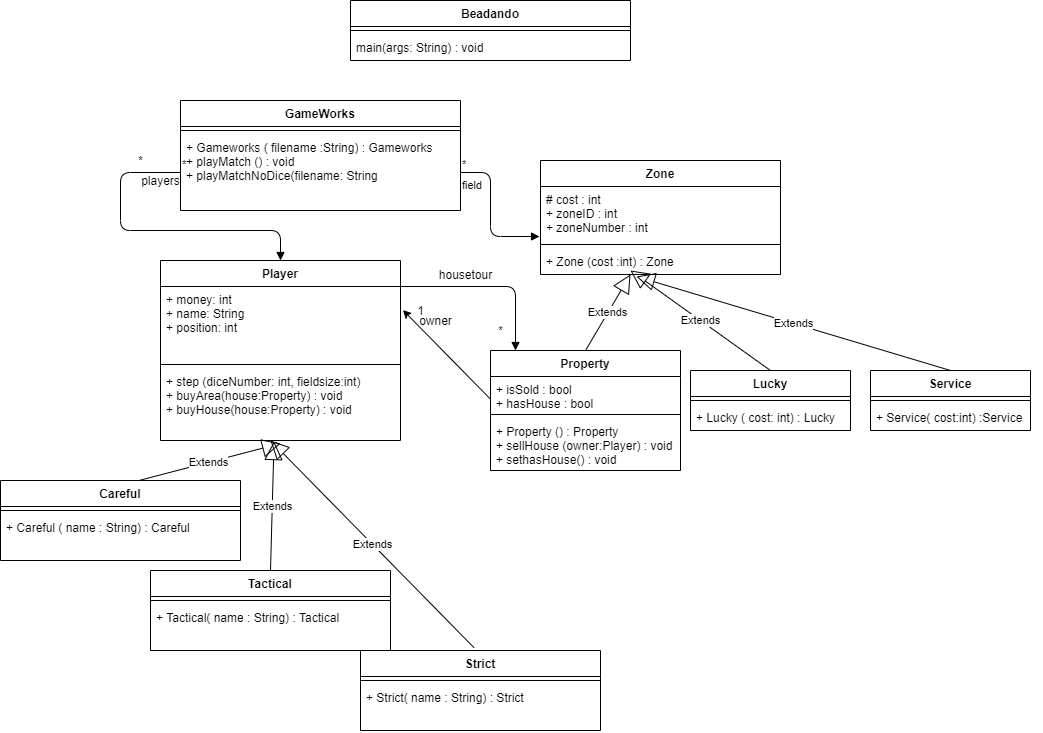
Lukács Dávid István, idu27k   
programozási technológia

1. beadandó

A feladat leírása  
  
Szimuláljuk az alábbi egyszerűsített Capitaly társasjátékot! Adott néhány eltérő stratégiájú játékos és egy körpálya, amelyen különféle mezők sorakoznak egymás után. A pályát körbekörbe újra és újra bejárják a játékosok úgy, hogy egy kockával dobva mindig annyit lépnek, amennyit a kocka mutat. A mezők három félék lehetnek: ingatlanok, szolgáltatások és szerencse mezők. Az ingatlant meg lehet vásárolni 1000 Petákért, majd újra rálépve házat is lehet rá építeni 4000 Petákért. Ha ezután más játékos erre a mezőre lép, akkor a mező tulajdonosának fizet: ha még nincs rajta ház, akkor 500 Petákot, ha van rajta ház, akkor 2000 Petákot. A szolgáltatás mezőre lépve a banknak kell befizetni a mező paramétereként megadott összeget. A szerencse mezőre lépve a mező paramétereként megadott összegű pénzt kap a játékos. Háromféle stratégiájú játékos vesz részt a játékban. Kezdetben mindenki kap egy induló tőkét (10000 Peták), majd a „mohó” játékos ha egy még gazdátlan ingatlan mezőjére lépett, vagy övé az ingatlan, de még nincs rajta ház, továbbá van elég tőkéje, akkor vásárol. Az „óvatos” játékos egy körben csak a tőkéjének a felét vásárolja el, a „taktikus” játékos minden második vásárlási lehetőséget kihagyja. Ha egy játékosnak fizetnie kell, de nincs elégendő pénze, akkor kiesik a játékból, házai elvesznek, ingatlanjai megvásárolhatókká válnak. A játék paramétereit egy szövegfájlból olvassuk be. Ez megadja a pálya hosszát, majd a pálya egyes mezőit. Minden mezőről megadjuk annak típusát, illetve ha szolgáltatás vagy szerencse mező, akkor annak pénzdíját. Ezt követően a fájl megmutatja a játékosok számát, majd sorban minden játékos nevét és stratégiáját. A tesztelhetőséghez fel kell készíteni a megoldó programot olyan szövegfájl feldolgozására is, amely előre rögzített módon tartalmazza a kockadobások eredményét.   
  
Adjuk meg, melyik játékos nyeri meg a játékot és mekkora vagyona (mennyi a tőkéje, milyen ingatlanokat birtokol) van ekkor!

Osztálydiagram

mETÓDUSOk

## Property :: Sellhouse

Az ingatlan zónák területének kiadására szolgál, beálíltja a bool értéken keresztül, hogy az a zóna már el van adva. Emellett a paraméterben megadott Player referenciát is eltároljuk, hogy később visszakövethető legyen kinek kell fizetnie annak, aki rálép a mezőre.

## Property :: Sethashouse

az ingatlan zónákon történő házépítést kezeli, egy bool-t állítunk be rajta keresztül, mivel a tulajdonos már ismert.

## player :: Step

A játékosnak a mezőkön történő lépegetést teszi lehetővé. Mivel körbe megy a pálya, ezért maradékos osztással megoldottuk azt is, hogy az utolsó után az első mezőre térjen vissza.

## player :: buyaera

Az ingatlanos mező megvásárlása, először csak a területé. Hozzáadjuk a zónát a játékos tulajdonai közé, amely egy gyűjtemény, és meghívjuk a terület SellHouse metódusát.

## player :: buyhouse

ház építése a meglévő területen, ezt csak akkor hívjuk meg ha a játékosnak biztosan megvan már a területi rész.

## gameworks::gameworks

A GameWorks konstruktora, amit egy pálya generátornak használunk jelen esetben. A fájlból beolvasás

során először a sorok számát, majd egy betűt és egy számot várunk, szóközzel elválasztva, soronként. Az első betű lesz a zóna típusa ( S – service, P – property, L – luck ) majd a Service és Luck esetében a zónák számértéke. Kezeljük azon kivételeket, amikor a fájl rosszul van formázva, vagy ha rossz a fájlnév/nem elérhető bármilyen okból.

## gameworks::playmatch

Egy játék lejátszása. Véletlenszerűen generált számokat dob a dobókockával, és addig játszatja a játékosokat, míg csak egy marad. A végén kiírja, hogy mennyi pénze van a nyertesnek, és mely ingatlanmezőket birtokolta, esetleg épített-e rajtuk házat vagy sem.

## gameworks::playmatchnodice

Egy játék lejátszása. Előre megadott számokat dob a dobókockával, és addig játszatja a játékosokat, míg csak egy marad. A végén kiírja, hogy mennyi pénze van a nyertesnek, és mely ingatlanmezőket birtokolta, esetleg épített-e rajtuk házat vagy sem. Kivételt dob akkor, ha nem elég a dobások száma a nyertes eldöntéséhez, illetve ha bármilyen okból nem elérhető a fájlunk, vagy az rosszul van formázva.

## gameworks::tostring

A játék elején kiírja az összes játékost, azok stratégiáját, és kezdőpénzét, valamint soronként a mezők típusát, és Service vagy Luck mező esetén azok számértékét.

## gameworks::playround

Ez a metódus tartalmazza a játék fő szabályait, hogy a játékos stratégiájának megfelelően cselekedjen, hogy mi történik ha Service vagy Luck mezőre lép a játékos, vagy ha valaki más területére. Itt folynak le a pénzlevonások és hozzáadások is. Minden a feladat leírásának megfelelően működik, és ez a metódus van iterálva a playmatch metódusok ciklusmagjában.

## gameworks::tostringwinner

Ez a speciális toString metódus azért készült, hogy a nyertest egy kicsit más formátumban tudjuk kiírni a konzolra a játék végén (péládul kiírjuk a neve, a pénze mellett a házait is.)

# tesztelés

## creatinggameworks

Ebben a tesztesetben a pálya megfelelő létrejöttét vizsgáljuk, hogy minden mező megfelelően beolvasásra került, illetve, hogy a játékosok is megfelelően bekerültek a gyűjteménybe.

## properend

Ebben a tesztesetben a játék megfelelő lezárulását vizsgáljuk, azaz hogy a játék végére ténylegesen csak egy játékos marad a gyűjteményben. Ehhez a Random osztályt használó playMatch metódust használjuk.

## nowinner

Ebben a tesztesetben a játék nem megfelelő lezárulását vizsgáljuk, azaz ha elfogynak az előre megírt dobókockaszámok a fájlból, akkor megfelelően exceptiont kapunk e. (Ez most a kivételkezelés miatt az 2-es szám visszatéréseként jelenik meg.)

## worksthesameeverytime(2)

Ebben a tesztesetben a játék ugyanolyan lefutását vizsgáljuk, az egyszeri lefutásnál megvizsgált végeredményt nézzük, hogy minden esetben ugyanazt kapjuk-e.

## nodice

Ebben a tesztesetben a játék kivételdobását vizsgáljuk, abban az esetben ha nem található a dobókocka számait tartalmazó fájl.

## oneplayer

Ebben a tesztesetben a játék viselkedését vizsgáljuk, egy játékos esetén, hogy megmarad-e a kezdőpénze (10000.)