UltraFighters

https://github.com/PajaS22/UltraFighters

Pavel Svoboda - maturitní program

Stručný popis

Jedná se o 2D desktopovou hru "plošinovku" pro více hráčů. Cílem každého z nich je zůstat jako poslední živý, ale hru lze hrát i na týmy. Hráči ovládají bojovníky, kteří se umí různě pohybovat, bojovat na blízko i střílet.

Herní postavy

Ve hře mám v současné době 3 různé postavy: Billy, Jeff a Igor. Každý má svůj vlastní design, ale co se týče schopností se neliší.

Bojovníci mají nastavitelný počer životů, který je zobrazován v horní části obrazovky formou červených ukazatelů životů. Pokud počet životů klesne na nulu, bojovník umírá, na jeho pozici se místo něj objeví hrobeček a místo jeho ikonky u životů lebka.

počítadla životů

Schopnosti

Každý hráč má na výběr z velké škály pohybů, které jím ovládaný bojovník umí provést. Hráč bojovníka ovládá přes klávesnici a klávesy jsou nastavitelné. Primární nastavení prvního hráče je takovéto:

Klávesy	Akce
Vpravo/vlevo	Běh
Nahoru	Skok
Dolů	Skrčení
2x vpravo / vlevo	Sprint
Dolů + vpravo / vlevo	Parakotoul
M	Rána pěstí / kopanec (pokud zrovna ve skoku)
N	Střela z pistol
,	Výstřel z brokovnice
N / , + nahoru / dolů	Míření

Druhý hráč se pohybuje primárně klávesami WASD a útočí RTZ

Třetí hráč se pohybuje primárně klávesami na Numerické klávvesnici 5369 a útočí 0 , Enter

Nastavení ovládacích kláves je libovolné a počet hráču je omezen pouze tím, kolik se jich vejde k jedné klávesnici.

Herní prostředí

Hra se odehrává, jak jsem již zmínil ve 2D mapě. Současně mám připravené 3 různé mapy: měso, les a skály. Každá mapa má jiné pozadí a jinak rozmístěné plošiny. Na každé mapě působí gravitace.

Ve hře jsou 2 druhy objektů: nepohyblivé a nezničitelné stěny a pohyblivé a zničitelné překázky. Oba typy objektů lze animovat sérií obrázků.

Pokud se bojovník, nebo cokoliv jiného, dostane mimo mapu, bude usmrcen.

Technologie

Hra je napsaná v jazyce C# s využitím frameworku .NET 4.8.

Grafiku jsem dělal pomocí Windows Prezentation Foundation - WPF.

Popis některých zajímavých algoritmů a principů programu

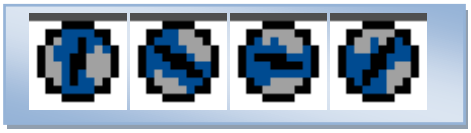
Chunky

Vzhledem k tomu, že jsem se chystal pohybovat opravdu hodně objekty najednou a řešit tak spousty kolizí, rozhodl jsme se rozdělit herní pole na chunky = oblasti. Každý chunk má list ID objektů, které se v něm zrovna nacházejí a kolize řeším pouze mezi objekty ve stejných chuncích. Samozřejmě, u pohybujících se objektů musím updatovat, v jakém chunku se právě nachází, ale to je mnohem méně náročné, než řešit kolize sposty střel z kulometu nebo střepin ze šrapnelového granátu se vším ostatním.

Input

Input mám formou klávesnice, kdykoli někdo zmáčně klávesu, spustí se funkce... to mi hlídá objekt. Důležité ale je to, že input běží asynchronně vůči hlavnímu vláknu, na kterém je GameLoop. Tak částečně zajišťuji plynulost animací.

Animace



Animace mi fungují tak, že mám ve složkách ke každému objektu několik obrázků, které vyměňuji na plátně. Pro každý objet si sólo nastavuji, jak dlouho chci který obrázek zabrazovat, tak mám totální kontrolu nad vzhelem mé hry. Hra má 66 FPS.

Architektura

Na jednom vlákně mi běží Timer, který každých 15 ms spustí tick() metodu objektu GameArea. V té se odehrává všechna magie. GameArea má referenci všech herních objektů a každému pošle informaci, at´ se pohne, upraví jeho zařazení v chuncích a

vyřeší kolize. Pokud byl v průběhu toho všeho objekt zničen nebo se dostal mimo herní pole, bude jeho reference v GameArea odebrána a objekt bude odebrán z Canvasu. Současně běží na třetím vlákně odposlech klávesnice a objekt k tomu určený dává vědět podle nastavení preferovaného ovládání patřičným objektům, že byla stisknuta ta a ta klávesa. Podle toho se potom může ovlivnit mohyb onoho objektu.