SPŠE Ječná

Informační technologie

Ječná 517, 120 00 Nové Město

Tower Defense

Filip Josef Hrouda

IT

2025

Obsah

1 Cíl práce

Cílem projektu je vytvořit strategickou hru ve stylu Tower Defense, ve které hráč brání přicházejícím vln nepřátel pomocí různých typů věží. Hráč musí strategicky rozmístit věže tak, aby efektivně zastavil nepřátelský postup. Každý typ věže má unikátní vlastnosti. Každá další vlna je těžší a silnější, což nutí hráče k plánování herní ekonomiky.

2 Popis hry

Tower Defense Game je strategická hra, kde hráč musí bránit svou základnu proti postupujícím vlnám nepřátel pomocí rozmístění obranných věží.

2.1 Algoritmus

Hlavní algoritmus hry:

Spawn nepřátel – Každá vlna obsahuje různý typ nepřátel, kteří postupují po předem určené cestě.

Útok věží – Pokud nepřítel vstoupí do dosahu věže, věž automaticky zahájí střelbu.

Poškození nepřátel – Nepřátelé mají různé hodnoty životů, pokud dosáhnou nuly, jsou odstraněni a hráč získá peníze.

Ztráta životů hráče – Pokud nepřítel dosáhne cíle, hráč ztrácí životy.

Konec hry – Hra končí vítězstvím, pokud hráč ubrání všechny vlny, nebo porážkou, pokud mu dojdou životy.

2.2 Postavy

Hra obsahuje dvě hlavní skupiny herních entit: věže, které slouží jako obranné jednotky, a nepřátele, kteří představují útočící jednotky.

2.3 Mechaniky

Hra obsahuje nakupování a pokládání věží na mapu, spouštění jednotlivých vln nepřátel a přepínání mezi samotnou hrou, lobby a tutoriálem.

Nakupování věží – Hráč může vybírat z dostupných typů věží a zakoupit je za herní měnu.

Pokládání věží na mapu – Zakoupené věže lze umístit na políčka kliknutím na dostupné pozice na mapě.

Spouštění vln nepřátel – Hráč může ručně zahájit vlnu nepřátel pomocí tlačítka "Start".

Přepínání mezi herními režimy – Hráč může přecházet mezi samotnou hrou, lobby a tutoriálem podle potřeby.

3 Systém requirements

Program byl vyvíjen v jazyce Java, konkrétně ve verzi Java 23. Pro jeho spuštění je tedy nutné mít nainstalované odpovídající JDK (Java Development Kit) verze 23. Kromě standardních knihoven Javy je v rámci testování využita knihovna JUnit Jupiter (součást JDK) a externí knihovna Mockito Core: 5.18.0 (Maven: org.mockito:mockito-core:5.18.0). Program lze spustit ve vývojovém prostředí IntelliJ IDEA.

4 Základní struktura

Program je navržen objektově, a jeho hlavní třídy spolu vzájemně komunikují za účelem správy různých herních funkcionalit. Níže jsou klíčové komponenty, které tvoří základ hry.

Hlavní třídy programu

Main – Slouží jako vstupní bod aplikace, spouští hlavní okno hry. MainWindow – Hlavní okno aplikace, používá CardLayout pro přepínání mezi lobby, tutoriálem a samotnou hrou. GamePanel – Hlavní herní plocha, která vykresluje mapu, nepřátele a věže, zpracovává herní logiku a umožňuje hráči stavět věže.

Herní manažery

GameLogic – Spravuje všechny manažery ve hře – aktualizuje nepřátele, věže, střely a kontroluje konec hry. WaveManager – Řídí postupné vlny nepřátel, aktivuje jejich spawn a zvyšuje obtížnost. EnemyManager – Spravuje nepřátelské jednotky, jejich pohyb, interakce s věžemi a vyhodnocení kolizí. BulletManager – Spravuje střely věží, aktualizuje jejich pohyb a kontroluje zásahy nepřátel. TowerManager – Spravuje věže, jejich stavbu, střelbu a vylepšení. MoneyManager – Udržuje herní ekonomiku, počítá zisky za zničené nepřátele a umožňuje nákup věží. LifeManager – Sleduje stav hráčových životů, pokud klesnou na nulu, hra končí.

Komunikace mezi komponentami

GamePanel komunikuje s GameLogic, která spravuje všechny manažery. GameLogic pracuje s WaveManager, EnemyManager, TowerManager a BulletManager pro řízení hry. TowerManager zpracovává akce věží, jejich střelbu a interakce s nepřáteli přes EnemyManager. WaveManager informuje EnemyManager o nových nepřátelích a kontroluje průběh jednotlivých vln.

5 Testovací data

WaveTest: Ověřuje správnou inicializaci vln (spawnování nepřátel, ukončení vlny po dosažení limitu, chování prázdné vlny).

WaveManagerTest: Kontroluje správnost správy vln, jejich číslování a správné vyhodnocení vítězství po dokončení poslední vlny.

MoneyManagerTest: Testuje operace s penězi – přidávání, útratu a validaci stavu účtu.

LifeManagerTest: Zajišťuje, že odečítání životů funguje správně a hra se ukončí při překročení limitu.

GameLogicTest: Ujistí se, že běžná aktualizace herní logiky nevede k nechtěnému završení hry.

6 Uživatelská příručka

Program se ovládá jenom pomocí tlačítek. Po spuštění se zobrazí hlavní menu, odkud se může hráč přesunout do tutoriálu, kde se dozví základní informace a mechaniky hry, nebo rovnou začít hru. Na pravé straně herního rozhraní se nachází obchod s věžemi, které lze nakupovat a pokládat na mapu.

Hráč si nejprve vybere typ věže kliknutím na její tlačítko v obchodě. Poté ji může umístit na volné místo na mapě, což způsobí automatické odečtení peněz z jeho herního rozpočtu. Je důležité si uvědomit, že jakmile je věž položena, nelze ji odstranit ani přesunout.

Vlny nepřátel lze spouštět tlačítkem "Start". Po jeho stisknutí se nepřátelé začnou pohybovat po vyznačené trase směrem zleva doprava. Pokud se jim podaří dosáhnout cíle, hráč přijde o určité množství životů. Pokud ztratí všechny životy, hra končí porážkou. Pokud se mu podaří ubránit všechny vlny nepřátel, dosáhne vítězství.

7 Závěr

Už na začátku vývoje jsem musel kvůli špatné objektové struktuře celý program smazat a začít znovu. Tentokrát jsem si před psaním kódu nejprve sepsal jednotlivé třídy, enumy atd. a přibližně nastínil, jak spolu budou souviset. Další problém nastal při implementaci pohybu nepřátel na mapě. Zdržel jsem se tím několik hodin, a algoritmus stále nefungoval správně. Občas se nepřátelé pohybovali vertikálně, jindy se ani nenačetli.

Po vyřešení tohoto problému šel vývoj dobře až do chvíle, kdy jsem začal implementovat střelbu věží na nepřátele. Opět jsem se zasekl stejně jako u pohybu nepřátel. Všechny výpočty a logické souvislosti pro mě byly dost složité. Po těchto komplikacích jsem začal mít problém stihnout dokončit hru včas. Čas se začal krátit a já měl zatím jen hotovou kostru herní logiky bez vizuální stránky.

Kreslení obrázků mi také zabralo nějaký čas, ale nakonec jsem vše stihl. Jediné, co jsem už nestihl dodělat, bylo odstranění věží po jejich umístění na mapu, což jsem kvůli časové tísni musel vynechat. Celkově mi ale projekt hodně pomohl – naučil jsem se nové věci, jako například použití třídy Math pro různé výpočty, práci s MouseListener, načítání souborů a obrázků, správu dvourozměrného pole nebo využití CardLayoutu.

Celkově hodnotím projekt jako velmi náročný, ale zároveň přínosný.

8 Zdroje

Důležité použité knihovny

V projektu byly použity následující knihovny:

- javax.swing – pro uživatelské rozhraní (JFrame, JPanel, JButton)
- java.awt – pro práci s grafikou (Graphics, Color)
- java.util – kolekce jako ArrayList
- java.io – pro načítání souborů (např. konfigurace věží, mapy)
- Math – pro výpočty vzdáleností, úhlů a další matematické operace

Externí zdroje použité při vývoji

Baeldung. Výpočet vzdálenosti mezi dvěma body v Javě. Dostupné z: https://www.baeldung.com/java-distance-between-two-points [cit. 2025-05-29].

GeeksforGeeks. Přehled metod třídy Math v Javě. Dostupné z: https://www.geeksforgeeks.org/java-math-class/ [cit. 2025-05-29].