

Střední Průmyslová Škola Elektrotechnická Ječná

Informační technologie

Ječná 30, Praha 2, 130 00

Memory game

(popř. Pexeso)

Jan Vavroušek

Informační Technologie

2024

Obsah

1	Cíl práce	3
2	Hardware (Volitelné)	3
3	Software	3
4	Program	3
5	Závěr	3
6	Zdroje	3

1 Cíl práce

Závěrečná ročníková práce, na které jsem mohl využít své zkušenosti, dovednosti a znalosti, kterých jsem dosáhl za svého působení na střední průmyslové škole elektrotechnické v Ječné. Nejen využití znalosti programovacího jazyka, ale také využití anglického jazyka, ve kterém je celá hra prováděna. Znalosti jsem získal formou samostudia, přípravou na školní projekty, komunikací mezi spolužáky a zkušeností, které jsem získal na závěru roku z praktické části výuky v různých firmách. Práce má za cíl naučit pracovat s časem, donutit mě vyhledávat si nové věci a učit se nové věci. Konečná verze ročníkové práce má obsahovat správně naprogramovanou hru piškvorek, která nelze shodit, je plně funkční a obsahuje spoustu vymožeností, které jsem dokázal vymyslet. Finální práce probíhá několika testy a celkové fungování programu by mělo být podchycené výjimkami a neshoditelné. Celý program byl od začátku zamýšlen jako program, který je vyvolán z konzole a pouze vize do budoucna spočívala v pokus aby existovala 2. verze, která má mít základní prvky grafického rozhraní.

2 Software

Java Development Kit 22.0.2 stažený z oficiálních stránek Oracle a následně instalován, Java version 8 stažená z oficiálních stránek Java. Knihovny Jackson, která podporuje zapisování souborů do formátu JSON, tato knihovna je stažená ze stránky Maven repository, jenž podporuje klasický zápis do formátu JSON a byla vložena do programu. Mezi tyto knihovny se řadí soubory jackson-annotations-2.17.1, jackson-core-2.17.0, jackson-databind-2.17.0. Veškeré tyto knihovny jsou již obsažené ve finální verzi hry. JUnit test, knihovna implementovaná skrz IntelliJ Idea. IntelliJ Idea, vývojové prostředí využívané pro programování celého projektu.

3 Popis hry

3.1 Mechaniky

Hra nevyžaduje složitých mechanik, důležité v této hře je číst si co po vás aktuální metoda a chce umět na to řádně odpovědět. Spuštění programu uživatel vpustí do prostředí konzole, kde uživatel může využít příkazy start, help, history, exit. Po startnutí hry, stačí pouze číst anglicky napsaný text a řádně vyplňovat vstupy, které po Vás aplikace chce. Uživatel potřebuje funkční počítač, klávesnici a čistou mysl na přemýšlení během hraní.

3.2 Příběh

Uživatel po startu hry se ocitne téměř v normální situaci, ve které se může ocitnout úplně kdokoli. Ovšem v normálním životě uživatel nemá možnosti určit kolik párů chce rozložit.

4 Manuál

Program se základně spustí v konzoli. Nejprve vyskočí text s uvítáním a prosbou a zadání prvního příkazu. Příkazy, které dokáže tato konzole zpracovat jsou následující: help, history, start, exit. Jednotlivé příkazy mají jednotlivé funkce, help vypíše všechny možné příkazy, které program zahrnuje. Za pomoci příkazu History lze vypsát veškeré příkazy a veškeré vstupy, které uživatel během toho, co měl spuštěný program napsal. Start má funkci spustit

program, jenž by měl bez problémů spustit hlavní kód, který tento program obsahuje. Exit umožňuje

uživateli opustit konzoli a tím i celkově program. Spuštěním samotného základního kódu, skrz příkaz Start, který program nabízí Vás automaticky doprovodí do prostředí, ve kterém si uživatel může nastavit veškeré vymoženosti, které aplikace nabízí. Zejména se jedná o určení počtu hráčů, který následně bude hru hrát. Následně se hráči jednotlivě pojmenují a jejich jméno společně s automatickým ID a skórem se запиše do JSON souboru. Dále si hráč zvolí přesný počet dvojic, které chce aby byly vyobrazeny na hrací ploše, které následně poté také bude chtít uhodnout. Poslední věc, kterou je program obohacena je funkce přidávání symbolů. Tato funkce má zejména využití na implementování emoji na hrací plochu, jelikož Java je obohacena sadou Unicode. Základní znaky, které jsou generovány na hrací plochu, jsou písmena abecedy. Po zadání všech těchto možností, které si uživatel může volit, se hra vygeneruje a záleží pouze na uživateli, jakou podobu hrací plocha má. Program na základě všech uvedených informací dokáže sám pracovat, automaticky vygeneruje hrací plochu s náhodně rozloženými políčkama pexesa, napíše jaký uživatel začíná a položí první otázku. Otázky jsou celkem dvě a první otázka spočívá v tom, že se uživatele zeptá v jakém řádku chce políčko otočit a druhá otázka je v jakém sloupci chce políčko otočit. Následně uživatel udělá stejný proces u druhého políčka a pokud se symboly shodují, vymažou se a uživatele to informuje a pokud nejsou, tak se políčka otočí zpět. Také pokud uživatel uhodne správné dvojice, má výhodu okamžitého druhého pokusu na otočení dalších dvojic a tím pádem šanci získat více dvojic. Tento proces se opakuje do otočení všech políček. Výhercem se stává hráč, jenž má největší skóre. Pokud takový hráč není a výsledné skóre se shoduje u dvou hráčů s nejvyšším skórem, nevyhraje nikdo.

5 Závěr

Práce mně osobně se dělala mnohem lépe než v 1. ročníku. Celá hra mi dávala mnohem větší smysl, byl jsem schopen využít mnoho znalostí, které jsem za tyto 2 roky získal a ukázat si osobní progres za uplynulé 2 roky bylo velmi uspokojivé. Během této práce mě programování velmi bavilo a ukázalo jaké možnosti až může mít. Během práce jsem se setkal s komplikacema, zejména v oblasti vymýšlení algoritmů a podobně, ovšem si myslím, že se všema problémama jsem si poradil ať už sám, tak už za použití internetu, který byl často dobrým pomocníkem a na spoustu algoritmů a myšlenek mi dal nápad.

6 Zdroje

- [1] Jackson-core 2.17.0 dostupný z:
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core>
- [2] Jackson-databind 2.17.0 dostupný z:
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind>
- [3] Jackson-annotations 2.17.1 dostupné z:
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations>