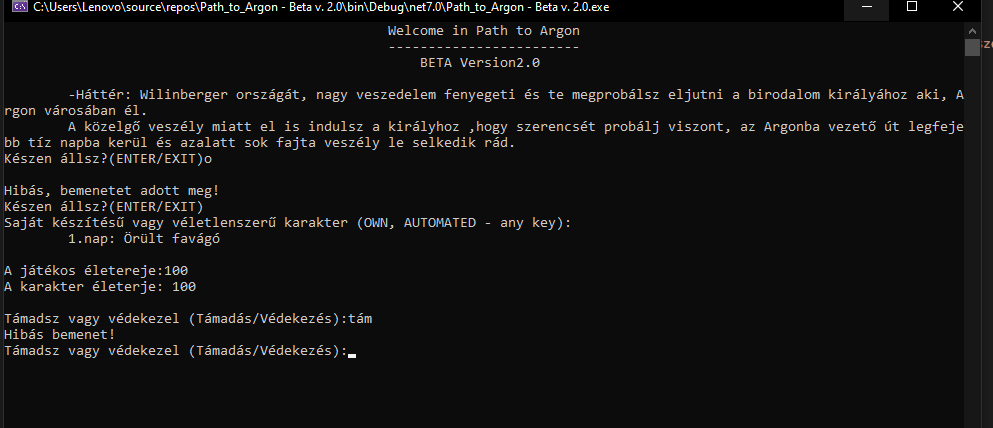
* + 1. Minta bemenet és kimenet



* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha az elixírek száma nem egész szám, vagy nem esik az **0... megvehető elixírek(**penz\elixir\_ar) intervallumba; vagy valamely mérési érték nem szám vagy szöveg. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi vagy ki jön egy hiba üzenet.

* + - 1. Mintafutás hibás bemeneti adatok esetén:



1. Fejlesztői dokumentáció
   1. Feladat
   2. A játékban megadott karakter elérhető egyedek közül kell választani, mellyel az adott játékosnak 10 napot kell átvészelnie amíg, el nem ér Argon városába. A napok (pályák) folyamán N helyen döntéseket kell hoznunk, sikeres pályának mondjuk azt a pályát, amelyben a játékos életereje nem éri el a 0-t, ha viszont igen akkor az sikertelen és azt a napot teljesen előröl kell kezdeni. A játékos legfontosabb tulajdonságai az életerejéből, a harci erejéből, a pontjai számából és az aranyai összegéből áll. Mindezen túl, a karakternek rengeteg más funkciójuk van: támadásnál a játékos, tud hárítani és támadni és ha rendelkezik gógyitófőzettel akkor, még gyógyítani is tudja magát. Bronz szinttel rendelkezik, ha a pontokszáma megegyezik vagy meghaladja az 1000 pontot, ezüsttel, ha pontokszáma megegyezik vagy meghaladja 5000 pontot, arannyal, ha megegyezik vagy meghaladja a 10000 pontot. A teljesített napok jutalommal járnak, ezek lehetnek további fejlesztések a tulajdonságoknak (harcierő, penz, pontszam), viszont tudunk vásárolhat elixirt.
   3. Specifikáció

**Bemenet**:

*create*∈string,új játék generálása.

*valaszt*∈string, új játékos generálása.

*nev*∈string, új játékos nevének megadása, NEV:Tulajdonság(nev)

nem∈string, a játékos nemének meghatározása, NEM:Tulajdonság(nem)

keppeseg∈string, a játékos képpeségeinek meghatározása, KEPPESEG:Tulajdonság(keppeseg)

*taktika*∈string, a játékos támadásának meghatározása.

*input*∈string, a játékos gyógyításához való bekérés.

*vasarolsz\_e*∈string, a játékos vásárlásához való bekérés.

*elixirek\_szama*∈string, a játékos vásárlásában megvásárlandó elixírek száma.

**Kimenet**:

nap∈ N

Ellensegek\_nevei:Tömb[nap – 1] ∈string[];

jatekos.GetEletero()∈N

ellenseg.GetEletero()∈N

jatekos.GetHarciero()∈N

jatekos.GetHarciero()∈N

jatekos.GetPontszam()∈N

jatekos.GYOGYITAS()∈N

jatekos.GetPenz()/elixir\_ar∈N

jatekos.NEV∈string

* 1. Fejlesztői környezet

LENOVO laptop, exe futtatására alkalmas operációs rendszer (pl. Windows). (v17.3) fejlesztői környezet.

Forráskód

A teljes fejlesztői anyag –kicsomagolás után– az Konzoljatek nevű könyvtárban található meg. Fej­lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| prog\_beadando\_SS\Konzoljatek\bin\Debug\Path\_to\_Argon - Beta v. 2.0.exe | futtatható kód |
|  |  |
| prog\_beadando\arviz\_feladat\Path\_to\_Argon - Beta v. 2.0.css | C# forráskód |
|  |  |
| prog\_beadando\_SS\dokumentacio.docx | dokumentációk |

* 1. Megoldás
     1. Programfelépítés

A program által használt osztályok (és helyük):

Path\_to\_Argon - Beta v. 2.0.css – program, a forráskönyvtárban  
IRandom.css - program, a forráskönyvtárban

Harc.css - program, a forráskönyvtárban

Karakterek.css - program, a forráskönyvtárban

Jatekos.css – program, a forráskönyvtárban

* + 1. A kód

A program.css fájl és a többi osztály tartalma:

//Keszitette: Simon Soma Benedek

//Kréta: 72596374630

//E-mail: simon.soma.benedek@blathy.info

//Feladat: Path\_to\_Argon

using System;

using System.Diagnostics.Tracing;

using System.Drawing;

using System.Linq.Expressions;

namespace Path\_to\_Argon\_\_\_Beta\_v.\_2.\_0

{

internal class Program

{

static void Main(string[] args)

{

Console.SetCursorPosition((Console.WindowWidth - "Welcome in Path to Argon".Length) / 2, Console.CursorTop);//Középre igazítja a szöveget.

Console.WriteLine("Welcome in Path to Argon");//Ezt, majd középre igazíthatom.

Console.SetCursorPosition(0, Console.CursorTop);

Console.SetCursorPosition((Console.WindowWidth - "Welcome in Path to Argon".Length) / 2, Console.CursorTop);//Középre igazíja a szöveget.

Console.WriteLine("------------------------");//Ezt, majd középre igazíthatom.

Console.SetCursorPosition(0, Console.CursorTop);

Console.SetCursorPosition((Console.WindowWidth - "BETA Version2.0".Length) / 2, Console.CursorTop);//Középre igazíja a szöveget.

Console.WriteLine("BETA Version2.0");

Console.SetCursorPosition(0, Console.CursorTop);

Console.WriteLine();

Console.WriteLine("\t-Háttér: Wilinberger országát, nagy veszedelem fenyegeti és te megprobálsz eljutni a birodalom " +

"királyához aki, Argon városában él.\n\tA közelgő veszély miatt el is indulsz a királyhoz ,hogy szerencsét probálj viszont, " +

"az Argonba vezető út legfejebb tíz napba kerül és azalatt sok fajta veszély le selkedik rád.");

Console.Write("Készen állsz?(ENTER/EXIT)");

string create = Console.ReadLine();

while (true)

{

if (create.ToUpper() == "ENTER" || create == "")//A jatékba való, belépés.

{

try

{

Harc harc = new Harc();

harc.Napok();

break;

}

catch (Exception ex)

{

Console.WriteLine($"Valami baj van..: {ex.Message}");

break;

}

}

else if (create.ToUpper() == "EXIT")

{

Console.WriteLine("A játék elhagyása....");

Console.Clear();

break;

}

else

{

Console.WriteLine();

Console.WriteLine("Hibás, bemenetet adott meg!");

Console.Write("Készen állsz?(ENTER/EXIT)");

create = Console.ReadLine();

}

}

}

}

using System;

using System.Collections.Generic;

using System.IO.Pipes;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Path\_to\_Argon\_\_\_Beta\_v.\_2.\_0

{

internal class Harc: Random

{

protected readonly string[] ellensegek\_nevei = { "Örült favágó", "Barbárok", "Dalnok", "Kovács", "Mike a gomba", "Kereskedő", "Szekta", "Boszorkány", "Főszekta", "Sárkány" };

//Az ellenséges karakterek nevei.

private readonly int elixir\_ar = 50;

private int nap = 1;

public void Sebzes(Karakterek ellenseg, Jatekos jatekos, string taktika)

{

while (true) {

if (Vedekezes() != 0 && taktika.ToLower() == "védekezés")

{

int ellenseg\_eletero = ellenseg.GetEletero();

int jatekos\_eletero = jatekos.GetEletero() - 2;

int vedekezes\_sebzese = Random(1,21);

ellenseg.SetEletero(ellenseg\_eletero - vedekezes\_sebzese);

jatekos.SetEletero(jatekos\_eletero);

break;

}

else if(taktika.ToLower() == "támadás")

{

int ellenseg\_tamadas = ellenseg.GetHarciero();

int jatekos\_tamadas = jatekos.GetHarciero();

int ellenseg\_eletero = ellenseg.GetEletero() - jatekos\_tamadas;

int jatekos\_eletero = jatekos.GetEletero() - ellenseg\_tamadas;

ellenseg.SetEletero(ellenseg\_eletero);

jatekos.SetEletero(jatekos\_eletero);

break;

}

else

{

Console.WriteLine("Hibás bemenet!");

Console.Write("Támadsz vagy védekezel (Támadás/Védekezés):");

taktika = Console.ReadLine();

}

}

}

public Karakterek GenerateEllenseg()

{

Karakterek ellenseg = new Karakterek();

if (nap<4)

{

ellenseg.SetEletero(100);

ellenseg.SetHarciero(Random(20,36));

ellenseg.SetPenz(Random(70,86));

ellenseg.SetPontszam(Random(800,1501));

return ellenseg;

}

else if (nap!=10)

{

ellenseg.SetEletero(150);

ellenseg.SetHarciero(Random(50,66));

ellenseg.SetPenz(Random(70, 86));

ellenseg.SetPontszam(Random(800, 1501));

return ellenseg;

}

else

{

ellenseg.SetEletero(250);

ellenseg.SetHarciero(Random(70,86));

ellenseg.SetPenz(500);

ellenseg.SetPontszam(2000);

return ellenseg;

}

}

public int Vedekezes()

{

int vedekezes = Random(1,3);//Egy véletlen számú pontszámot generál a védekezésnek.

return vedekezes;

}

public void Gyogyitas(Jatekos jatekos)

{

//A jatekos, gyógyíthatja magát, ha van neki elxirje, de akár nélkülözheti is.

if (jatekos.GYOGYITAS!=0)

{

//Bemenetet kér

Console.WriteLine($"A játékos életereje:{jatekos.GetEletero()}");

Console.Write("Szeretne gyógyítani (IGEN,NEM):");

string input = Console.ReadLine();

if (input.ToUpper() == "IGEN")

{

jatekos.SetEletero(100);

jatekos.GYOGYITAS = jatekos.GYOGYITAS - 1;

}

else if(input.ToUpper() == "NEM")

{

Console.WriteLine("Nem használtad az elixirt!");

}

else

{

Console.WriteLine();

Console.WriteLine("Hibás bemenet!");

input = Console.ReadLine();

}

}

}

//Itt, futatjuk, le a harcot, a 10 naponta megjelenő ellenséggel szemben.

public void Napok()

{

Console.Write("Saját készítésű vagy véletlenszerű karakter (OWN, AUTOMATED - any key):");

Jatekos jatekos; //A játékos objektumot, itt lesz leképezve.

//Két, fajta modón csinálhatsz, játékost...: Saját, kezűleg vagy automatikusan.

string valaszt = Console.ReadLine();

if (valaszt.ToUpper() == "OWN")

{

Console.Write("Név:");

string nev = Console.ReadLine();

Console.Write("Nem:");

string nem = Console.ReadLine();

Console.Write("Képpeségek:\n\t-Varázsló\n\t-Alkimista\n\t-Zsoldos\n\t-Paraszt\nBekérés:");

string keppeseg = Console.ReadLine();

jatekos = new Jatekos();

jatekos.GetKeppeseg(keppeseg);

}

else

{

jatekos = new Jatekos();

}

do

{

Karakterek ellenseg = GenerateEllenseg();

Console.WriteLine($"\t{nap}.nap: {ellensegek\_nevei[nap - 1]}");

Console.WriteLine();

int osszesElixir = jatekos.GYOGYITAS;

while (ellenseg.GetEletero()>0 && jatekos.GetEletero()>0)//A ciklus, aktív még egy valaki megnem hal.

{

//Eltárolja, a játkos által összedett, elixireket, ha elvesztené a csatát.

Console.WriteLine($"A játékos életereje:{jatekos.GetEletero()}");

Console.WriteLine($"A karakter életerje: {ellenseg.GetEletero()}");

Console.WriteLine();

Console.Write("Támadsz vagy védekezel (Támadás/Védekezés):");

string taktika = Console.ReadLine();

Sebzes(ellenseg, jatekos, taktika);

if (ellenseg.GetEletero() > 0 && jatekos.GetEletero() > 0)

{

Console.WriteLine();

Console.WriteLine($"Gyógyító elixir: {jatekos.GYOGYITAS}");

Console.WriteLine();

Gyogyitas(jatekos);

}

}

if (jatekos.GetEletero()>0 && ellenseg.GetEletero()<=0)

{

Console.WriteLine();

Console.WriteLine($"Legyőzted a {ellensegek\_nevei[nap - 1]}");

nap++;

Lada(jatekos, ellenseg);// A játékos objektum harci tulajdonságait, fejleszti.

Vasarlas(jatekos);// Növelheti, a játékos GYOGYITAS adattaghoz.

}

else

{

Console.WriteLine("Elvesztetted a csatát!");//Újra, índitja a csatát!

jatekos.SetEletero(100);

ellenseg = GenerateEllenseg();

jatekos.GYOGYITAS = osszesElixir;

}

}while (nap<=10);

int jatekosPontszama = jatekos.GetPontszam();

Console.WriteLine(jatekosPontszama);

if (jatekosPontszama>=10000)//Játékos pontszámának, értékelése.

{

Console.WriteLine();

Console.WriteLine($"Gratulálok {jatekos.NEV} elérted a GOLD-szintet!");

}

else if (jatekosPontszama>=5000)//Játékos pontszámának, értékelése.

{

Console.WriteLine();

Console.WriteLine($"Gratulálok {jatekos.NEV}elérted a SILVER-szintet!");

}

else if(jatekosPontszama>=1000)//Játékos pontszámának, értékelése.

{

Console.WriteLine();

Console.WriteLine($"Gratulálok {jatekos.NEV} elérted a Bronz-szintet!");

}

}

public void Lada(Jatekos jatekos, Karakterek ellenseg)//Megmutatja, a harci fejlesztéseket.

{

jatekos.SetPontszam(jatekos.GetPontszam() + ellenseg.GetPontszam());

jatekos.SetPenz(jatekos.GetPenz() + ellenseg.GetPenz());

jatekos.SetEletero(100);

jatekos.SetHarciero(jatekos.GetHarciero() + Random(10, 21));

Console.WriteLine();

Console.WriteLine($"Kaptál egy ládát!\nHarcierőd értéke:{jatekos.GetHarciero()}");

Console.WriteLine($"Az általad elért pontok száma:{jatekos.GetPontszam()}");

Console.WriteLine($"Aranyaid száma:{jatekos.GetPenz()}");

}

public void Vasarlas(Jatekos jatekos)//A metodusban, vásároluhatunk elixirt.

{

if (jatekos.GetPenz() >= elixir\_ar && nap<=10) //Megnézzi, a jatekosnak van-e elég pénze rá.

{

Console.WriteLine();

Console.Write("Vásárolsz-e elixirt(IGEN/NEM - any key):");

string vasarolsz\_e = Console.ReadLine();

if (vasarolsz\_e.ToUpper() == "IGEN")

{

Console.WriteLine();

Console.WriteLine($"A pénzedből ennyi elixirt vehetsz:{jatekos.GetPenz()/elixir\_ar}");

int elixirek\_szama; //Hány, elixirt tudna venni.

if (jatekos.GetPenz()/elixir\_ar > 1)//Első bemenet: Ha, többet tudsz venni.

{

Console.Write($"Hány elixirt kérsz:");

elixirek\_szama = int.Parse(Console.ReadLine());

while (elixirek\_szama> jatekos.GetPenz() / elixir\_ar && elixirek\_szama < jatekos.GetPenz() / elixir\_ar)//Ha, nem add, meg jó értéket.

{

Console.WriteLine("Ennyit, nem tudsz venni!");

Console.Write("Probáld, újra:");

elixirek\_szama = int.Parse(Console.ReadLine());

}

jatekos.GYOGYITAS = jatekos.GYOGYITAS + elixirek\_szama;

jatekos.SetPenz(jatekos.GetPenz() - elixirek\_szama \* elixir\_ar);

}

else//Második bemenet: Ha, csak egyet tudsz venni.

{

elixirek\_szama = 1;

jatekos.GYOGYITAS = jatekos.GYOGYITAS + elixirek\_szama;

jatekos.SetPenz(jatekos.GetPenz() - elixirek\_szama \* elixir\_ar);

}

}

}

}

public int Random(int start, int end)

{

Random random = new Random();

return random.Next(start, end);

}

}

}

using System;

using System.Collections.Generic;

using System.Diagnostics.Tracing;

using System.Linq;

using System.Linq.Expressions;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

namespace Path\_to\_Argon\_\_\_Beta\_v.\_2.\_0

{

internal class Jatekos : Karakterek

{

public string NEV { get; set; }

private string NEM { get; set; }

private string KEPPESEG { get; set; }

public int GYOGYITAS = 0;

private readonly string[] nevek = new string[4] { "Guest001", "Guest002", "Guest003", "Guest004" };

private readonly string[] nemek = new string[2] { "Férfi", "Nő" };

private readonly string[] keppesegek = new string[4] { "Alkimista", "Varázsló", "Zsoldos", "Paraszt" };

public Jatekos()

{

Generate\_player();

}

public Jatekos(string nev, string nem, string keppeseg){

NEV = nev;

NEM = nem;

KEPPESEG = keppeseg;

}

public void Generate\_player()

{

int nev\_valaszt = Random(0, nevek.Length);

int nem\_valaszt = Random(0, nemek.Length);

int keppesegek\_valaszt = Random(0, keppesegek.Length);

NEV = nevek[nev\_valaszt];

NEM = nemek[nem\_valaszt];

KEPPESEG = keppesegek[keppesegek\_valaszt];

GetKeppeseg(KEPPESEG);

}

public void GetKeppeseg(string keppeseg)

{

while (true)

{

if (keppeseg == "Alkimista")

{

SetEletero(100);

SetHarciero(35);

break;

}

else if (keppeseg == "Varázsló")

{

SetEletero(100);

SetHarciero(60);

break;

}

else if (keppeseg == "Zsoldos")

{

SetEletero(100);

SetHarciero(70);

break;

}

else if (keppeseg == "Paraszt")

{

SetEletero(100);

SetHarciero(80);

break;

}

else

{

Console.WriteLine();

Console.WriteLine("Hibás, bemenet!");

Console.Write("Képpeségek:\n\t-Varázsló\n\t-Alkimista\n\t-Zsoldos\n\t-Paraszt\nBekérés:");

keppeseg = Console.ReadLine();

}

}

}

}

}

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Path\_to\_Argon\_\_\_Beta\_v.\_2.\_0

{

internal class Karakterek : Random

{

private int eletero { get; set; }

private int harciero { get; set; }

private int penz { get; set; }

private int pontszam = 0;

public void SetEletero(int Eletero)

{

eletero= Eletero;

}

public int GetEletero()

{

return eletero;

}

public void SetHarciero(int Harciero)

{

harciero= Harciero;

}

public int GetHarciero()

{

return harciero;

}

public void SetPontszam(int Pontszam)

{

pontszam = Pontszam;

}

public int GetPontszam()

{

return pontszam;

}

public void SetPenz(int Penz)

{

penz = Penz;

}

public int GetPenz()

{

return penz;

}

public int Random(int start, int end)

{

Random random = new Random();

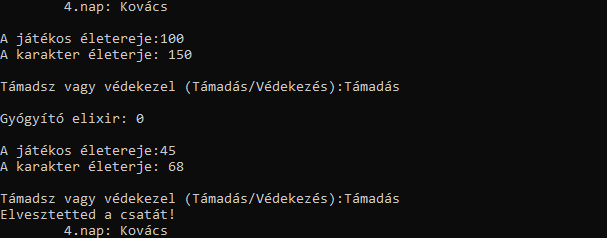
return random.Next(start, end);

}

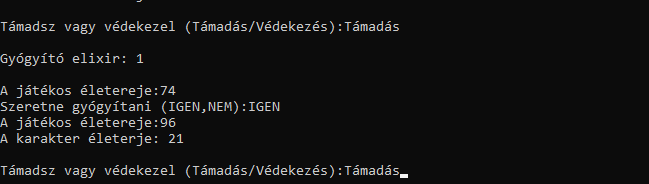
}

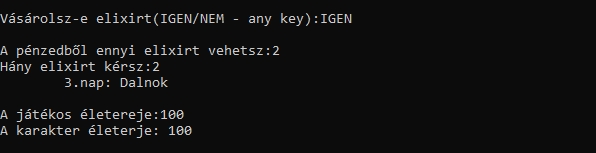
}

* 1. } Tesztelés
     + 1. Teszt: A játékos legyőzése után, újrakezdett pálya/nap



* + - 1. Teszt: A játékos gyógyítása, és vásárlási lehetőségei.





Fejlesztési lehetőségek

1. Adatok –a felhasználó igénye szerint– akár fájlból is fogadása.
2. Kód béli szintaktika és szemantika leegyszerűsítése, feljavítása, olvashatóbbá alakítása.
3. Hibás konzolos bemenetek kezelése, valamint, bizonyos bemenetek reagálásának megváltoztatása.
4. Felhasználóbb felület: A bemeneti sorozat grafikus megjelenítése, s a játékosok tulajdonságainak folyamatos kijelzése.