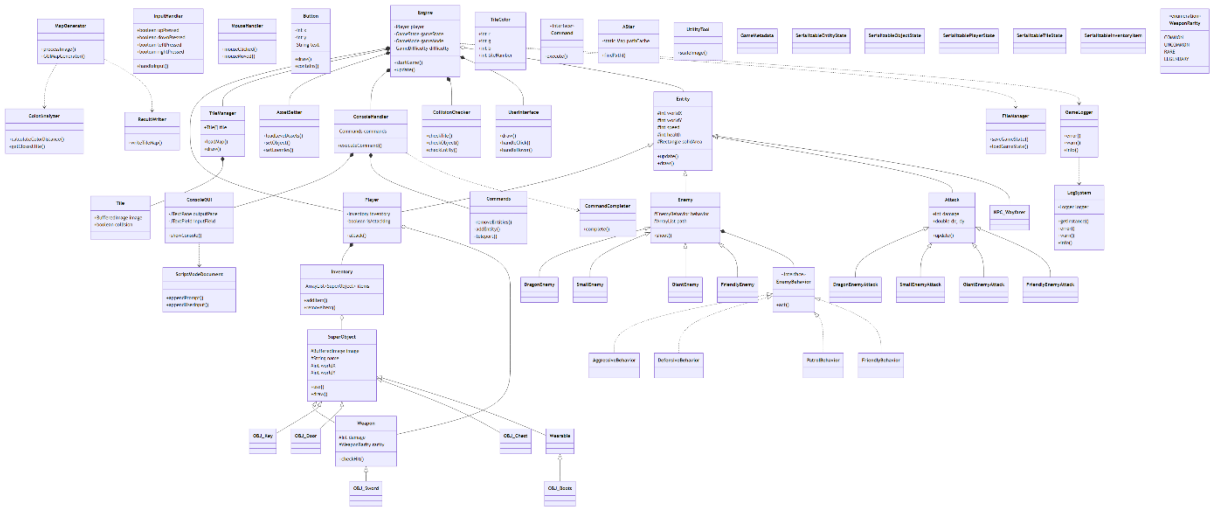
Horváth Tamás – IYE7V8

Dokumentáció

Áttekintés:

A projekt egy 2D játékmotort és a köré épült infrastruktúrát valósítja meg. A rendszer egy tile alapú, amely tartalmaz ellenségeket, harci mechanizmust, tárgykezelést és mentés/betöltés funkcionalitást. A játék két fő játékmódot támogat: a történet módot és az egyedi pályagenerálást.

Osztálydiagram:



Package szerkezet:

A projekt több logikai egységre tagolódik, amelyek külön csomagokban helyezkednek el. A fő csomagok közé tartozik a main csomag, amely a játékmotor alapvető komponenseit tartalmazza, az entity csomag, amely a játékban szereplő entitásokat (játékos, ellenségek, NPC-k) foglalja magába, valamint az object csomag, amely a játékbeli tárgyak kezeléséért felelős. A pályakezelésért a tile és map csomagok felelnek, míg a játékállapot mentését és betöltését a serializable csomag végzi. A fejlesztői funkciókhoz tartozó konzol kezelését a main.console csomag biztosítja, a naplózási funkciókat pedig a main.logger csomag tartalmazza.

Öröklődési hierarchia:

A játék objektumrendszere három fő hierarchiára épül. Az első az Entity hierarchia, amelynek gyökere az absztrakt Entity osztály. Ebből származik le a Player osztály, amely a játékos karaktert reprezentálja, valamint az absztrakt Enemy osztály, amely az ellenségek alapját képezi. Az Enemy osztályból származnak le a különböző ellenségtípusok: SmallEnemy, GiantEnemy, DragonEnemy és FriendlyEnemy. Az NPC_Wayfarer osztály szintén közvetlenül az Entity osztályból származik.

A második jelentős hierarchia a SuperObject hierarchia, amelynek alapja az absztrakt SuperObject osztály. Ez az osztály adja a játékban található tárgyak alapját. Ebből származik le a Weapon absztrakt osztály, amelynek konkrét megvalósítása az OBJ_Sword, valamint a Wearable osztály, amelyből az OBJ_Boots származik. További közvetlen leszármazottak az OBJ_Key, OBJ_Door és OBJ_Chest osztályok.

A harmadik hierarchia az Attack rendszer, amely a különböző támadástípusokat definiálja. Az alap Attack osztályból származnak a specifikus támadástípusok: DragonEnemyAttack, SmallEnemyAttack, GiantEnemyAttack és FriendlyEnemyAttack.

Osztályok közötti kapcsolatok:

A játék központi eleme az Engine osztály, amely tartalmazza a főbb komponenseket: Player, TileManager, AssetSetter, InputHandler, ConsoleHandler és CollisionChecker példányokat. A Player osztály rendelkezik egy Inventory objektummal, amely a loot kezelésért felelős. Az Enemy osztályok a stratégia tervezési mintát követve használják az EnemyBehavior interfészt a különböző viselkedések megvalósítására. Az AssetSetter felelős az Entity és SuperObject példányok létrehozásáért és kezeléséért.

Engine csomag:

Az Engine osztály a játék központi vezérlőegysége, amely kezeli a játékciklust és az állapotokat. Főbb metódusai között szerepel a startGameThread(), amely elindítja a játék fő szálát, az update(), amely a játékállapot frissítéséért felel, valamint a paintComponent(), amely a grafikus megjelenítést végzi. A startGame() metódus új játék indításakor inicializálja a szükséges komponenseket, míg a checkLevelCompletion() a pálya teljesítését ellenőrzi.

Az AssetSetter osztály felelős a játékobjektumok kezeléséért. A loadLevelAssets() metódus betölti az aktuális pályához tartozó objektumokat, a createObject() és createEnemy() metódusok pedig új játéktárgyakat, illetve ellenségeket hoznak létre. A spawnItemFromChest() metódus a ládákból való véletlenszerű tárgygenerálást valósítja meg.

A CollisionChecker osztály az ütközésvizsgálatért felel. Négy fő ellenőrzési metódust tartalmaz: a checkTile() a tile-okkal való ütközést, a checkObject() az objektumokkal való ütközést, a checkEntity() az entitásokkal való ütközést, a checkPlayer() pedig specifikusan a játékossal való ütközést vizsgálja.

Entity csomag:

A Player osztály a játékos karaktert valósítja meg. Az update() metódus frissíti a játékos állapotát, az attack() metódus a támadás végrehajtásáért felel. A setDefaultValues() az alapértékek beállítását, a getPlayerImage() pedig a játékos grafikájának betöltését végzi.

Az Enemy absztrakt osztály az ellenségek alapvető viselkedését definiálja. Az update() metódus az ellenség állapotának frissítését, a shoot() a lövés végrehajtását, a followPath() az útvonalkövetést valósítja meg. Az initializeBehavior() metódus az adott ellenségtípus viselkedésének inicializálásáért felel.

Felhasználói kézikönyv

A játék irányítása az alábbi billentyűkkel történik:

- W: Felfelé mozgás

- A: Balra mozgás

- S: Lefelé mozgás

- D: Jobbra mozgás

- E: Támadás

- F: Item rotáció az inventoryban

- Q: Tárgy eldobás

- ESC: Játék szüneteltetése

A játék két fő játékmódot kínál: a történet módot és az egyedi pálya módot. A történet módban előre definiált pályákon kell végig haladni, ahol a nehézség fokozatosan növekszik. Az egyedi pálya módban lehetőség van saját készítésű pályák betöltésére és szabadon bejárható területek létrehozására.

A nehézségi szintek a következők:

- Könnyű: Egyszerű rajta a játék

- Közepes: Normális nehézségű a játék

- Nehéz: Alig teljesíthető a játék

- Lehetetlen: Lehetetlen kijátszani

Az inventory rendszer maximum három tárgy tárolását teszi lehetővé, amelyek között az F billentyűvel lehet rotálni. A játékos mindig az első slotban lévő elem hatását kapja meg. A használható tárgyak között szerepel a kulcs az ajtók nyitásához, fegyverek a támadáshoz és a cipők a gyorsabb mozgáshoz.

A játékban négy különböző ellenségtípus található:

- Small Enemy ellenség: gyors, nem sebez sokat

- Giant Enemy ellenség: lassú, ez sebzi a legtöbbet

- Dragon Enemy ellenség: kiegyensúlyozott

- Friendly Enemy ellenség: a játékost segíti a harcban

A fejlesztői konzol a Q gombbal érhető el szünet módban, ahol különböző parancsok használhatók a játék vezérléséhez és módosításához. A mentés és betöltés funkció elérhető mind az ESC menüből, mind a konzolból a megfelelő parancsokkal.

A játék mentési és betöltési rendszere többféle módon is elérhető. Mentést kezdeményezhetünk az ESC menüben található "Save Game" opcióval, vagy a konzolban kiadott "save" paranccsal vagy az ’o’ betű lenyomásával a szünet módban. Mindkét esetben meg kell adnunk egy fájlnevet a mentés azonosításához. A játékállás betöltésére szintén több lehetőségünk van: használhatjuk a főmenü "Load

Game" opcióját, az ESC menü megfelelő gombját, vagy a konzol "load" parancsát vagy az ’l’ betű használatával a szünet módban.

A fejlesztői konzol számos parancsot tartalmaz, amelyekkel lehet változtatni a játék aktuális állapotán. A teljes parancslistát a "help" paranccsal tekinthetjük meg. A "set" és "get" parancsokkal különböző játékértékeket módosíthatunk és kérdezhetünk le. Az "add" paranccsal új objektumokat helyezhetünk el a pályán, míg a "remove" paranccsal eltávolíthatjuk azokat. A "teleport" parancs lehetővé teszi a játékos azonnali áthelyezését a pálya bármely pontjára.

A játék egyedi script rendszert is tartalmaz, amely lehetővé teszi összetett parancssorozatok létrehozását és végrehajtását. Script készítéséhez először a "make" parancsot kell használnunk, majd megadhatjuk a végrehajtandó parancsokat. A script írását az "end" paranccsal fejezhetjük be. A létrehozott scriptet később a "script" paranccsal futtathatjuk.

A térképgenerálási rendszer lehetővé teszi egyedi pályák létrehozását PNG fileokból. A generátor a bemeneti képet feldolgozza és a színek alapján megfelelő játékelemeket helyez el a pályán. A világos zöld színek füves területekké, a kék színek vízzé, a szürke árnyalatok pedig falakká alakulnak. A generált pályák automatikusan mentésre kerülnek és később újra betölthetők.