

Fejlesztői útmutató összefoglalás

Game

A játék a Game class-ban valósul meg itt van a main rész itt indul a threadel a játék és itt frissítjük a játékot, itt váltunk a különböző Scenek között, amik más-más funkciókért felelősek.

Scenek

A játékban 6 Scene van ezek kötött váltva más más megjelenítést és más dolgokat tud a játékos csinálni, ezek közül 3 fontos.

A Menu innen érünk el minden más Scene-t kivéve Victory és Game Over. A Playing itt történik a játék itt a handlereket használva kezeljük a játékot és a különböző gombokat. Az Edit itt tudunk pályát szerkeszteni elmenteni itt a TileHandler-t alkalmazzuk a pálya megalkotásához.

Handlerek és azok objectjei,eventsei és ellenik

Összesen 5 handler van TowerHandler, EnemyHandler,WaveHandler, ProjectileHandler és TowerHandler. Ezek közül az első négyet a Playingben használjuk az utolsót az Editben. A Handlerok a különböző eventek objectek és enemyket kezelik hozzák létre itt vannak megvalósítva a különböző fontos függvények pl: az ellenfelek mozgatás, az ütközés detektálása és lekezelése, a megfelelő object létrehozása, az objectek törlése és a grafikus és logikai frissítések és az inputok kezelése.

Az EnemyHandler az Enemy típusokat kezeli le, az Enemy osztály abstract és 5 leszármazottja van Mordo, Mosquito, Shaman, Spriggan és Warrior ezek a különböző ellenség típusok. Ezekhez kollektíven Enemy típusra írt függvényeket definiálunk az EnemyHandlerbe a fentebb említett mozgatás, sebzés, új iránykérés, grafikus és logikai frissítés, grafikus betöltés, törlés / kill enemy.

Az ellenfelek létrehozásáért a WaveHandler felelős hiszen különböző hullámokba szervezzük az ellenségeket. Itt különböző timerek és huzzájuk függvények vannak, illetve a hullámokat létrehozó függvény. Hullámokból jelenleg 1 típus van, ami csak egy enemyList így a WaveHandlerben töltjük fel őket.

A TowerHandler felelős a tornyok létrehozásáért, tornyok grafikus és logikai frissítésért és az ellenségek megtámadásáért. Tornyokból több típus is van, de nincs külön osztály minden egyes típusra, hanem egy Constant osztályban vannak definiálva az értékeik ezeket kérjük le onnan és a megfelelő Type alapján inicializáljuk a Tornyt.

A Tornyok különböző projectile-okat lőnek ki ezt a ProjectileHandler kezeli le itt hozza létre a projectileokat, indítja el megfelelő irányba azokat, ellenőrzi, hogy talált-e el ellenséget és a különböző projectile típusokat kezeli le, ezen felül logikai és grafikus frissítésért felelős és betölti a projectileok megfelelő képét. A projectileok ugyan úgy vannak megoldva, mint a tornyok a Constantsok közül töltjük be a megfelelő típust, amivel inicializáljuk a projectile-t.

A TileHandler felelős a pálya megfelelő megalkotásáért itt töltjük fel a megfelelő Tile-okkal a pályát és kirajzoljuk azokat. A Tileokatis típus alapján különböztetjük meg és az alapján inicializáljuk.

UI barok

Ezenfelül két fontos UI osztály van az egyik a Playingben az ActionBar itt a különböző gombok és a játékos által használt elemeket kezeljük pl: Játékmegállítás, toronylerakás fejlesztés. A másik

az Editben van a ToolBar itt a különböző gombok és Tileok lerakása, azaz a pálya szerkesztéséért felelős függvények vannak.

Helpz

Fontos még hogy a helpz packageben vannak statikus függvények, amik a konstansoktól a pálya betöltése és mentését segítő függvényeken át lévő osztályok és függvények vannak benne.