

HÁZI FELADAT

Programozás alapjai 2.

Tervezés

Kovács Donát

BYVO90

2023. április 15.

Tartalom

1. Feladat	2
2. Feladatspecifikáció	2
2.1 Játékszabályok	2
2.2 Játékosok	2
2.3 Menü	3
2.4 Játékmenet	3
3. Terv	4
4. Algoritmusok	6

1. Feladat

Kiterjesztett kő-papír-olló játék

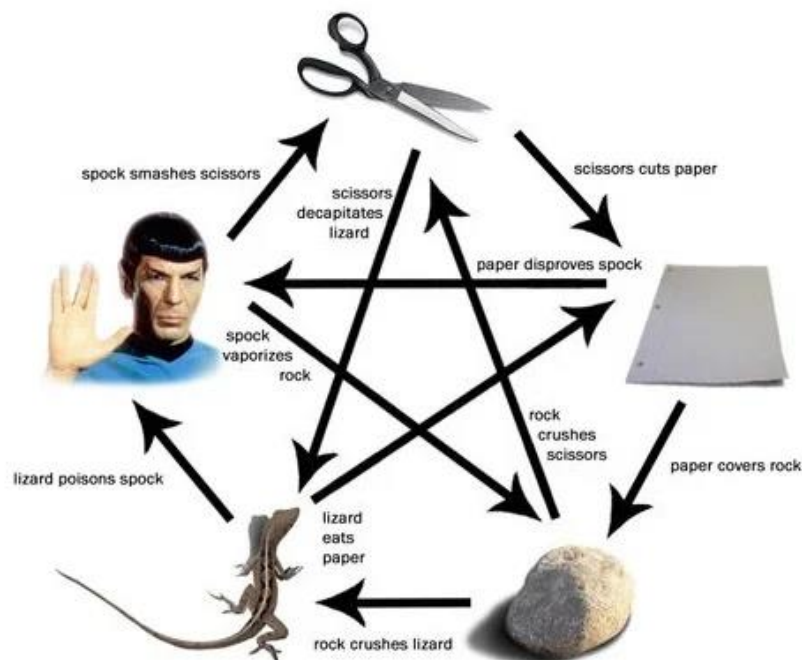
Tervezzon objektummodellt a kő-papír-olló játék modellezéséhez! Célunk, hogy a különböző stratégiával játszó játékosokat összesorsolva megállapítsuk a legjobb stratégiát, ha van ilyen. A modellben legyenek "Játékos" objektumok, melyek egy "Napló" objektum felügyeletével játszanak. Ez utóbbi gyűjti a statisztikát. Demonstrálja a működést külön modulként fordított tesztprogrammal! A játék állását nem kell grafikusán megjeleníteni, elegendő csak karakteresen, a legegyszerűbb formában! A megoldáshoz **ne** használjon STL tárolót!

2. Feladatspecifikáció

2.1 Játékszabályok

A kő-papír-ollót két játékos játssza. A játékosok egyszerre választanak ki egy-egy alakzatot (kő, papír vagy olló). A játék célja, hogy olyan alakzatot válasszunk, amely kiüti a másik játékos által választottat. A kő kicsorbítja az ollót, az olló elvágja a papírt, a papír becsomagolja a követ. Ha mindketten ugyanazt választották, akkor a kör eredménye döntetlen.

A játék kiterjesztett változata a kő-papír-olló-gyík-Spock, melynek lehetséges győzelmi kimenetei a következők:



A játéknak akkor van vége, ha az előre megbeszélt mennyiségű győzelmet valaki eléri.

2.2 Játékosok

A játékot két játékos játszhatja, amelyek közül bármely lehet ember vagy számítógép. A számítógép esetében több stratégiát (pl.: random, az ellenfél előző lépéseit felhasználó, stb.) lehet kiválasztani. Alapértelmezetten ember játszik a gép random stratégiája ellen.

2.3 Menü

A program angol nyelven és karakteres módban működik.

1. Play

2. Options

1. Game Type – kő-papír-olló vagy kő-papír-olló-gyík-Spock
(alapértelmezett: kő-papír-olló)
2. Game Mode – Step by step, Simulation vagy Tournament
(alapértelmezett: Step by step)
3. Game length – meddig tartson egy játék
(alapértelmezett: 20 győzelemig)
4. Player 1 Name: Player 1 – első játékos nevének beállítása, nincs megjelenítve
tournament módban (alapértelmezett: Player 1)
5. Player 1 Type: Human – első játékos típusának beállítása, nincs megjelenítve
tournament módban
6. Player 2 Name: Player 2 – második játékos nevének beállítása, nincs megjelenítve
tournament módban (alapértelmezett: Player 2)
7. Player 2 Type: Computer Strategy 1 – második játékos típusának beállítása, nincs
megjelenítve tournament módban
0. Main Menü – visszalépés a főmenübe

3. Rules – játékszabályok megjelenítése

0. Exit – kilépés a játékból

2.4 Játékmenet

Step by step

Ebben a játékmódban egy játékot lehet egyesével léptetve végigjátszani. Ez az egyetlen mód, ahol játszhat ember a számítógép ellen. A program minden lépésnél bekéri a játékostól az alakzat számát. Ha pedig számítógép játszik, akkor az Enter megnyomására történik a lépés.

Simulation

Ebben a játékmódban lehet léptetés nélkül, automatikusan végigfuttatni egy teljes játékot. A szimulációban csak számítógép játszhat számítógép ellen.

Tournament

Ebben a játékmódban lehet léptetés nélkül szimulálni különböző számítógép stratégiák teljes bajnokságát és erről egy statisztikát megjeleníteni.

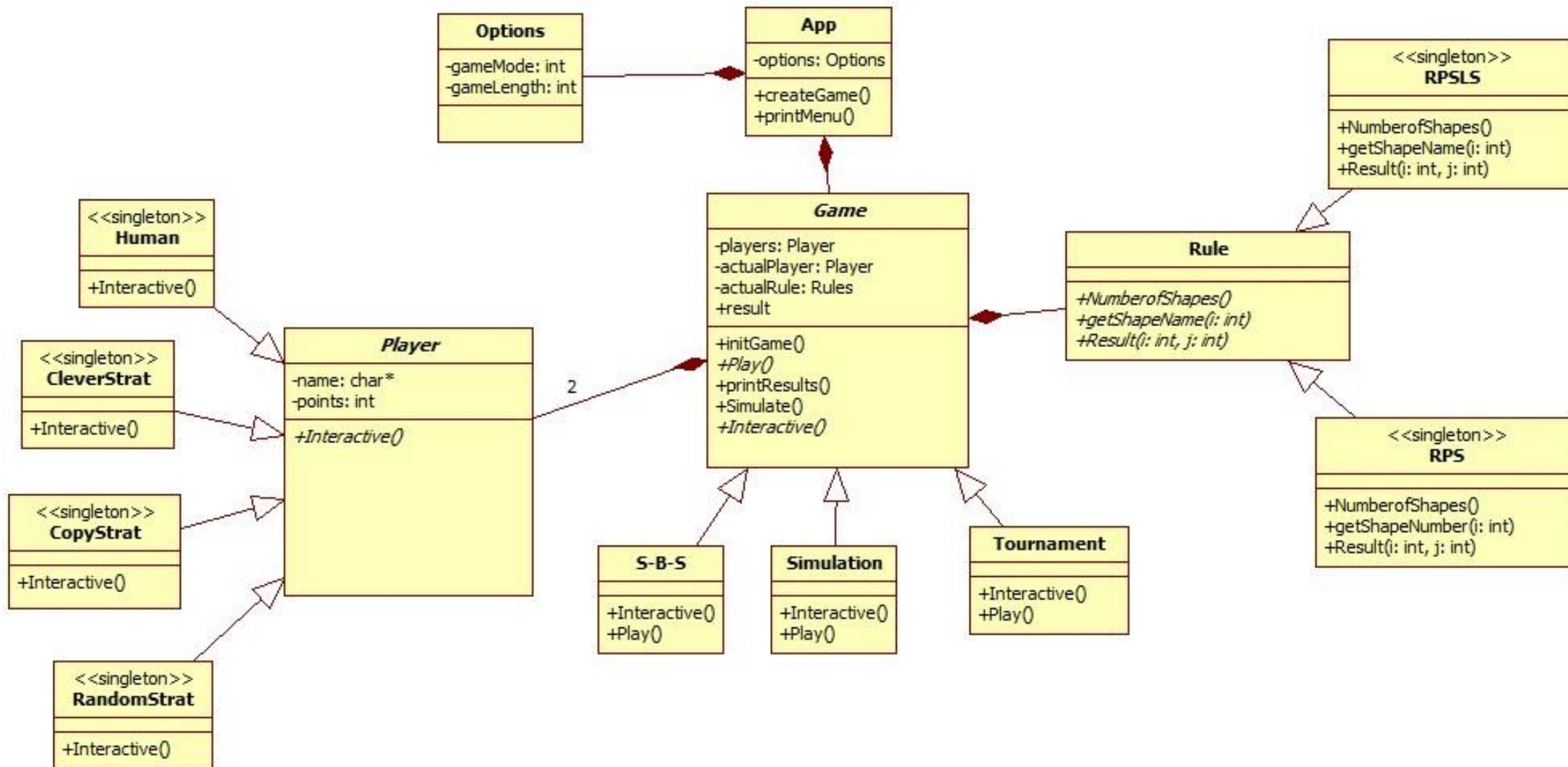
A játékmenet ha sok döntetlen miatt eléri a 100. játszmat, akkor ott vége lesz és az addig több győzelmet szerzett játékos nyeri a menetet. Ha viszont ugyanannyi győzelmük van, akkor a kimenetel döntetlen lesz.

A program a későbbiekben újabb algoritmusokkal tetszőlegesen bővíthető.

3. Terv

A feladat egy program objektum orientált tervezését mutatja be UML diagramban.

A program az alábbi osztályokat, azok kapcsolatát és attribútumait tartalmazza:



4. Algoritmusok

A feladatban lévő három stratégia algoritmusainak leírása:

Random Strategy:

Random számot generál 0-tól 2-ig vagy 0-tól 4-ig (a játéktípustól függően) és azt a számot adja vissza az algoritmus, amely egy alakzatnak a számát jelöli.

Copy Strategy:

Első lépésnél random generál számot 0-tól 2-ig vagy 0-tól 4-ig (a játéktípustól függően) és azt a számot adja vissza, majd utána az algoritmus minden lépésnél eltárolja az ellenfele alakzatának a számát és a következő lépésben azt adja vissza.

Clever Strategy:

Első lépésnél random generál számot 0-tól 2-ig vagy 0-tól 4-ig (a játéktípustól függően) és azt a számot adja vissza, majd a második lépéstől kezdve utolsó lépéskombináció gyakoriságának a kiszámolásával adja vissza azt az alakzatot, amely az ellenfél legnagyobb valószínűséggel bekövetkező lépését legyőzi.

A tournamentben lévő játékosok teljesítményét pontozó algoritmus leírása:

Az algoritmus egy szimulált játékmenet után a győztesnek 2 a vesztesnek 0 pontot ad. Döntetlen kimenetel esetén mindkét játékos 1-1 pontot kap. Így a bajnokság végén a pontok által statisztikát kapunk az adott stratégiák nyeresi valószínűségéről.