**PROJEKT TERV**

**Csoport tagjai**

|  |  |
| --- | --- |
| Grecmájer Máté |  |
| Musatics Gilbert |  |
| Szeszák Ádám |  |
| Teket Dávid |  |
| Tóth Előd |  |

**Fejlesztési terv: mérföldkövek**

* **H1 [ projekt terv ]** : csapat összeállítás, csapatnév meghatározás, projekt-ötlet kiválasztás, prezentáció készítés.
* **H2 [ követelmény analízis ]** : vízió dokumentum és projekt terv dokumentum elkészítése. Munkanapló összeállítása.

**#1 mérföldkő 16/10/11**

Készen vagyunk a

*player, map, model, quaternions, path, weapons, & visio*

dokumentumok kidolgozásával.

Sikerült leírni, hogy a szoftver egyes feladatkörök megvalósításaihoz milyen

problémákat kell megoldanunk. Egy egységes dokumentumot (visio) is megírtunk,

ami által egy teljesebb leírást adunk a projekt összes lényeges oldaláról.

A **játékos** (player) leírja a pajzs, életerő, páncél, sebesség, energia, lövés és

energia túlterhelés modelljeit és a velük járó működési részleteket.

A **modell** (model) leírja a .Obj formátum, fájl struktúra, felületek közötti kapcsolatok,

csoportok, leképezések, vertexek és ezek szintaktikájával foglalkozó definíciós

és fogalomlistáját.

A **pálya** (map) dokumentumai leírják a betölthető pályaformátumot, annak tulajdonságait, geometriai típusú követelményeket támaszt, megadja, hogy

milyen technológiát kell használni a megvalósításhoz.

A **kvaterniók**kal (quaternions) foglalkozó dokumentumok a kamera forgatásához szükséges algebrai problémák megoldására fókuszál, itt leírja, a vektorokkal és kvaterniókkal való számítási módszereket, 2D-s, 3D-s forgatásokat.

Az **útvonal** (path) leírja a 2D-s nézetbeli pontok létrehozásának, mozgatásának, törlésének, adatainak küldésének és módosításának módszerét.

A **fegyverek** (weapons) meghatározza a fegyverek típusát: elsődleges, speciális,

erős, oldalakon lévő. Leírja egyes fegyverek attribútumait, és a hozzájuk tartozó

viselkedések mintáit.

**Programozói nyelvek, fejlesztő eszközök**

A projekt megvalósítását C# programozási nyelv alatt, Visual Studio IDE fejlesztőkörnyezetben szeretnénk megvalósítani, a kódokat és egyéb dokumentumokat Githubon tároljuk, kezelésére SmartGit verziókövetőt használunk.

**Használni tervezett eszközök, technikák**

DirectX 11-et használunk, 3Ds max 2016 modellező szoftvert és Blender a modellek létrehozására,

opcionálisan Nodepad++ kézi adat módosításra,

Valve Hammer editor pálya geometria tervezésére és létrehozására mivel egy okos kis kód által könnyedén át tudjuk fordítani a vmf-et a számunkra megfelelő pálya formátummá.

**Létrejött és leendő produktumok és létrejöttük ideje**

|  |  |  |
| --- | --- | --- |
| Neve | Formátuma | Elkészülés (várható) időpontja |
| H1 prezentáció | ppt | 2016.09.20. |
| H2 prezentáció | ppt | 2016.10.11. |
| Vízió dokumentum | docx | 2016.10.11. |
| Projekt terv | docx | 2016.10.11. |

**Tesztelési követelmények**

Ahhoz, hogy a szoftvert megfelelő minőségben ki lehessen adni, fontos, hogy

minden szempontot kielégítő tesztelési és validálási lépések történjenek a

fejlesztés során. Tulajdonképpen egy fejlesztéssel párhuzamos tesztelési

ciklusról van itt szó, ami a következő részekből áll:

* Azonosítás: a helytelen programviselkedést kielemezzük és  
  bugként határozzuk meg.
* Jelentés: a bugokat rögzíteni kell *issue*-ként a felhasznált verziókövetőben, így a probléma napirenden és dokumentálva van.
* Javítás: azon fejlesztőnek kell kidolgoznia a módszert a hiba kijavítására aki az adott szoftverkomponenst kifejlesztette. Komplexebb hibák esetén a fejlesztői csapatnak kell megoldania a problémát.
* Jóváhagyás: a javítás tesztelése után meg kell győződni arról, hogy   
  többé nem áll fenn a jelzett hiba. Ennél a pontnál rendkívül fontos, hogy adott hiba kijavítása ne okozzon máshol jelentkező hibát. Több teszt esetet is le kell futtatni, majd a folyamat végén röviden dokumentálni kell a hibát.

A bug három szinten lehet:

* Lokális: modulszinten / komponensszinten
* Távoli: X modul funkcionálisan szabotálja (bug-ot okoz)   
   Y modulban
* Rejtett: létezik, de nem tudunk róla

Fontos, hogy a fejlesztés során a kódbázis módosítása kihathat az előző stabil

állapotára. Egy *feature* kifejlesztése és működése nem feltétlenül jelenti azt, hogy   
nem fog kárt okozni más *feature-*ökben.

A fejlesztés során a fent felsorolt pontok alapján fogjuk koordinálni a teszteket.

**Bemutatás (demó) és szállítás módja, ideje**

**Információ források**

.Obj wavefront fájl olvasása és használata http://www.martinreddy.net/gfx/3d/OBJ.spec

.map fájl alapján készült a hozzá tartozó dokumentum egy Amerikai Egyetem hallgatója készítette el. Az eredeti forrás megsemmisült de a PDF ami hozzá tartozik megmaradt és a személyes dropbox fiókomban le lett tárolva.

https://dl.dropboxusercontent.com/u/86461047/MAPFiles.pdf