

Space Invaders

Tervezői megfontolások

A játékot 5 ablakra osztottam, amik a játék fő részeit mutatják. Az ablakok közti mozgást egy switch segítségével valósítom meg, ami egy enum értékeit figyeli. Ezek a menu, load, game, gameover, win értékek lehetnek. Ezt egy AnimationTimer handler függvényében valósítottam meg, hogy a játék folyamatosan fusson.

A játék főbb entitásai a játékos, az invaderek, és a lövedékek. Mivel a játékos és az invader sok mindenben hasonló, létrehoztam egy Figure osztályt, amiből leszármaztattam a Playert és az Invadert. Ugyan így tettem a Bullettel, amiből leszármazik a PlayerBullet és az EnemyBullet, mivel ezek is hasonlóan működnek, csak a mozgásuk iránya és sebessége, meg a képük más.

Ahhoz, hogy az invaderek csoportját is kezelhessem, létrehoztam egy Invaders osztályt, ami például a csoportos balra, illetve jobbra mozgást valósítja meg, valamint olyan függvényeket, amiket nem akartam a gamebe rakni, mert akkor az tele lett volna függvényekkel. A Bulletekhez is létrehoztam egy Bullets osztályt, ami a lövedékeket tárolja típustól függetlenül. A playerhez és az invaderekhez létrehoztam listenereket, amik szólnak a rájuk „feliratkozott” osztálynak, ha lőttek (ez a Game osztály csak), ezek az osztályok pedig hozzáadnak egy adott lövedéket a lövedékeikhez. Ugyan így valósítottam meg az ablakok közti mozgást is, csak egy ChangeWindow listenerrel.

A játékok elmentéséhez minden Game által használt osztályt szerializálhatóvá tettem, és létrehoztam egy osztályt, ami egy game osztályt ment el egy game.ser fájlba. A visszatöltést úgy oldottam meg, hogy létrehoztam egy setGame(Game) függvényt, ami olyan mint az összes setter függvény egyben. Minden adatát a játéknak beállítja a paraméterként kapott értékre.

A játékos mozgását úgy oldottam meg, hogy egy függvény figyeli, hogy milyen gombok lettek lenyomva/felemelve, és ettől függően állítja be a játékos dir attribútumát (ami egy enum left, right és stand értékeit veheti fel).

Az invaderek lövéséhez az AnimationTimer után létrehoztam egy Timeline-t, aminek feladata pár másodpercenként végigmenni az invadereken, amiken kis valószínűséggel meghívni a shoot() függvényüket.

A program use-case diagramja:

