Technická Dokumentace

1. Úvod

Tato dokumentace poskytuje podrobný přehled o herním enginu, který byl vytvořen v jazyce Java s využitím knihovny JavaFX. Tento engine je navržen pro 2D platformové hry a zahrnuje základní funkce jako je pohyb hráče, gravitace, kolize, interakce s objekty a správa herního stavu.

2. Uživatelský manuál

Následuje kompletní a podrobný uživatelský manuál pro použití herního enginu. Před použitím a jakoukoli snahou tvořit hru je silně doporučeno tuto sekci pečlivě nastudovat.

2.1. Hlavní nabídka

Pro změnu hlavní nabídky se ve spodku adresáře resources nachází menu.css. Tento soubor kaskádových stylů lze použít na měnění vzhledu elementů v hlavní nabídce. Hlavní nabídka je definovaná souborem menu-view.fxml též uložený ve spodku adresáře resources.

V hlavní nabídce jsou pevně dány následující elementy:

- Label: Tento label má v sobě nadpis hlavní nabídky.

- Tlačítko Start Game: Spustí hru od začátku, začne nový průchod vaší hrou.

- Tlačítko Load Game: Spustí hru z naposled uložené pozice.

- Tlačítko Quit Game: Ukončí aplikaci a zavře její okno.

Elementy tlačítek musí zůstat v stejné pro správný chod aplikace, uživatel může pouze měnit jimi zobrazovaný text.

2.2 Ukládání textur a assetů

Všechny soubory požívané pro vaši hru, ať už v menu nebo používané v level souboru, musí být uloženy na spodku adresáře resources

2.3. Tvorba vlastní hry (Levelů)

Užití herního enginu spočívá v tvorbě vlastní level souborů, respektive souborů typu JSON. Tyto soubory umožňují jednoduché načítání do interaktivních objektů, se kterými pracuje herní logika a grafické rozhraní. Také se v nich uchovává konfigurace některých herních pravidel.

2.3.1. Základní konvence k pojmenování level souborů:

- všechny level soubory jsou uložené ve složce data/levels

- První level neboli začátek vaší hry musí být vždy označen start.json, tento soubor je vždy načten po kliknutí na výše zmíněné tlačítko Start Game.

- Soubor last_checkpoint, uložen ve složce data/saves, slouží pro uložení posledního dosaženého levelu tedy když hráč dohraje level uloží se pozice na začátku levelu dalšího.

- Do souboru last_session.json, uložen ve složce data/saves, se ukládá poslední pozice hráče při odchodu do menu, tento soubor se vždy načte při stisknutí tlačítka Load Game. V případě smrti hráče se soubor přepíše na stejné hodnoty uložené v last_checkpoint.json, toto je z důvodu, aby hráč si nemohl ukládat záchytné body, kde se mu zlíbí a také, aby se hráč nemohl zacyklit v nekonečné smyčce smrti.

2.3.2. Specifikace atributů levelu

Level má několik povinných atributů:

"player": (specifikace hráče)

"name": jméno daného levelu

"hazards": list (specifikací jednotlivých nebezpečí) tedy list věcí co ubližují hráči

"npcs": list (specifikací jednotlivých nehratelných postav) tedy list postav se kterými může hráč interagovat, ale ne za ně hrát

"obstacles": list (specifikací jednotlivých překážek) tedy list pevný objektů

"collectibles": list (specifikací jednotlivých sběratelných předmětů) tedy list věcí co může hráč sebrat

"nextLevelFileName": jméno souboru následujícího levelu, pokud má daný level být poslední je toto pole zapsáno jako "end" anebo null.

"gravityForce": číslo, které specifikuje, jak moc na hráče působí gravitace neboli o jakou vzdálenost je hráč posunut dolů každý herní tick, pokud padá. Optimální hodnota je tak 2–8, záleží, ale na velikosti objektů a hráče ve vaší hře.

"endPoint": (specifikace herního objektu) který slouží jako bod kde končí level, pokud hráč se dotkne a splňuje náležitosti přepne se na nový level

"required": list (specifikací itemů) tento list slouží jako výpis předmětů, co hráč musí mít u sebe, když chce postoupit do dalšího levelu. Mělo by být zmíněno, že toto pole nepodporuje více od jednoho předmětu tedy nedá se vytvořit požadavek typu seber n počet daného předmětu.

"backgroundFileName": název souboru obrázku na pozadí levelu.

2.3.3. Specifikace herních objektů

Každý následující herní objekt má společné atributy.

Společné atributy herních objektů:

- "coords": souřadnice x, y, tedy momentální poloha, na které se objekt nachází.

- "textureFileName": název souboru obrázku, který slouží jako textura objektu.

Hráč ("player"):

- "inventory": (specifikace inventáře) tedy prostor věcí, co si hráč nese s sebou za předměty(itemy).

- "walkSpeed": číslo definující rychlost hráče neboli vzdálenost, o kterou je hráč posunut každý tick, když se hýbe doleva nebo doprava. Optimální hodnota je 3–7 zase záleží na velikosti objektů ve vaší hře.

- "jumpHeight": číslo specifikující, jak vysoko dokáže hráč skočit. Optimální hodnota je 40–150 opět záleží, jak vypadá vaše hra.

- "jumpForce": číslo specifikují rychlost s jakou hráč se pohybuje nahoru při skoku neboli o kolik se posune každý tick když zrovna skáče.

- "healthBar": (specifikace ukazatele zdraví) tedy objekt, který sleduje hráčovo zdraví a zajištuje informování hráče o jeho zdraví.

Překážka ("obstacle"): nemá jiné atributy krom těch společných. Slouží jako pevný objekt v levelu na, kterém může hráč stát a nemůže jím projít.

Nebezpečí ("hazard"):

- "damage": číslo, kolik se má ubrat hráči životů při kontaktu(kolizi). Může být nastaveno i na zápornou hodnotu, a tak se dá vytvořit objekt,

    co léčí hráče dotekem.

Sbíratelné předmět ("collectible"):

- "item": (specifikace předmětu) tedy ten předmět který hráč získá sebráním.

Nehratelná postava ("npc"):

- "quests": list (specifikací úkolů) tedy list úkolů, co postava má pro hráče, které musí splnit pro postup do dalšího levelu nebo pro získání odměnného předmětu.

- "commonDialog": je hláška, kterou postava hráči řekne (napíše do dialogového okna), když hráč s postavou interaguje a ona již nemá žádný úkol pro hráče. Něco jako pozdrav nebo věta jako "To máme, ale hezké počasí".

Bod konce levelu ("endPoint"): Po každé, když se hráč dotkne bodu konce zkontrolují se, jestli hráč má u sebe v inventáři všechny předměty, které jsou uvedeny v poli required a jestli splnil veškeré povinné úkoly (viz sekce 2.3.4) od nehratelných postav v levelu.

Tento objekt může být definován jako kterýkoli herní objekt, pokud je definován jako nebezpečí nebo jako hráč anebo sběratelný předmět, jeho funkce budou přepsány tedy nebude mít funkce těchto objektů nicméně ani nemá smysl definovat endpoint jako

nebezpečí nebo jako hráče (ovladatelný hráč může být pouze jeden a je nesmyslné, aby bod konce ubližoval hráči), kdežto koncový bod jako sběratelný předmět má využití jako objekt, kterým se dá procházet. Nehratelná postava a překážka si zachovají svoje funkce i když jsou koncovým bodem.

Takže se dá zakončit level pomocí konverzace s nehratelnou postavou nebo konce levelu může být fyzická překážka, kterou se dá projít do dalšího levelu, po získání specifických předmětů nebo po splnění úkolů od postav.

2.3.4 Specifikace zbylých logických a datový herních struktur

Inventář ("inventory"):

- "slotCount": počet políček v inventáři neboli kolik předmětů se vejde do inventáře.

- "rowSize": počet políček v řádku inventáře. Specifikuje, jak se vypadá mřížka inventáře při zobrazení.

- "rowCount": počet řádků inventáře. Specifikuje, jak se vypadá mřížka inventáře při zobrazení.

- "items": list (specifikací předmětů) tedy list toho co je v inventáři za předměty.

- "fullSlotCount": počet zaplněných políček neboli kolik předmětů je v inventáři.

- "displayWidth": šířka zobrazeného box při otevření inventáře.

- "displayHeight": výška zobrazeného box při otevření inventáře.

- "emptySlotTextureFileName": název souboru obrázku, který slouží jako vzhled prázdného políčka inventáře.

Předmět ("item"):

- "name": název předmětu. Důležité: Podle tohoto názvu se porovnávají předměty tedy pokud se rovnají názvy, tak se jedná o jeden a tentýž předmět. Toto je podstatné pro plnění úkolů a vylepšování předmětů.

- "itemType": Specifikuje typ itemu, mezi typy patří:

  - healing: předmět mění zdraví hráče, když ho hráč použije.

  - quest: předmět je určen pro splnění úkolu nebo je potřebný k postupu do dalšího levelu.

- upgrade: tento předmět lze zkombinovat s jiným předmětem a tím daný předmět vylepšit.

- health_upgrade: předmět, co mění maximální zdraví hráče.

- "statModifier": číslo, o které se mění daný stav hráče, závisející na typu daného předmětu. např. když je 1 a typ předmětu healing při použití se hráč ozdraví o jeden bod zdraví.

- "upgradableBy": specifikace předmětu, kterým se dá daný předmět vylepšit. V případě že takový předmět není je null. Pokud je předmět sám vylepšením tak toto pole specifikuje předmět, který je výsledkem vylepšení.

- "iconFileName": název souboru obrázku, který slouží jako ikona zobrazení uvnitř inventáře. Důležité zmínit je, že taková ikona by měla jako pozadí obsahovat texturu prázdného políčka v inventáři pro správné grafické zobrazení.

Úkol ("quest"):

- "questItem": specifikace předmětu, který je požadován pro splnění úkolu.

- "Dialogs": list dialogů neboli textů použitých jako jednotlivé dialogy postavy.

- "indexOfChange": index dialogu v listu dialogů, kdy postava už vysvětlila podstatu úkolu a nyní očekává požadovaný předmět k dokončení úkolu

- "dialogIndex": index momentálního dialogu, který se má zobrazit v dialogovém okně.

- "questState": aktuální stav úkolu, může být:

- "NOT_STARTED": Jiný, než tento stav by měl být vždy nastaven ve specifikaci levelu, který není rozehraný. Indikuje to, že buď hráč s postavou s daným úkolem ještě vůbec nemluvil nebo že postava ještě nedovysvětlila podstatu úkolu, tedy ještě nečeká, že hráč přinese požadovaný předmět.

- "AWAITING_ITEM": postava již vysvětlila podstatu úkolu a už čeká, že hráč jí přinese požadovaný předmět.

- "COMPLETED": úkol byl splněn a odstraněn s listu úkolů, co má postava.

- "sideQuest": je boolean hodnota true/false, která říká, jestli je úkol nepovinný pro dokončení levelu.

- "reward": specifikace předmětu, který je obdržen hráčem jako odměna za splnění úkolu. Pokud není odměna za splnění je toto pole null.

Ukazatel zdraví ("healthBar"):

- "health": číslo, momentální počet bodů zdraví.

- "maxHealth": číslo, maximální počet bodů zdraví.

- "heartTextureFileName": název souboru obrázku, který slouží jako textura reprezentující jeden bod zdraví.

- "emptyHeartTextureFileName": název souboru obrázku, který slouží jako textura reprezentující chybějící bod zdraví.

2.4 Dodatečné poznatky o herním enginu

- V případě, že hráč se dostane mimo herní pole neboli, že vypadne ze světa je nastaveno, že automaticky umře. Toto je provedeno tak, že když je hráč podle y souřadnice pod hranicí -1080 prostě se zabije.

- Při jakýchkoli pochybách při tvorbě levelu je možno se inspirovat přiloženými levely hry The last man's journey a použít je jako template.

3. Použité technologie

Závislosti

Projekt využívá následující závislosti, které jsou definovány v souboru pom.xml:

JavaFX: Knihovna pro vytváření grafických uživatelských rozhraní.

javafx-controls: Verze 21

javafx-fxml: Verze 21

JUnit: Framework pro testování v jazyce Java.

junit-jupiter-api: Verze 5.10.0

junit-jupiter-engine: Verze 5.10.0

SLF4J: Simple Logging Facade for Java.

slf4j-api: Verze 2.0.13

logback-classic: Verze 1.4.12

Lombok: Knihovna pro odstranění opakujícího se kódu pomocí anotací.

lombok: Verze 1.18.32

Mockito: Knihovna pro tvorbu mock objektů při testování.

mockito-core: Verze 3.12.4

Jackson: Knihovna pro práci s JSON daty.

jackson-databind: Verze 2.13.4.1

Aether: Knihovna pro správu závislostí.

aether-impl: Verze 1.7

Další použité technologie

Java 21: Projekt využívá nejnovější verzi Javy pro plné využití nových jazykových funkcí a optimalizací.

Maven: Nástroj pro správu projektů a závislostí.

4. Struktura projektu

Projekt je strukturován podobně jako MVC (Model-View-Controller) architektura:

Models: Obsahují třídy, které reprezentují herní objekty a logiku.

Views: Obsahují třídy pro vykreslování grafiky a uživatelského rozhraní.

Controllers: Obsahují třídy, které řídí herní logiku a interakci mezi modely a pohledy.

Hlavní balíčky

org.tomek.engine: Obsahuje hlavní herní třídy a herní smyčku.

org.tomek.engine.controls: Obsahuje třídy pro ovládání hry a uživatelských vstupů.

org.tomek.engine.enums: Obsahuje výčtové typy používané v projektu.

org.tomek.engine.exceptions: Obsahuje třídy pro vlastní výjimky.

org.tomek.engine.gameobjects: Obsahuje třídy reprezentující herní objekty (hráč, NPC, překážky, sběratelské předměty).

org.tomek.engine.views: Obsahuje třídy pro vykreslování obrazovek (menu, pauza, smrt).

5. Architektura game enginu

Herní engine je navržen tak, aby efektivně spravoval herní smyčku, vykreslování grafiky, interakci s uživatelem, herní logiku a načítání levelů z JSON souborů.

Herní smyčka

Hlavní herní smyčka běží v samostatném vlákně a aktualizuje herní stav každých 16 ms (přibližně 60 FPS). Smyčka zajišťuje:

Aktualizaci pozice hráče na základě uživatelských vstupů (pohyb doleva, doprava, skok).

Aplikaci gravitace na hráče.

Kontrolu kolizí s herními objekty.

Kontrolu dosažení konce úrovně.

Zpracování úmrtí hráče a načtení posledního uloženého checkpointu.

Vykreslování grafiky

Grafika je vykreslována na plátno (Canvas), které je součástí JavaFX. Hlavní třída pro vykreslování je Game, která zajišťuje:

Vykreslení pozadí.

Vykreslení všech herních objektů (hráč, překážky, NPC, sběratelské předměty).

Aktualizaci uživatelského rozhraní (zdraví hráče, inventář).

Interakce s uživatelem

Uživatelské vstupy jsou zpracovávány pomocí událostí klávesnice:

Pohyb vlevo/vpravo (KeyCode.A, KeyCode.D, KeyCode.LEFT, KeyCode.RIGHT).

Skok (KeyCode.SPACE, KeyCode.UP).

Otevření inventáře (KeyCode.TAB).

Pauza (KeyCode.ESCAPE).

Interakce s objekty (KeyCode.E).

Herní logika

Herní logika zahrnuje správu herních objektů, kontrolu kolizí, správu úrovní a interakce s herními prvky.

Načítání levelů z JSON souborů

Úrovně jsou načítány z JSON souborů pomocí knihovny Jackson. Každý level obsahuje definice herních objektů, jejich pozice a další parametry.

6. Technické specifikace tříd a struktur herního enginu

6.1. Shrnutí herních prvků

Hráč

Hráč je hlavní postavou ve hře. Má atributy jako pozice, rychlost, zdraví a inventář. Hráč může:

Pohybovat se vlevo a vpravo.

Skákat.

Sbírat předměty.

Interagovat s NPC.

Inventář

Inventář hráče obsahuje sbíratelné předměty, které může hráč během hry získat. Inventář je zobrazen v uživatelském rozhraní a lze jej otevřít pomocí klávesy TAB.

NPC (Nehratelné postavy)

NPC jsou postavy, se kterými může hráč interagovat. Mohou poskytovat dialogy, úkoly nebo informace.

Překážky

Překážky jsou statické objekty, které blokují pohyb hráče. Hráč se s nimi může srazit, ale nemůže je projít.

Sběratelské předměty

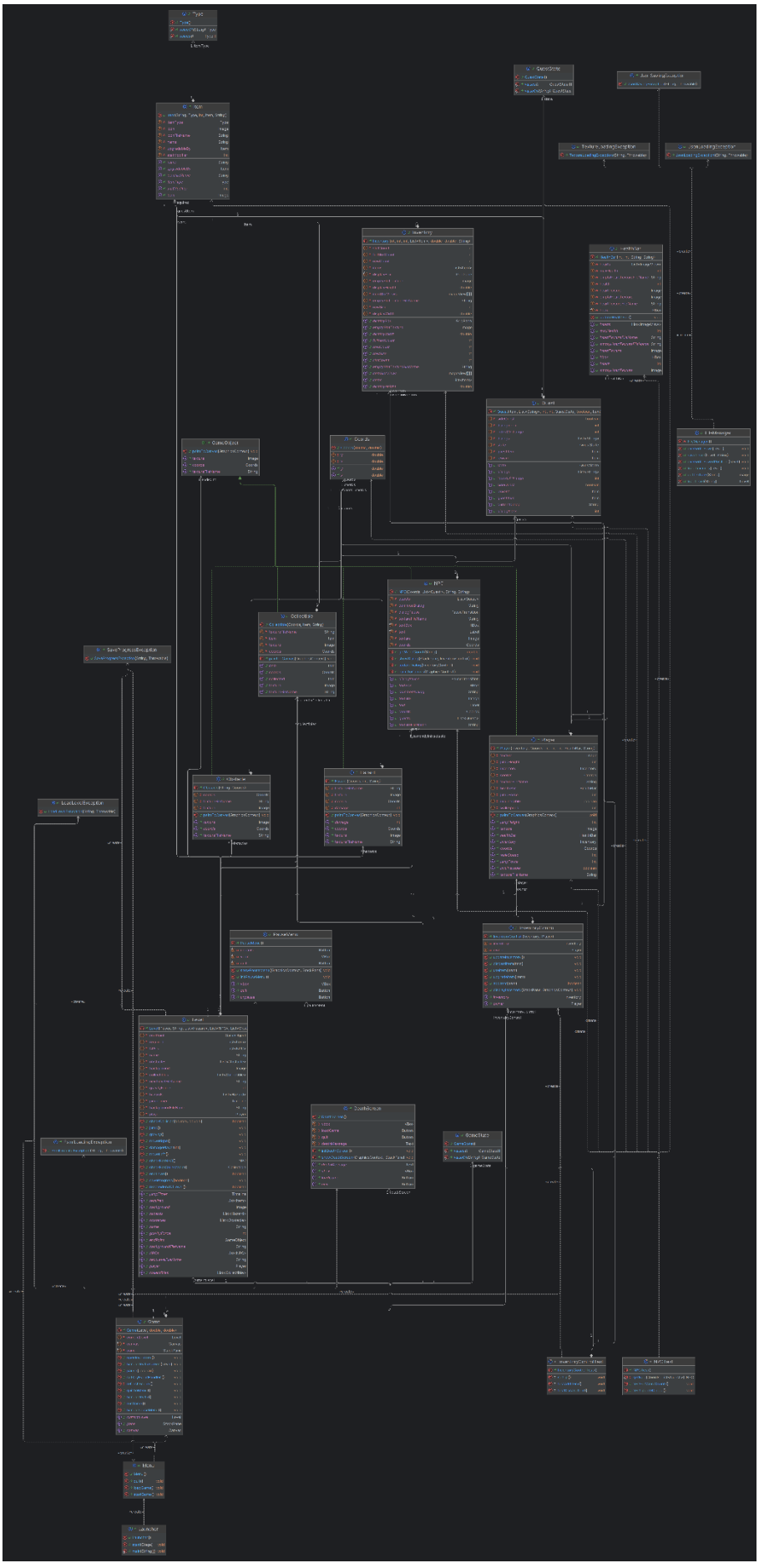
Sběratelské předměty jsou objekty, které hráč může sbírat a přidávat do svého inventáře. Tyto předměty mohou mít různé efekty na hru (zvýšení zdraví, speciální schopnosti).

Nebezpečí

Nebezpečí jsou objekty, které mohou hráči ublížit, například pasti nebo nepřátelé. Hráč může s těmito objekty interagovat, což může vést ke ztrátě zdraví.

6.2. Diagram tříd herního enginu

Pro podrobnější pohled do fungování a struktury enginu, jeho tříd a jejich závislostí na sobě lze prostudovat ULM diagram tříd (přiložen v souborech):



6.3. Speciální výjimky

LoadLevelException (runtime):

- nastává, když se nepodaří deserializace level soubor obsahuje syntaktickou chybu nebo chybějící nebo neplatné hodnoty.

SaveProgressException (runtime):

- nastává, když se nepodaří serializovat level objekt do souboru.

JsonLoadingException (IO):

- nastává, když json level soubor není nalezen nebo je nečitelný.

JsonSavingException (IO):

- nastává, když se nepodaří otevřít nebo vytvořit json level soubor pro zápis.

FxmlLoadingException (runtime):

- nastává, když se nepodaří načíst fxml soubor tedy menu.

TextureLoadingException (runtime):

- nastává, když při procesu načítání textur nenajde nebo je nečitelný daný obrázkový soubor.

6.4 Testování a ladění

Pro zajištění správného fungování herního enginu jsou implementovány jednotkové testy, které ověřují funkčnost různých komponent a tříd. Testování je klíčovou součástí vývoje, protože pomáhá identifikovat a opravovat chyby, zajišťuje stabilitu kódu a umožňuje snadnější údržbu projektu.

InventoryControlTest testuje základní funkce třídy InventoryControl, která spravuje inventář hráče.

NPCTest ověřuje funkčnost třídy NPC, která představuje nehratelné postavy ve hře.

Jednotkové testy jsou implementovány pomocí knihoven JUnit 5 a Mockito, které umožňují tvorbu a testování mock objektů. Tyto testy zajišťují, že klíčové funkce herního enginu fungují správně a podle očekávání. Testy jsou důležitým nástrojem pro identifikaci a řešení chyb a pomáhají udržovat vysokou kvalitu kódu během vývoje herního enginu.

7. Závěr

Tato dokumentace poskytuje ucelený přehled o herním enginu, jeho architektuře a technických specifikacích. Engine je navržen tak, aby byl snadno rozšiřitelný a udržovatelný. Díky využití moderních technologií a osvědčených postupů nabízí robustní základ pro tvorbu 2D platformových her.

Vytvořil:

- Petr Jan Tomek