Sbírka krátkých lehkých her na pár minut nebo vteřin

Zápočtový program

Dokumentace

Autor: Jan Růžička
Garant: prof. RNDr. Tomáš Bureš, Ph.D.
Studium: bakalářské, 2. ročník, letní semestr 2023/2024
Předmět: Programování v Pythonu NPRG065

Obsah

Obsah

Uživatelská část

Start

Pro spuštění sbírky krátkých lehkých her je třeba mít stažený jazyk Python 3.x. Následně je třeba mít stažený projekt, kde se nacházejí všechny potřebné soubory pro plnohodnotný zážitek hraní her. Pro spuštění je třeba najít main.py soubor, kde se nachází hlavní kostra projektu a její propojení se všemi hrami, kterou je třeba spustit za pomoci jazyka Python. Po zapnutí hlavní kostry projektu je možné vybírat z několika her.

Průběh:

Na začátku má uživatel možnost si vybrat za pomocí čísel dostupnou hru, kterou následně spustí hlavní menu vybrané hry. Zatím je možné vybírat hry: univerzální kostka, dáma, miny, univerzální kámen nůžky papír a had.

Veškeré hry se hrají za pomocí konzolové aplikace CMD – příkazový řádek.

Příkazy

Každá hra má svá vlastní pravidla zadávání příkazů. Příkazy můžete najít v uživatelské části podle názvu hry.

Dáma

Příkazy

Na začátku hraje Player 1 a je po něm požadováno, aby napsal platný příkaz ve tvaru [Zdrojová Figurka → Cílová Pozice] - [X1 Y1 X1 Y1 – kde X je sloupec figurky a Y řádek figurky]

Ukázka příkazu:

0 5 1 4

Po zadání platného příkazu se přesune zadaná figurka do cílové pozice a následně pak začíná Player 2. Toto se opakuje až do skončení hry.

Výběr skoků

Pokud by se nastala situace, kdy hráč musí skákat, systém hry udělá tento skok místo hráče a rovnou se hráči vystřídají. V případě, že skoků je více, systém vypíše seznam možných tahů, takže si hráč může vybrat skok, který chce. Po zapsání do konzole číslo 1 až N podle velikosti možných skoků.

Ukázka výběru skoků:

Výpis možných skoků:

1) X1 Y1 X2 Y2

2) X1 Y1 X2 Y2

Do konzole:

1 → vybere se tah 1)

Univerzální kostka

Příkazy

Na začátku má možnost si hráč nastavit nebo vybrat defaultní nastavení klasické kostky 1-6. Při nastavení vlastní kostky musí hráč předem vybrat kolik stran taková kostka má mít. Po výběru počtu stran přichází fáze postupného přidávání hodnot pro každou stranu.

Formát přidávání strany: [jméno ikonka pravděpodobnost], kde

jméno – jméno strany

ikonka – jaký je vzhled strany

pravděpodobnost – šance, že padne daná strana

Po fázi nastavení kostky může hráč libovolně házet stejnou kostku, než se rozhodne, že chce hru skončit.

Miny

Příkazy

Na začátku má možnost si hráč nastavit vlastní desku nebo vybrat defaultní nastavení desky klasické hry miny 10x10 s 20 minami. Při nastavení vlastní desky musí hráč předem vybrat délku a výšku desky a počet bomb, kolik má být na desce.

Formát vlastní desky: [délka výška počet min]

Na začátku hráč musí zadat délku a výšku desky za pomocí čísla z množiny přirozených čísel 0-nekonečno a potom počet bomb za pomocí čísla, které musí být kladné a menší než násobek výšky a délky.

Po fázi nastavení desky začíná hráč postupně vyhledávat miny. Pro výběr políčka zadává hráč příkaz ve tvaru: [X Y R/L], kde

X a Y - je pozice políčka na desce

R/L - je druh kliku – pravý nebo levý.

Pravý klik znamená nastavení vlajky, na které si hráč řekl, že tam může být mina a nemůže žádným způsobem být otevřena. Jediným způsobem je na dané políčko znovu kliknout pravým tlačítkem a odstranit tak vlajku.

Levý klik znamená, že hráč chce otevřít dané políčko. Po otevření políčka se může ukázat mina nebo číslo. Pokud se tam ukázala mina, hráč končí a je po něm požadováno rozhodnutí, zda chce hrát znovu nebo ne. Pokud se ukáže číslo 0, tak se otevřenou všechna sousední políčka. Toto se opakuje pro všechny nadcházející políčka s číslem 0 nebo končí s libovolně jiným číslem. Jakmile hráč najde všechny čísla, vyhrál a je po něm požadováno rozhodnutí, zda chce hrát znovu.

Had

Příkazy

Na začátku je dotaz, zda je hráč přípraven hrát za použití tlačítka ENTER. Poté se objeví deska, na které se neustále pohybuje had, který je ovládán hráčem. Deska je předem nastavená a není ji možné předem nakonfigurovat podle hráče. Nastavení je možné jen přímo v souboru, kde se nachází hlavní kód hry.

Hráč do konzole žádné příkazy nezadává, ale mačká na klávesnici 4 tlačítka – WASD, kde

W – směr nahoru

A – směr doleva

S – směr dolů

D – směr doprava

Hráč na desce hledá potravu pro hada. Po snědení potravy se had zvětší. Hra běží tak dlouho, dokud had není tak veliký, že naboural do stěny nebo sám do sebe.

Univerzální kámen nůžky papír

Příkazy

Na začátku se připraví 2 hráči, kteří budou hrát univerzální kámen nůžky papír. Pro přidání hráče je třeba zadat specifikaci hráče podle formátu: [mód skóre jméno], kde

mód – zda je AI nebo hráč

skóre – počáteční hodnota skóre hráče

jméno – jméno hráče, který hraje hru univerzální kámen nůžky papír

Po nastavení všech hráčů se zapne hra a hráči si po jednom vybírají zbraň, která je dostupná. Pro výběr zbraně hráč musí napsat do konzole přesné jméno dostupné zbraně. Jakmile si hráči vyberou zbraň, nastane fáze kontroly výběru zbraní. Vyhraje ten, kdo má lepší zbraň.

Vypnutí hry

Uživatel může vypnout hru pomocí X, které se nachází vpravo nahoře. Případně lze program vypnout i přes Ctrl-C.

Programátorská část

Anotace

Tato dokumentace popisuje menu a sbírku her. Zdrojové kódy jsou psány v jazyce Python. Hlavním cílem této dokumentace je popis tohoto projektu.

Zadání

Cílem zadání je vytvořit sbírku krátkých lehkých her na pár minut nebo vteřin za pomocí jazyka Python, včetně prvků vlastní grafiky za pomoci konzole. Každá hra může nebo nemusí být podle klasických pravidel.

Menu

Pravidla

- Hráč má možnost si v hlavním menu vybírat dostupné hry.
- Po výběru se ukáže hráči hra, kterou může hrát.

Funkčnost

Menu funguje za pomocí 3 souborů:

- GameLoaderClass.py
- main.py
- AbstractGameClass.py

V hlavním programu main.py běží za pomoci enumerátoru 3 stavy:

- Load - počáteční stav, kdy nejdříve vyhledáváme všechny soubory- Vyhledávání souboru má na starost soubor GameLoaderClass.py. Ten funguje tak, že si načte všechny soubory, které se nachází ve složce ./Games a pak postupně vybírá ty, které v sobě mají nějakou třídu, která dědí abstraktní třídu AbstractGameClass, který se nachází v souboru AbstractGameClass.py a zároveň musí mít v sobě implementované metody start() a getNameGame(). Pokud nemá, vyhodí výjimku. Pokud žádné hry nenašlo, ukončí se program a nic se neudělá. Jakmile se najdou hry, přepne se do stavu Main Menu.
- Main Menu - stav, který běží tak dlouho, dokud se nevybere daná hra za pomocí čísel, která jsou vypsána do konzole. Po výběru hry se přepne do stavu Selected Game a zavolá metodu start(), která zapne hru.
- Selected Game - stav, při kterém právě běží nějaká vybraná hra. Jsme v tomto stavu tak dlouho, dokud hra se nedohraje nebo se neukončí, pak se vrátíme zpět do stavu Main Menu a můžeme znovu vybírat hru.

Dáma

Pravidla

- Velikost šachovnice je 8x8
- Na dvou protilehlých stranách má každý hráč 12 kamenů.
- Figurky se pohybují po diagonále pouze vpřed o 1 políčko, ale ne vzad a nemohou přeskakovat vlastní figurky
- Pokud se daná figurka dostane na druhý konec šachovnice, přemění se na dámu
- Dáma se pohybuje dopředu i dozadu vždy o 1 políčko
- Jestliže se figurka nachází na diagonále v sousedství soupeřovy figurky, za kterou je volné pole, je povinen soupeřovu figurku přeskočit a sebrat ji
- Skákání je povinné pro všechny figurky. Jestliže je více možných skoků, může si hráč vybrat, kterou figurkou bude skákat
- Obyčejná figurka může sekvenčně skákat směrem dopředu, ale figurku, která byla přeskočená, nemůže znovu přeskočit zpátky
- Dáma může sekvenčně skákat směrem dopředu i dozadu, ale figurku, která byla přeskočená, nemůže znovu přeskočit zpátky
- Hra končí, pokud na jedné straně nezbydou žádné figurky

Funkčnost

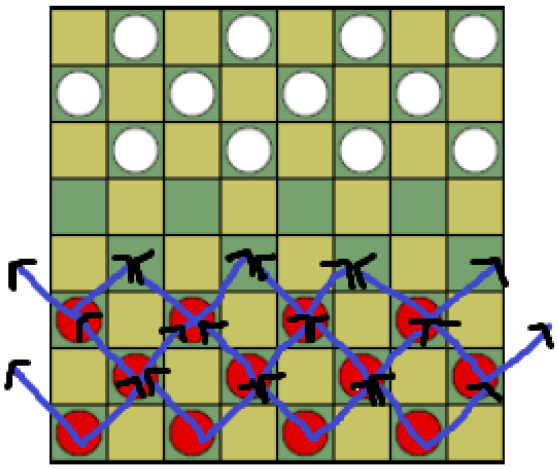
Hra funguje na principu donekonečna běžících stejných příkazů, dokud někdo z hráčů nevyhraje.

Nejdříve se vygenerují bílé a černé figurky pro dva hráče na šachovnici ve tvaru 8x8.

První příkaz, který se provede je vykreslení šachovnice a poté se vždy provede kontrola, zda již někdo hru nevyhrál. To se zjistí podle toho, kolik je černých a bílých figurek. Pro pokračování hry je potřeba alespoň jedna figurka od každé barvy.

Poté proběhne druhý příkaz a to je nejdůležitější algoritmus na prohledání všech možných skoků. Pokud je možností více, hráč vybere jednu možnost a tu musí povinně provést. Pokud je pouze jedna možnost, musí ji hráč povinně provést, toto udělá program za hráče automaticky.

Funguje to tak, že pro každou figurku v dané barvě, se kterou bude hráč táhnout, proběhne kontrola, zda v diagonále ve směru hry neexistuje protivníkova figurka. Pokud ano proběhne kontrola, zda za figurkou je volné políčko. Pokud ano, uloží se daný tah do seznamu možností. Toto se provede pro všechny figurky hráče, který je na tahu. Jakmile se proběhly všechny kontroly, tak se algoritmus podívá kolik existuje možností skoků. Pokud je možností více než 1, vypíše se seznam možných tahů a hráč vybere jeden z nich, potom se tah provede. Pokud je pouze jedna možnost, musí ji hráč povinně provést, toto udělá program za hráče automaticky. Na tahu potom protihráče, u něj probíhá stejný postup. Pokud nebyla nalezena žádná možnost, tak se ukončí tento algoritmus a proběhne poslední příkaz.



Ukázka algoritmu pro vyhledávání automatických skoků – obrázek převzat z Wikipedie

Následuje další algoritmus. U této části si může hráč libovolně vybrat pohyb figurky za předpokladu dodržení pravidel, což kontroluje algoritmus. Pokud nejsou pravidla dodržena, musí hráč zadat nový příkaz. Pokud dodržena jsou, tak se provede tah a pokračuje ve hře protihráč.

Tyto 3 příkazy běží donekonečna, dokud z jeden hráčů nevyhraje.

Vybral jsem si tento způsob zpracování pohybů ve hře kvůli tomu, že mi to přišlo jednodušší a snadněji implementovatelné než když bych spojil volné pohyby a povinné skoky v jednom algoritmu. Uvědomil jsem si, že skoky nastanou právě tehdy, pokud protihráč udělá volný pohyb a pak se zjišťuje, jestli se má udělat povinný skok. Proto jsem rozdělil algoritmus na 2 části → automatické skoky a volný pohyb. Nikdy nestane, že bych musel dělat povinný skok a zároveň volný pohyb. Díky tomu jsem si ulehčil práci s vymýšlením algoritmu. V průběhu vytváření hry mě napadlo mnoho algoritmů, ale zahodil jsem je, jelikož by nefungovali, až na tento, který mi přišel jednoduchý a snadno implementovatelný.

Univerzální kostka

Pravidla

- Hráč může házet univerzální kostkou, kterou si sám nastavil. Lze jí házet několikrát
- Hráč má možnost nastavení počtu stran kostky
- Hráč má možnost nastavení jméno každé strany
- Hráč má možnost nastavení ikonky každé strany
- Hráč má možnost nastavení pravděpodobnosti každé strany
- Každá strana má vlastní jméno
- Každá strana má vlastní ikonku
- Každá strana má vlastní pravděpodobnost výskytu

Funkčnost

Tato hra vytváří univerzální kostku, která je nastavitelná. Tedy může mít libovolný počet stran s různými pravděpodobnostmi výskytu každé strany. Po spuštění této hry uživatel bude dotázán, zda chce použít výchozí nastavení kostky. Ta má 6 stran se stejnou pravděpodobností. Tyto strany se postupně přidávají za pomoci metody addNewSide. Dále má uživatel možnost si sám vybrat počet stran a jednotlivě přidávat postupně strany za pomocí 3 parametrů, které se následně vloží do metody addNewSide. Jakmile hráč vyplní všechny strany, které si zadal, spustí se metoda game, která běží tak dlouho, dokud chce uživatel házet kostkou. Jakmile bude spokojen, vypne za pomoci příkazu hru, kterou hráč uvidí v konzoli.

Miny

Pravidla

- Hráč musí na desce najít všechna políčka, na kterých se nenachází mina
- Hráč má možnost na daném políčku položit vlaječku, kterou nikdo nesmí otevřít. Hráč může také vlaječku odstranit
- Hráč postupně může vybírat neotevřená políčka levým klikem a otevřít je
- Po otevření políčka se musí vyskytnout číslo nebo mina. Pokud se na políčku ukáže číslo 0, znamená to otevři sousední políčka. Toto se opakuje tak dlouho, dokud se v daném okamžiku neutvoří stěny z čísel 1-8 nebo uzavřená políčka pomocí vlaječky.
- Hráč vyhrává, jakmile najde všechna políčka, která jsou označena nějakým číslem.
- Hráč pohrává, jakmile otevře nějaké políčko, na kterém se nachází mina.
- Hráč má možnost nastavení délky desky
- Hráč má možnost nastavení výšky desky
- Hráč má možnost nastavení počtu min na desce

Funkčnost

Tato hra vytvoří desku, která je náhodně vyplněná minami. Na začátku může hráč specifikovat desku za pomoci 3 parametrů: délka, výška a počet min, které chce do hry. Po zadání parametrů nastává fáze vytváření desky. Nejdříve se vytvoří jednorozměrné pole, které bude dlouhé podle velikosti desky - délka*výška a do tohoto pole budeme postupně přidávat instance políček typu Square, která má v sobě důležité informace, např. zda má minu, vlajku, pozici a tak dále. Do pozice vkládáme hodnoty x a y, tak aby se chovala jako dvojrozměrné pole v jednorozměrném poli. Přístup do dvojrozměrného pole můžeme dostat podle vzorce: celková výška * vybraná výška + délka pozice zleva (offset).

Jakmile je vytvořená deska vyplněná instancemi třídy Square, postupně se vybírají náhodná políčka z desky a vloží se mina. V každém políčku, které sousedí s minou se zvýší hodnota o +1. Tento algoritmus jsem si zvolil, abych stále nemusel sledovat, zda se někde v okolí nachází mina. Díky tomuto způsobu mohu otevírat políčka v konstantním čase, aniž bych musel něco počítat. Toto se opakuje tak dlouho, dokud se na každé unikátní políčko nevloží mina.

Hráč vybírá políčka v konzoli ve formátu: [X Y R/L]. Pokud vybere políčko pomocí levého kliku, otevře se políčko a na základě čísla se pomocí algoritmu BFS otevírají další sousední políčka, která mají 0. Tento algoritmus bude otevírat políčka tak dlouho, dokud se nevytvoří ohraničení čísel různé od 0. Tyto příkazy se opakují tak dlouho, dokud se neotevřou všechna možná bezpečně otevíratelná políčka. Jakmile jsou otevřena všechna políčka, hráč vyhrál a hra končí.

Pokud hráč otevře políčko, na kterém se nachází mina, prohrál hru.

Univerzální kámen nůžky papír

Pravidla

- Hráč/Hráči/AI si navzájem hrají konfigurovatelnou hru kámen nůžky papír.
- Hráč/Hráči/AI si předem vyberou zbraň, se kterou budou hrát a utkají se s protivníkem
- Vyhrává ten, kdo má lepší zbraň

Funkčnost

Hra nemá možnost předem v průběhu přidávat zbraně, se kterými lze hrát. Ale je možné předem v souboru nastavit zbraně, které se nacházejí v enumerátoru Weapons a je možné vložit nové zbraně a přidat pole hodnot, proti kterým vyhrává zbraň vůči jiným. Upozornění předem - zbraně musí být navzájem disjunktní, tedy to znamená, že 2 různé zbraně nemohou navzájem vyhrát, jelikož toto se nekontroluje a zároveň se očekává, že správnost zajišťuje programátor, který tyto zbraně přidá. Jejich nesprávnost neukončí hru, ale nebude fungovat jak má.

Hra má 4 možné módy: hráč vs hráč, hráč vs AI, AI vs hráč, AI vs AI.

Vybírání zbraní záleží na pořadí, tedy začíná ten, kdo je v první pozici v daném módu. Tedy např. mód AI vs hráč. V tomto případě začíná AI a poté hráč.

AI si náhodně vybírá ze seznamu zbraní, které se nacházejí v enumerátoru Weapons za pomocí knihovny random.

Hráč si vybírá na konzoli podle názvu zbraně, při zadání špatné hodnoty se hráči vypíší všechny možné zbraně, které se nachází v enumerátoru Weapons.

Jakmile si všichni vyberou zbraň, proběhne kontrola ve stejném pořadí jako při výběru zbraně. Kontrola probíhá tak, že se vezme zbraň prvního hráče a ověří se, zda vyhrává nad zbraní soupeře. Poté i pro druhého hráče proběhne kontrola. Pak se ověří výsledky u obou hráčů a dojde k vyhlášení vítěze. Pokud ani jeden z nich nevyhrál, nastala remíza. Pokud je vítěz jasný, získává skóre +1.

Had

Pravidla

- Hráč může ovládat hada na obrazovce pomocí klávesnice.
- Hráč s hadem hledá potravu, která se generuje postupně na obrazovce
- Pokud had sní nějakou potravu, prodlouží se mu ocas.
- Hra pokračuje tak dlouho, dokud had nenaboural do stěny nebo sám do sebe.

Funkčnost

Hra má předem nastavené parametry. Ovládání hada funguje za pomoci globálního sledování nastaveného směru. Enumerátor má 4 stavy – UP, DOWN, LEFT a RIGHT. Hra se aktualizuje v předem určený čas, který je v souboru nastavený. Při každé aktualizaci obrazovky se uklidí předchozí obrazovka a nahradí ji nová obrazovka s aktualizovaným stavem. Při každé aktualizaci se had posune dopředu v určitém směru za pomocí globální stavu. Také se kontroluje, zda had nenaboural nebo nesnědl nějakou potravu. Kontrola probíhá tak, že se podíváme do seznamu těla hada, vezmeme první prvek a zkontrolujeme, zda hlava hada nemá stejnou pozici jako některá část jeho těla. Zvětšení hada funguje tak, že si vezmeme poslední prvek ze seznamu a zkopírujeme jeho pozici a přidáme ji jako nový prvek.

Sada testovacích příkladů

Testovací příklady jsou dostupné ve složce Tests, zatím pouze pro hru Dáma. Testy pro další hry nejsou z důvodu, že výsledky jsou často nedeterministické a to se týká hlavně her: miny, univerzální kámen nůžky papír, univerzální kostka a had. Je možné udělat unit testy, ale ty nedávají pro některé funkce smysl, protože mají nedeterministické hodnoty a mohou se měnit na základě průběhu hry. Dáma na rozdíl od těchto her je velice deterministická a proto je jedinou hrou, která má v sobě nějaké testy.

Pro spuštění testovacích příkladů napište následující příkaz:

python main.py < cesta/kod.txt

Korektnost

Korektnost her jsem testoval manuálně:

– hra byla vždy ukončena řádně

– proběhly všechny možné situace

Po otestování všech možných variant jsem dospěl k závěru, že hry fungují bez problému. Je možné, že jsem některé možnosti nevzal v potaz a nemusí vždy správně fungovat, ale pro správné fungování projektu by to mělo být zanedbatelné. Takové případy by musely být hodně extrémní, skoro nemožné v běžné hře. Případně bude nutné zapnout projekt znovu.

Kódy

Dáma

1 – Ukázka jednoho automatického skoku + výběr z možných automatických skoků

2 – Ukázka výhry jednoho z hráčů

3 – Ukázka, že hráč nemůže skočit na svojí figurku

4 – Ukázka, že figurka nemůže jít zpátky

5 – Ukázka, že dáma může jít pohybem zpátky

6 – Ukázka, že dáma může skákat dozadu + sekvenční skok dámy

7 – Ukázka sekvenčního skoku figurky

8 – Ukázka, že hráč nemůže udělat pohyb na protihráčovu figurku

9 – Ukázka, že po automatickém skoku se prohodí pořadí hráčů

10 – Ukázka, že po neplatném příkazu může hráč znovu zadat příkaz + nemůže hrát protihráčovou figurkou

Závěr

Touto dokumentací jsem se pokusil popsat fungování hry v jazyce Python. Shrnul jsem všechny důležité části.

Zadávání příkazů může být pro někoho matoucí a neintuitivní, ale pro zábavu je taková maličkost zanedbatelná. Totéž platí i pro grafické zobrazení za pomoci konzole.

Pro budoucí vývoj je možné přidávat další hry do složky games, ve které je nutné dodržet strukturu dané hry. Je také možné převést hry na grafické rozhraní a tak dále.

Zdroje

Pravidla

https://cs.wikipedia.org/wiki/D%C3%A1ma_(deskov%C3%A1_hra) – dáma
https://cs.wikipedia.org/wiki/K%C3%A1men,_n%C5%AF%C5%BEky,_pap%C3%ADr – kámen nůžky papír
https://cs.wikipedia.org/wiki/Hled%C3%A1n%C3%AD_min – hledání min
https://cs.wikipedia.org/wiki/Hrac%C3%AD_kostka – házení kostkou
https://www.hranostaj.cz/hra681 – had

Třetí knihovny

https://docs.python.org/3/library/enum.html – Enum
https://docs.python.org/3/library/os.html – Os
https://docs.python.org/3/library/importlib.html – Importlib.util
https://docs.python.org/3/library/inspect.html – Inspect
https://docs.python.org/3/library/random.html – Random
https://docs.python.org/3/library/msvcrt.html – Msvcrt
https://docs.python.org/3/library/time.html – Time

Hra dáma a její část dokumentace byla převzata z minulého ročníku, dělal jsem ji pro předmět pro Programování 1 v roce 2022/23. Povolení k využití mé práce do tohoto zápočtového programu jsem dostal.
Ověření z emailů:

Povolení od RNDr. Tomáš Holan, Ph.D.:
„Dobrý večer,

dobře, proti tomu nic nemám.

Přeji Vám hodně zdaru!

Tomáš Holan “
Povolení od doc. RNDr. Jan Kofroň, Ph.D.

„Dobrý den,

to je určitě vhodné téma, tedy pokud výsledek bude mít rozumný rozsah kolem těch 1000 řádků. Využití
programu v jiném předmětu mi nevadí, ale ověřte to zejména s vyučujícím toho druhého předmětu.

S pozdravem,
Jan Kofroň“