

PlatStreamer

Verseghy Coders

Kovács Marcell, Szepesi Zoltán

Felkészítő tanár: Tekse Marianna

Verseghy Ferenc Gimnázium, 5000 Szolnok, Tisza park 1.

1. Bevezetés

A videójátékok legfőbb célja az interakció, a játékosok ezáltal kerülnek kapcsolatba a programmal. Mi tettünk egy merész lépést, hogy ezt az interakciót egy magasabb szintre emeljük és már ne csak a játékos, hanem a nézők, vagy akár esportok esetén a szurkolók is bekapcsolódhassanak, így jött létre a PlatStreamer. A nézők interakcióba való bevonását, mi a mostanra nagyon népszerűvé vált streamekkel képzeltük el, de, ez csak egy felület, ami könnyedén lecserélhető. Mint ahogy a játék nevében található szójátékból kikövetkeztethetjük, a program alapjaiban egy platformer játék, mivel ezt a műfajt találtuk legalkalmasabbnak terveink megvalósítására, amire egy gyors (fast-paced) játékokat tudtunk teremteni.

A játék célja, hogy egy mini gamekkel teli platformert játszanak a felhasználók. Ez önmagában nem lenne meglepő, ám itt a játék menetét a Twitch chat segítségével befolyásolhatják a nézők is. Úgy gondoljuk, hogy ez a videójátékok és remélhetőleg az e-sportok jövője is, mivel a játékok interakciót biztosítanak nem csak a játékosokkal, hanem a nézőkkel is.

A nézők a játékmódot szavazhatják, bónuszokat adhatnak a játékosoknak (ezen utóbbi funkció még nem tökéletes, további fejlesztést igényel).

2. Probléma megoldásának menete

A négy játékost jelenleg egy számítógépről lehet irányítani (a hálózati irányítás még fejlesztés alatt áll). WASD, IJKL, nyilak és a numpaden található 8456 gombokkal tudunk mozogni. Lőni a Q, az U, a space és a numpad 7-es gombjával lehet (amennyiben ilyen a gamemode). Az első játékmód mindig a rifle elnevezésű játékmód utána pedig véletlenszerűen követik egymást, valamint 1-a 4-hez az esélye, hogy minden mód után szavazásra bocsátják a twitch chatben, hogy mi legyen a következő. Ha szeretnénk az F2 gomb segítségével képernyőképeket is készíthetünk a játék folyamán, ezek a screenshots/mappába automatikusan mentésre kerülnek.

Pályák egyszerűen létrehozhatók, a /resources/maps/ mappába rakhatunk pályákat vagy módosíthatjuk a jelenlegit, ezekhez saját textúrákat is rakhatunk a resources/textures/ mappába a pálya nevének megfelelő mappába rakva a textúrákat. A pályákat .map formátumban kell létrehozni (a

példa map1.map alapján, ezt használja most a program), a textúrákat, pedig .png formátumban.

A map fájlok először a játék verziószámát tartalmazzák, majd minden új sorban a platformok részeinek koordinátáit találhatjuk, ezt egy kettőskereszt (#) követi és az ez alatti sorok a platformok hitbox-ait tartalmazzák.

Jelenlegi Játékmódok:

- **Bomb Tag** (szavazásnál „bombtag”)

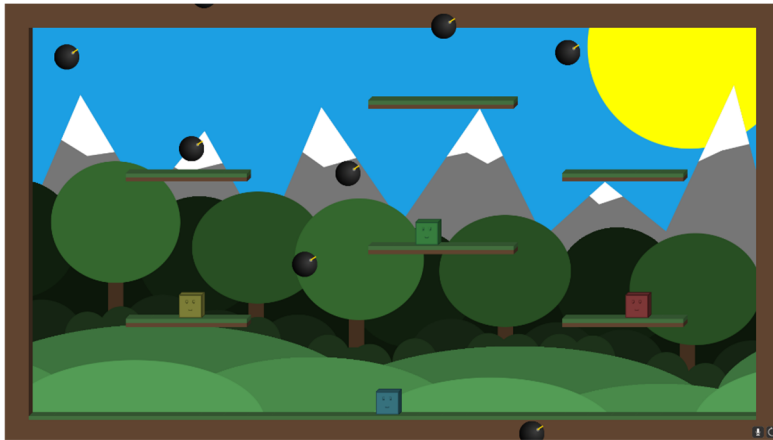
A Bomb Tag játékmód lényege, hogy a négy játékos egyike véletlenszerűen megkapja a bombát, ami egy 10 és 15 másodperc közötti intervallumon belül fel fog robbanni. Amennyiben hozzáér egy másik játékoshoz átadja annak a bombát és fél másodperc védelemet kap, hogy ne lehessen neki azonnal visszaadni. Amint a bombát tartó játékos felrobban, a játékos kiesik és újra véletlenszerűen valakihez kerül a bomba, ez addig ismétlődik amíg csak egy játékos marad a pályán.



1. ábra

- **Bomb Rain** (szavazásnál „bombrain”)

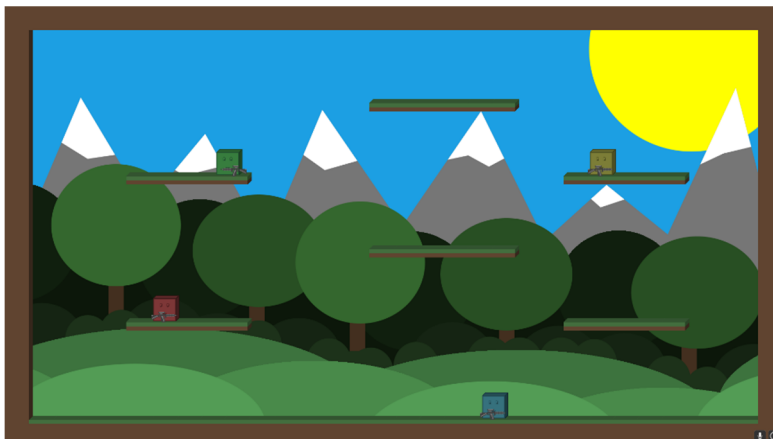
A Bomb Rain nevű játékmódban egy bomba „esőben” található magunkat, itt az a cél, hogy kikerüljük a zuhanó bombákat. Amennyiben valakit eltalál egy bomba, a bomba felrobban, a játékos pedig kiesik a körből. A játék addig tart, amíg csak egy játékos marad a pályán.



2. ábra

- **Rifle** (szavazásnál „rifle”)

A rifle játékmódban minden játékos rendelkezik egy mesterlövész puskával, ezeknek a puskáknak nagy előnye, hogy egyetlen egy lövéssel képesek bármelyik másik játékost megölni, de hihetetlenül sokáig - 5 másodpercig - tart újra tölteni. Ez a felállítás négy játékosal még gyorsabbá és élvezetesebbé teszi a játékot, de itt már komoly szerepe van a taktikának is. A játékmód addig tart még egyetlen egy játékos marad életben a pályán.



3. ábra

2.1. Technikai kivitelezés

A játék C++-ban íródott az SDL és annak alkönyvtáraival, az SDL_Net-el (TCP), curl-el (Twitch token ellenőrzés)

A fejlesztésben használtuk az egyik csapattag (Kovács Marcell) által írt C++ alapú játékmotort, amely dll-ként van csatolva, de nyílt forráskódú és elérhető a következő oldalon: <https://bitbucket.org/kovacsmarcell99/mengine-sdl>

A fizikai szimulációkhoz semmilyen könyvtárat nem használtunk, azok a program részei.

Igyekeztünk a C++11 és a C++14 szabvány adta lehetőségeket is kihasználni.

A programkód felépítésének és a saját játékmotornak köszönhetően a további játékmódok hozzáadása egyszerűvé vált a GameMode class egy child osztály létrehozása után az Update és Draw függvényeket kell implementálni.

A forráskódban egy twitchAPI mappába helyeztük el a Twitch integrációhoz írt kódjainkat, melyik szintén saját készítésűek.

A fejlesztés a Visual Studio 2015 segítségével történt. A modern projekteknél már megszokott eljárásokat is használtunk, mint például a source control (git). A forráskódot, így egy online git repository-ban tudtuk tárolni és bármilyen módosítást is végeztünk a kódon azt visszaállíthattuk a korábban már tesztelt és megírt forráskódokra, ami nagyban megkönnyítette a munkát.

A programozás nagy részét Kovács Marcell végezte, mivel ő már gyakorlottabb a C++ programozásban és otthonosan mozog ebben a témában, még a grafikák a pályák, valamint a Twitch-el való kommunikáció kódjának egy részét Szepesi Zoltán írta, mivel ő már rendelkezett ezzel kapcsolatban tapasztalatokkal (javascript-el már használta régebben).

A játékban látható összes grafika saját készítésű. A hangeffektek és a zene vagy Creative Commons Attribution 3.0 License alatt elérhetőek vagy public domain-ek, szükség esetén, a játékhoz csatolt fájlban megneveztük a forrást.

A programot, a saját grafikákat és a forráskódot szerzői jogok védik. A program üzleti célú felhasználása, továbbítása, vagy módosítása a készítők engedélye nélkül tilos.

3. Összegzés és további tervek

A játék alapjai készen vannak és nagyon biztosak, már csak néhány kisebb hibajavításra van szükség és néhány egyszerűbb funkció hozzáadására, hogy minden teljesen üzemeljen. A játékmódok bővítése viszonylag egyszerűen megvalósítható. Amire még szükség lehet az a nagyobb tartalom: több játékmód, pályák, több nézői interakció. Továbbá a hálózaton történő játék teljes implementálása.

Ezek közül elsődleges tervünk, amely egyben a célunk is a játékkal, a nézői interakció minél nagyobb kibővítése.