

GUIYogiBearAssignment

- Tárgy: Programozási technológia GY 2025/26/1
- Szerző: Boromissza András | HYCZBO

Feladatléírás

- A megvalósításnak felhasználóbarátnak és könnyen kezelhetőnek kell lennie. Törekedni kell az objektumorientált megoldásra, de nem kötelező a többretegű architektúra alkalmazása.
- A megjelenítéshez első sorban elemi grafikát kell használni. Az így kirajzolt 'sprite'-ok közül a játékoshoz tartozót billentyűzet segítségével lehessen mozgatni a jelenleg is standard (WASD) billentyűzet kiosztásnak megfelelően. Egyéb funkciókhoz egérhez kapcsolódó esemény vezérlőket is implementálhattok.
- Amennyiben nem algoritmussal generáltatod a játékteret, úgy legalább 10 előre definiált játékteret készíts különböző fájlokban eltárolva. Ügyelj arra, hogy mind az algoritmussal generált játékok esetén, illetve az előre definiált esetekben is végig játszható legyen az adott terület.
- Minden feladathoz tartozik egy időzítő, mely a játék kezdetétől eltelt időt mutatja.
- A dokumentációnak tartalmaznia kell a választott feladat leírását, elemzését, a program szerkezetének leírását (UML osztálydiagrammal), egy implementációs fejezetet a kiválasztott játék szempontjából és/vagy az általad érdekesebbnek gondolt algoritmusok leírásával. (Például: pálya generáláshoz implementált algoritmus.), valamint az esemény eseménykezelő párosításokat és a tevékenység rövid leírását.
- A feladatléírás a minimális követelményeket tartalmazza. A játékok tetszőlegesen bővíthetők.

Maci Laci (Yogi Bear)

A meséből jól ismert Maci Laci bőrébe bújva a Yellowstone Nemzeti Park megmászhatatlan hegyei és fái között szeretnénk begyűjteni az összes rendelkezésre álló piknik kosarat. Az átjárhatatlan akadályok mellett Yogi élelem szerzését vadőrök nehezítik, akik vízszintesen vagy függőlegesen járőröznek a parkban. Amennyiben Yogi egy egység távolságon belül a vadőr látószögébe kerül, úgy elveszít egy élet pontot. (Az egység meghatározása rád van bízva, de legalább a Yogi sprite-od szélessége legyen.) Ha a 3 élet pontja még nem fogyott el, úgy a park bejáratához kerül, ahonnan indult.

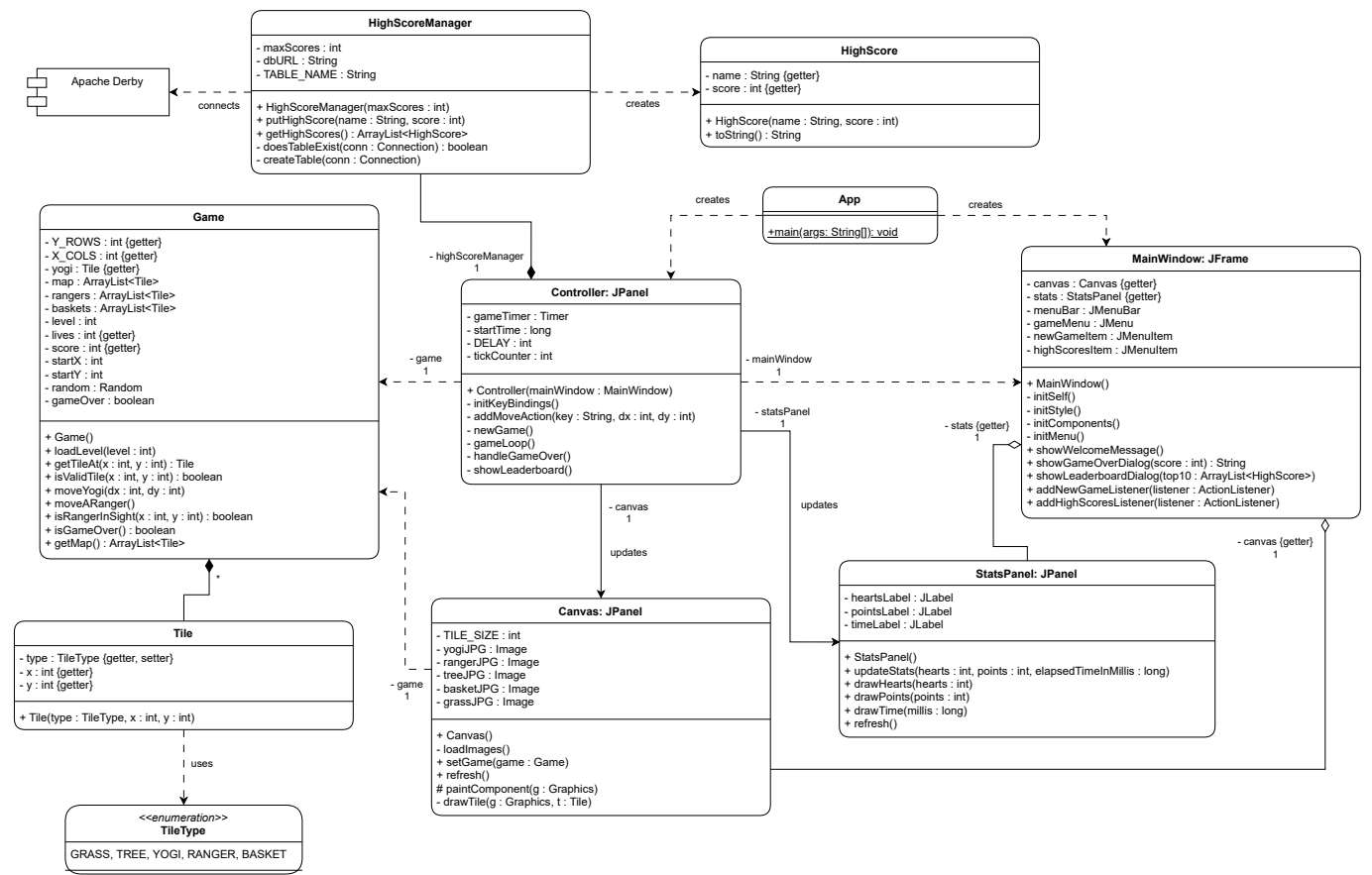
A kalandozás során, számon tartjuk, hogy hány piknik kosarat sikerült összegyűjtenie Lacinak. Amennyiben egy pályán sikerül összegyűjteni az összes kosarat, úgy töltünk be, vagy generálunk egy új játékteret. Abban az esetben, ha elveszítjük a 3 élet pontunkat, úgy jelenjen meg egy felugró ablak, melyben a nevüket megadva el tudják menteni az aktuális eredményüket az adatbázisba. Legyen egy menüpont, ahol a 10 legjobb eredménnyel rendelkező játékost lehet megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből.

Implementáció

Törekedtem a MVC architektúra megvalósítására, a három package ennek megfelelően: model, view, controller. Az App osztály main metódusa példányosítja a Controllert és a MainWindow-t. A Controller példányosítja a játék logikát tartalmazó Game objektumot és a perzisztenciáért felelős HighScoreManager objektumot, továbbá az eseménykezelés is itt zajlik, amivel összeköti a view és a model réteget. A pálya csempéi és az entitások egyetlen Tile listában alkotják a játék aktuális állapotát. Ennek megfelelően minden

változás az érintett Tile objektumok típusának változásával és az entitások referenciáinak frissítésével jár együtt. A view réteg Canvas és StatsPanel objektumai a Controlleren keresztül a gameLoop metódusban kapják meg a Game osztálytól a játék aktuális állapotát, amit minden iterációban kirajzolnak. Az adatperzisztencia beágyazott Apache Derby adatbázissal valósul meg.

Osztálydiagram



Eseménykezelés

esemény	eseménykezelő	leírás
"W"	moveW	Yogi felfele lép, ha tud
"A"	moveA	Yogi balra lép, ha tud
"S"	moveS	Yogi lefele lép, ha tud
"D"	moveD	Yogi jobbra lép, ha tud
"SPACE"	PauseGame	Játék megállítása vagy elindítása
NewGame menügomb	newGame()	Új játék indítása
Leaderboard menügomb	showLeaderboard()	TOP 10 ranglista megnyitása