Střední průmyslová škola elektrotechnická Ječná
Informační technologie
Ječná 517, 120 00 Nové Město

Kingdom Come: Deliverance 2 (KCD2)

Tomáš Kutil
Informační a komunikační technologie
2025

Obsah

1 Cíl práce

Cílem projektu je naprogramovat textovou dobrodružnou hru inspirovanou Kingdom Come: Deliverance 2. Hra poběží v konzoli a bude ovládána pomocí textových příkazů.

Hráč by měl být schopen:

- pohybovat se po třech mapách (dvě mapy světa a jedna mapa města),
- komunikovat s nehratelnými postavami (NPC),
- okrádat NPC postavy,
- obchodovat (nakupovat a prodávat předměty),
- hrát minihru s kostkami,
- sbírat bylinky a různé předměty,
- bojovat s nepřátelskými postavami (enemies),
- využívat funkci rychlého cestování mezi lokacemi.

Hra by měla být zábavná, přehledná a otevřená k budoucímu rozšíření (např. o další mechaniky nebo mapy). **Cílem projektu je procvičení práce s objektově orientovaným programováním v Java a vytvoření komplexnější interaktivní hry.**

2 Popis hry

2.1 Příběh/Algoritmus

Příběh hráče zasazuje do středověkého prostředí, kde muže interagovat s NPC, používat předměty a prozkoumávat síť propojených lokací. Prostředí je navrženo tak, aby připomínalo historické Čechy. Hráč interaguje se světem prostřednictvím jednoduchých textových příkazů a postupuje ve hře cestováním a interakcí s okolním světem.

2.2 Postavy

- **Hráč (Player):**
  Hlavní postava ovládaná uživatelem. Je schopna pohybu mezi lokacemi, sběru a používání předmětů, boje a interakce s ostatními postavami a prostředím.

- **Vesničan (Citizen):**
  Nehratelná postava (NPC), kterou může hráč okrást nebo s ní navázat rozhovor. Dialog s vesničanem může přinést nové informace a zároveň zvyšuje úroveň hráčovy schopnosti komunikace.

- **Nepřítel (Enemy):**
  Nepřátelsky naladěná NPC postava, se kterou může hráč vstoupit do boje. Po vítězství může hráč získat odměnu ve formě předmětu nebo jiného zisku.

- **Zvíře (Animal):**
  Lovitelná NPC postava, ze které může hráč po úspěšném ulovení získat trofej nebo jiný užitečný předmět.

2.3 Mechaniky

Příkaz	Popis funkce
`move`	Pohyb hráče po mapě mezi lokacemi.
`hunt`	Lov zvířat – hráč získá trofej.
`harvest`	Sběr bylin a rostlin.
`steal`	Kradení předmětů od NPC.
`use`	Použití vybraného předmětu z batohu.
`profile`	Výpis statistik hráče a obsahu batohu.
`sell`	Prodej předmětů obchodníkovi.
`buy`	Nákup předmětů od obchodníka.
`dice`	Minihra – hra kostky.
`show`	Zobrazení sortimentu obchodníka.
`exit`	Odchod z aktuálního města.
`help`	Výpis všech dostupných příkazů.
`print`	Výpis obsahu batohu.
`transfer`	Přesun mezi mapami.
`trade`	Zahájení obchodování – hráč volí mezi příkazy `sell`, `buy` a `show`.
`backpack`	Volba mezi příkazy `print` a `use` pro práci s batohem.
`end`	Ukončení hry.
`talk`	Rozhovor s NPC postavami. A zvýšení XP komunikace.
`enter`	Vstup do města.
`travel`	Rychlé cestování mezi objevenými městy
`fight`	Zahájení souboje s nepřátelským NPC a zvýšení XP síly.

3 Systém requirements

Program byl vyvíjen v programovacím jazyce **Java**, přičemž pro vývoj a testování byl použit **OpenJDK 21** s nastavenou úrovní jazyka (language level) na **20**. Pro spuštění aplikace je tedy potřeba mít na počítači nainstalované **OpenJDK 21** nebo vyšší verzi JDK, která podporuje jazykové funkce verze 20.
Kromě samotného JDK nejsou pro chod programu potřeba žádné externí knihovny či frameworky – veškerá funkcionalita je implementována pomocí standardních knihoven Javy.
Program lze spustit jak z příkazového řádku, tak i v libovolném vývojovém prostředí podporujícím Javu (např. **IntelliJ IDEA, Eclipse nebo NetBeans**), kde je vhodné nastavit projektové SDK na OpenJDK 21 a language level na 20, aby byly podporovány všechny využité jazykové konstrukce.

4 Základní struktura

Program je navržen objektově a rozdělen do několika základních tříd, které spolu vzájemně komunikují. Třída **Main** slouží jako výchozí bod spuštění aplikace, zatímco třída **Console** inicializuje potřebné příkazy a načítá herní svět. Třída **Player** uchovává nastavení a statistiky hráče, třída **Backpack** umožňuje interakci s hráčovým inventářem. Dále jsou zde třídy jako **Steal**, **Hunt** a **Fight**, jejichž důležitou součástí je načítání NPC ze souborů. Třída **Profile** slouží k podrobnému a přehlednému výpisu obsahu batohu a statistik hráče, zatímco třída **Dice** realizuje minihru kostky. Za zmínku stojí také třída **Transfer**, která slouží k přesunu hráče mezi mapami, a třídy **Enter** a **Exit**, které mají obdobnou funkci přesouvání hráče mezi herními lokacemi (Mapou světa a mapou města).

5 Testovací data

Program je možné testovat manuálně prostřednictvím různých herních scénářů, které pokrývají klíčové herní situace a mechaniky. Mezi základní testy patří například ověření, že příkaz move správně přemisťuje hráče mezi lokacemi a zároveň zabraňuje vstupu do neexistujících nebo nepřístupných míst. Dále je důležité zkontrolovat, zda příkaz talk spouští dialog s nehratelnou postavou a zda se po rozhovoru správně zvýší hráčova zkušenost v komunikaci. Testovat je potřeba také funkčnost kradení předmětů pomocí příkazu steal, aby program správně reagoval na úspěšné i neúspěšné pokusy o krádež. Souboj s nepřátelským NPC, realizovaný příkazem fight, by měl být otestován z hlediska průběhu souboje, zvýšení zkušeností a získání odměny po vítězství. Inventář hráče a jeho správu pomocí příkazů jako use, print a backpack je nutné otestovat, aby bylo zajištěno správné fungování manipulace s předměty. Minihru kostky, spouštěnou příkazem dice, je vhodné vyzkoušet, aby se ověřila její základní herní logika. Kromě toho je nezbytné testovat i hraniční a chybové situace, například zadání neplatného příkazu nebo nesprávného vstupu, kdy program musí uživatele vhodně upozornit a vyžádat nové zadání. Testovat je také potřeba pokusy o přesun do neexistující lokace či použití příkazů na nevhodném místě. Nakonec je vhodné ověřit, že program správně reaguje při pokusu použít nebo prodat předmět, který hráč nemá v batohu. Tyto testy jsou důležité pro zajištění stability a správné funkčnosti hry a pomáhají odhalit případné chyby v herní logice nebo uživatelském rozhraní.

6 Uživatelská příručka

Program se ovládá pomocí textových příkazů zadávaných v konzoli. Hráč komunikuje se hrou psaním jednoduchých slovních příkazů, které umožňují pohybovat se po mapě, interagovat s postavami, sbírat předměty nebo bojovat s nepřáteli. Mezi základní příkazy patří například **move** pro přesun mezi lokacemi, **talk** pro rozhovor s NPC, **steal** pro pokus o krádež, **fight** pro zahájení boje, **harvest** pro sběr bylin nebo **dice** pro spuštění minihry kostky. Příkazem **profile** si hráč může zobrazit své statistiky a obsah batohu, zatímco příkazy jako **buy**, **sell** a **trade** umožňují obchodování s NPC. Pro práci s inventářem slouží příkazy **backpack**, **use** a **print**. Hráč může také využít funkci rychlého cestování pomocí příkazu **travel** nebo přecházet mezi mapami pomocí příkazů **enter**, **exit** a **transfer**. Kdykoliv lze vyžádat pomocnou nápovědu příkazem **help**, která vypíše všechny dostupné příkazy. Hra je navržena tak, aby byla intuitivní a umožnila hráči plně prozkoumat herní svět prostřednictvím textových vstupů.

7 Závěr

Během vývoje hry jsem narazil na několik technických problémů, například s implementací metod toString a použitím printf pro formátování printů, které jsem ale díky dostupným zdrojům na internetu úspěšně vyřešil. Výsledek stál za to, protože nyní je výstup programu přehledný a odpovídá mým představám. Dalším problémem byl pokus o použití switch s hodnotou typu boolean, což v Javě není možné, a proto jsem tento problém vyřešil pomocí podmíněných příkazů if-else. Celkově jsem během tvorby hry získal mnoho nových zkušeností a určitě jsem se hodně přiučil, což hodnotím velmi pozitivně.

8 Zdroje

Při vývoji programu jsem nepoužil žádné externí knihovny, veškerý kód je postavený na standardních knihovnách Javy. Pro zlepšení formátování výstupu a využití metody printf jsem hledal informace na internetu, zejména mi pomohl článek:
Java print table format printf chart console scanner println line – TheServerSide.
Dále jsem si také zjišťoval informace o použití metod jako Thread.wait a dalších souvisejících technikách v Javě.