



Eötvös Loránd Tudományegyetem

Informatikai Kar

Komputeralgebra Tanszék

Gépi tanulás a legnagyobb sebzés eléréséhez

Dr. Vatai Emil

Adjunktus

Komputeralgebra tanszék

ELTE IK

Szalóki Sándor

Programtervező informatikus BSc.

Budapest, 2018

Tartalomjegyzék

1. Bevezető	2
2. Felhasználói dokumentáció	3
2.1. Feladat	3
2.1.1. Játék modell	3
2.1.2. Gépi tanulás	4
2.1.3. Grafikus felület	4
2.2. Módszerek	5
2.2.1. Játék model	5
2.2.2. Gépi tanulás	5
2.2.3. Grafikus felület	6
2.3. Rendszer-követelmények, telepítés	6
2.4. A program használata	7
3. Fejlesztői dokumentáció	12
3.1. Specifikáció	12
3.2. Módszerek	12
3.2.1. Fogalomtár	12
3.2.2. Játék model	12
3.2.3. Gépi tanulás	12
3.2.4. Grafikus felület	12
3.3. Szerkezet	12
3.3.1. Külső függőségek	12
3.3.2. Adatszerkezetek	12
3.3.3. Modulok	12
3.3.4. Képességek	12
3.4. Tesztelés	12

1. Bevezető

Legfőbb célom a szakdolgozattal a gépi tanulás mélyebb megértése, tanulása egy olyan példán keresztül, amely előfordulhat problémaként a való életben is.

A probléma, amivel itt foglalkozunk kellően bonyolult ahhoz, hogy a megoldásához gépi tanulást kelljen alkalmazni és megoldása ne legyen triviális egy egyszerű algoritmussal.

Ahhoz, hogy a feladat több legyen, mint egy egyszerű út keresés, minden lépés hatással van a környezetére, illetve a környezettől függően egy-egy lépés hatása, értéke változhat.

Erre egy kellő nehézségű probléma a Final Fantasy XIV harcrendszerének szimulálása, majd ebben a szimulációban való tanulás.

2. Felhasználói dokumentáció

2.1. Feladat

2.1.1. Játék modell

Első feladat a játéknak egy kellő pontosságú modellezése. A megoldás hatékonysága érdekében törekedni kell arra, hogy ne legyen felesleges adat, viszont minden, ami szükséges, az elérhető legyen. A harcrendszer szabályai pontosan meghatározhatóak, így a szimulációtól elvárható a pontos eredmény is. Két fő részre osztható a modell: Karakterek és ezek képességei.

Karakterek Minden karakternek három meghatározó eleme van:

Gyorsaság Ez adja meg a karakter globális időzítőjét

Támadás A képesség, ami a játékos irányítása nélkül történik a karakter gyorsaságától függően

MP/TP Magic Points, illetve Tactial Points, ezek a képességek erőforrásai

Képességek Egy képesség leírásához hét tulajdonság szükséges:

Potenciál A sebzés mértékének alapja

Időzítő Meghatározza, hogy az adott képesség a globális időzítőt, vagy a sajátját használja

Erőforrás A használatának ára MP, vagy TP-ben

Combo Képességek adott sorrendben használata esetén más-más lehet egy képesség hatása

Típus Fegyver, vagy varázslat

Feltétel A használatára vonatkozó feltétel

Hatás Használat után adott ideig hatással lehet a következő képességére, a hős állapotára, vagy adott időközönként plusz sebzést okozhat

A karakter tulajdonságai és képességeinek listája alapján egy adott képesség sorozatból készíthető egy szimuláció, amely megadja, hogy a sorozatot hány tizedmásodperc végrehajtani, illetve végrehajtás után ennek mekkora a teljes sebzési értéke. Fontos, hogy a szimuláció bármely pontján tudjuk, hogy mi a játékos pontos állapota: Erőforrásainak, globális időzítőjének, képességeinek saját időzítőjének állapota, az aktív hatások, illetve ha van, akkor a Combo álltal meghatározott következő képesség.

Ez az állapot megfelel annak, amit a játékos játék közben a képernyőn láthatna.

2.1.2. Gépi tanulás

A feladat ebben a szimulációban egy olyan képesség sorozat keresése, amely eléri a felhasználó által megadott időt, és ezalatt az idő alatt minél hatékonyabb legyen, azaz a legnagyobb sebzés elérésére törekszik.

A probléma tér elég nagy ahhoz, hogy egy egyszerű kereső algoritmus futási ideje túl nagy legyen. Ennek megoldásához megerősítéses tanulást használ a program.

Ehhez szükséges, hogy a játék állapotáról egy kellően pontos képet kapjunk, erre szolgál a játék szimulációjából tudható képszerű adat. Ezt az adatot állapotnak, a képességeket pedig akcióknak használhatjuk fel.

2.1.3. Grafikus felület

Szükséges egy egyszerű grafikus felület, amely megkönnyíti a program használatát, átláthatóságát. A képességek sorozatának egy megjelenítését, a gépi tanulás eredményének grafikus feldolgozását.

Az algoritmus eredményessége jól bemutatatható néhány egyszerű grafikonon.

2.2. Módszerek

A feladat megoldása funkcionális programozással történik.

2.2.1. Játék model

Az említett két fő elemet (Karakterek és Képességek) megfelelő típusokkal vesszük, majd egy-egy tetszőleges karaktert a képességeivel együtt már egyszerűen előállíthatunk. Miután kellő adat birtokában vagyunk, a képességekből összeállíthatunk egy sorozatot, amit szimulálni szeretnénk.

A játék harcrendszerének szimulációja három lépésben történik.

1. A képességek sorozatát validálni kell. Ellenőrizni kell, hogy a képesség időzítője szerint lehetséges-e már a használata, rendelkezésre áll-e megfelelő típusú és mennyiségű erőforrás. Illetve néhány képesség rendelkezhet saját feltétellel, amit szintén ellenőrizni kell. Amennyiben minden rendelkezésre áll, a képességet meg kell tartani és a megfelelő hatásokat, illetve időzítőket be kell állítani.
2. A validált sorozaton végig kell menni és minden ponton az adott állapot szerint ki kell számolni, hogy abban a tizedmásodpercen mennyi sebzés történt, majd ezeket összegezni.

2.2.2. Gépi tanulás

A megfelelő algoritmus választásához több tényezőt is figyelembe kell venni. A játék harcrendszerének teljes működése kellően bonyolult, így a működést kihasználó algoritmusok használata bonyolult, ha lehetséges. Az akciók értéke az állapottól függ, illetve az akciók változtathatják az állapotot, emiatt az a feladat nem vezethető vissza egy egyszerű többkarú rabló problémára sem.

A probléma megoldásához Monte-Carlo-módszer használható. Az algoritmusnak nincs szüksége a háttérben működő szimuláció pontos leírására, elég ha tudunk adni egy pontos képet az állapotról, illetve az adott állapotban választott akciók értékéről. A szimulációt olyan módon működik, hogy megfelelő képet tudjon adni az állapotról, az akció értéke pedig megadható az képesség sebzésének függvényében.

2.2.3. Grafikus felület

A program nem igényel bonyolult grafikus felületet. Egyszerű MVVM architektúra használatával épül fel WPF segítségével. Karakterenként a kilistázzuk a képességek képeit, amelyek majd választhatóak lesznek szimulációra. Lehetőséget kell adni az eredmény sorozatnak kilistázására, illetve grafikonok rajzolására. Fontos, hogy a számolás a háttérben fusson, hiszen ez egy idő után nagy számításokkal járhat, ami zárolhatja az ablakot.

2.3. Rendszer-követelmények, telepítés

A program futtatásához Windows operációs rendszer szükséges. Telepítés nem szükséges, viszont a futtatható állomány mellé el kell helyezni a (programmal együtt kiadott) dinamikus könyvtárakat is.

Ajánlott hardware: Quad-Core 2.3 Ghz processzor, 2 GB RAM

Software követelmény: Microsoft .NET Framework 4.6.1

2.4. A program használata

A program indításához a futtatható állomány szükséges. Ennek indításával rögtön egy ablak fogad (1). Az ablakon található egy menüsor rendre Add, Start, illetve Cancel gombokkal. Alatta található a program futásához szükséges beállítások mezői. Egy listában találhatóak a karakterek és hozzájuk rendelt képességek listája. Az ablak közepén kerül majd megjelenítésre a tanult képesség sorozat, majd alatta a két grafikon. Továbbá található még egy státusz sor a képernyő legalján. Ebben a következő információk találhatóak:

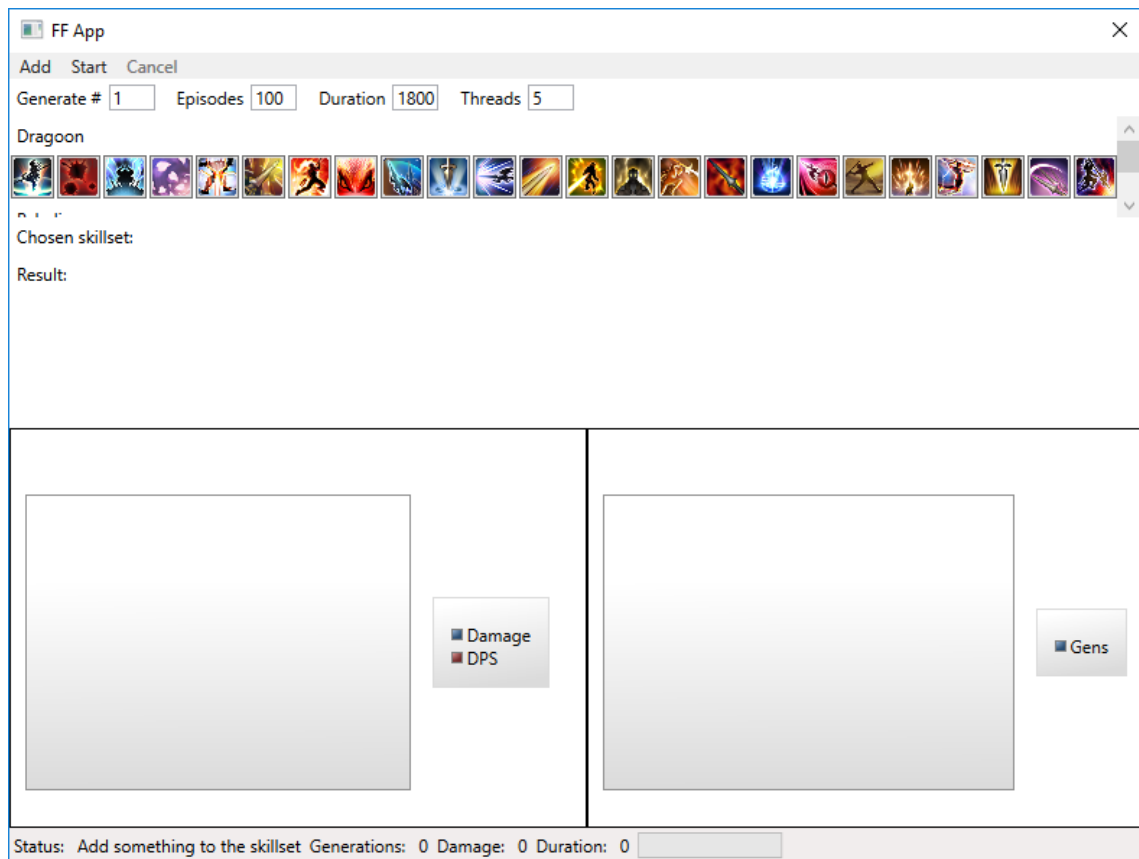
Status Rövid üzenet a program állapotáról

Generations A jelenlegi generáció száma

Damage A tanult képesség sorozat által elért sebzés értéke

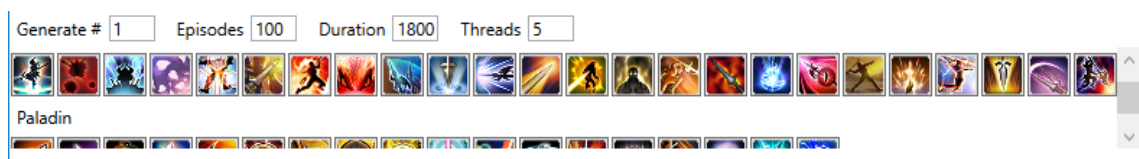
Duration A tanult képesség sorozat által elért idő tizedmásodpercben

Bar Töltő csík, amely mutatja, ha éppen számolás történik a háttérben



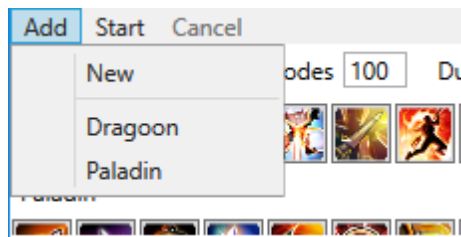
1. ábra. Kezdő képernyő

Mielőtt elindíthatnánk a tanulást, választanunk kell egyet a megadott karakterek képességlistái közül (2). Minden képesség lista fölött megtalálható a hozzá tartozó karakter neve. A kurzor egy képességen tartása után megjelenik a képesség neve, rövid leírása.



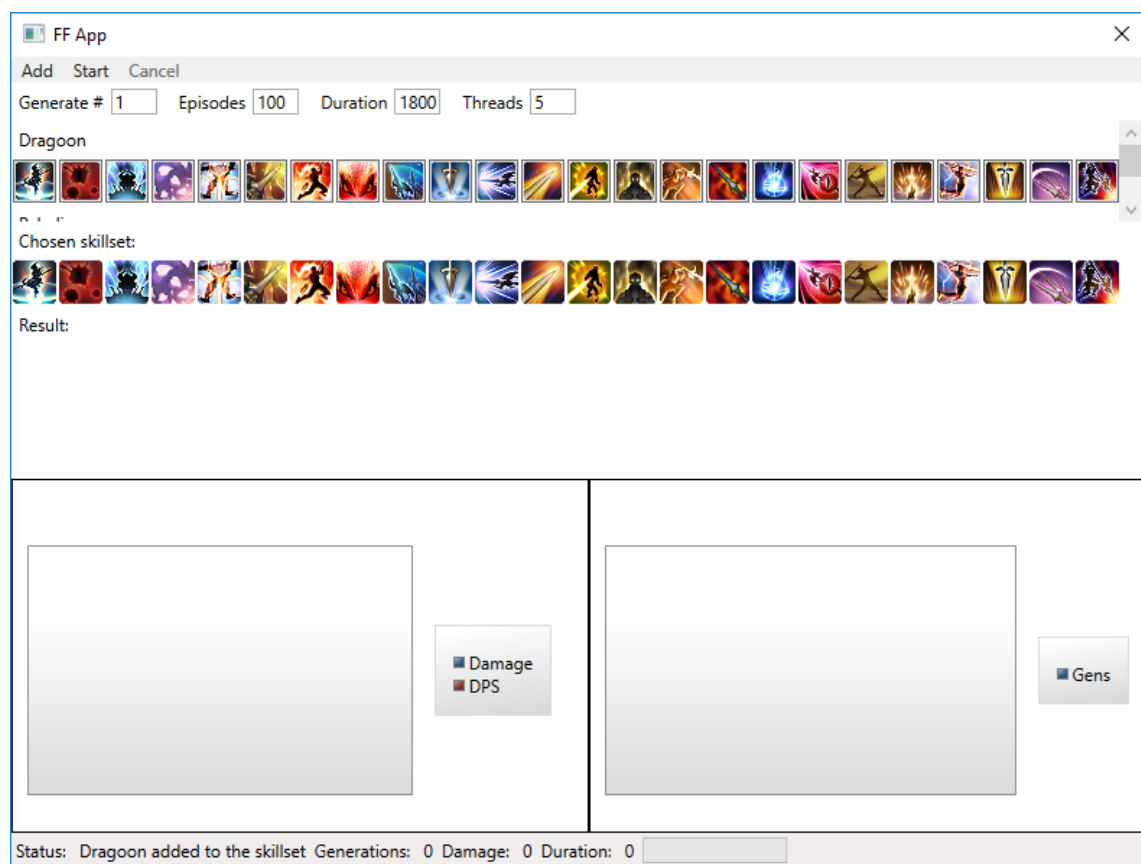
2. ábra. Lehetséges választható listák

Ha megtaláltuk a megfelelő listát, akkor az Add menüpontban (3) választhatjuk ki a szükséges listát név alapján. Fontos, hogy ilyenkor még szabadon cserélhetjük a karaktert, ez a menüpont Start-ra kattintás után nem lesz előrhető. (Kivéve a New gomb)



3. ábra. Lehetséges választható listák az Add menüpont alatt

A lista kiválasztása után erről információt kapunk, a választott lista megjelenik a Chosen skillset alatt (4). Ezen a ponton a tanulás elindítható a Start gombbal.



4. ábra. A választott lista megjelenik a képernyőn

Indítás előtt lehetőségünk van néhány beállítás változtatására (5).



5. ábra. Futási beállítások

A beállítások sorra:

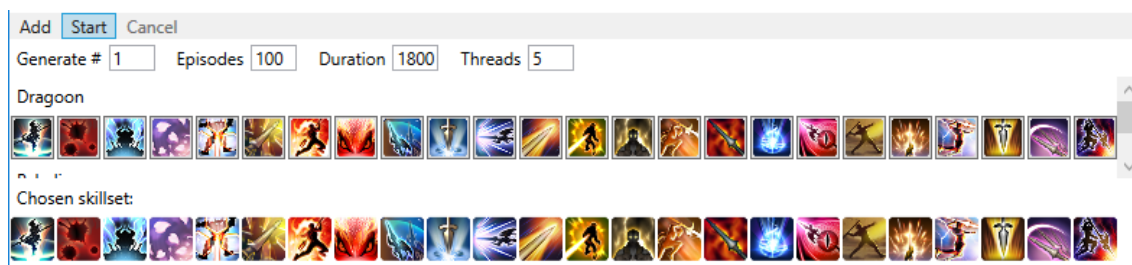
Generate A kívánt generációk számát adhatjuk meg

Episodes Generációnként létrehozott szimulációk száma

Duration Szimulációk hossza tizedmásodpercben

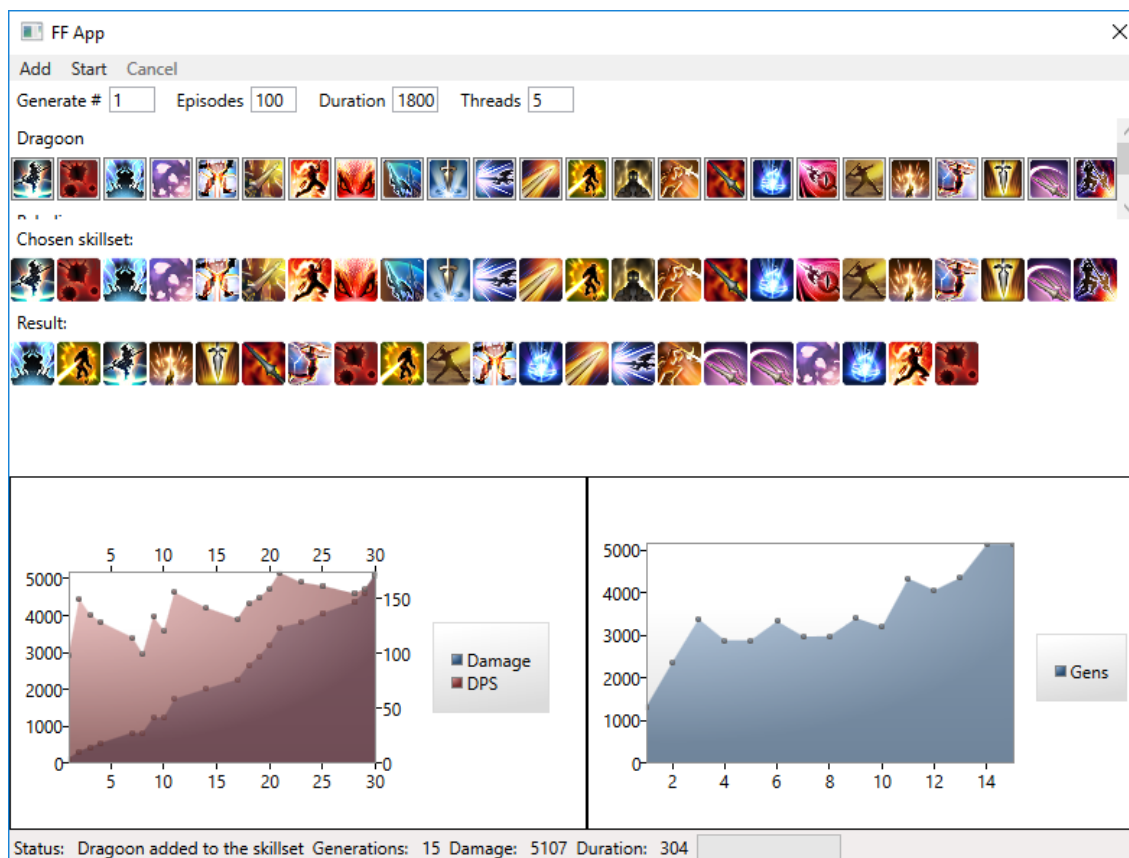
Threads Párhuzamosítás során használható magok száma

Amennyiben a beállításokat megfelelőnek találjuk, a Start gomb (6) lenyomásával elindítható a beállított számú generációk generálása.



6. ábra. Generálások indítása

A program a generációkat egyesével állítja elő és rajzolja őket ki a képre (7). Eközben a beállítási mezők, illetve az Add és Start gombok nem elérhetőek, viszont a generálás bármikor leállítható a Cancel gombra kattintva. Leállítást kérése esetén az éppen futó számolás még befejeződik, ennek eredménye megjelenik a képernyőn, majd az eddig letiltott mezők, gombok feloldásra kerülnek.



7. ábra. Eredmény 15 generáció futtatása után

Az eredményről három féle információt kapunk.

Result A jelenlegi tudás alapján generált képesség sorozat

Bal grafikon A Result-ban szereplő képesség sorozat szimulálása során kapott sebzés értékek másodpercre osztott értékei. Kék színnel és Damage névvel jelölt terület mutatja az adott másodpercben elért teljes sebzést. Piros színnel és DPS névvel jelölt terület mutatja az adott másodpercben elért sebzés és idő hányadosát.

Jobb grafikon Nyomon követhetjük minden generáció tudásának sebzés értékét

Amennyiben újra szeretnénk kezdeni a tanulás folyamatát, az Add menüpont alatt (3) található egy New gomb, amivel alaphelyzetre állíthatjuk a programot. Ilyenkor a kiválasztott képesség lista nem változik, ennek változtatásához a programot újra kell indítani.

3. Fejlesztői dokumentáció

3.1. Specifikáció

3.2. Módszerek

3.2.1. Fogalomtár

3.2.2. Játék model

3.2.3. Gépi tanulás

3.2.4. Grafikus felület

3.3. Szerkezet

3.3.1. Külső függőségek

3.3.2. Adatszerkezetek

3.3.3. Modulok

3.3.4. Képességek

3.4. Tesztelés