Projekt Terv

# Gyakorlatvezető:

# Dr. Barabás Péter

# Csoport tagjai

|  |  |  |  |
| --- | --- | --- | --- |
| Grecmájer Máté | KZIGX3 | gremate96@gmail.com | -Programozás  -játékos logika tervezése  -dokumentálás  -prezentáció |
| Musatics Gilbert | JW28VF | gilbert12@citromail.hu | -Programozás  -algebrai problémák megoldása (kamera elkészítése)  -prezentáció  -dokumentálás |
| Szeszák Ádám | AZCTJJ | szeszak.adam@gmail.com | -Programozás  -Model formátum tanulmányozása; olvasása és használata.  -dokumentálás  -prezentáció |
| Teket Dávid | H2543D | zaug@outlook.com | -Programozás  -Látvány tervek  -Mesterséges inteligencia elkészítése  -dokumentálás |
| Tóth Előd | FOOJPQ | sonic7185@gmail.com | -Programozás  -Grafika, Pálya geometria elkészítése; kezelése; olvasása és írása  -Csapat írányítása  -prezentáció  -dokumentálás |

Tartalom

[Fejlesztési terv: mérföldkövek 3](#_Toc465167975)

[Programozói nyelvek, fejlesztő eszközök 4](#_Toc465167976)

[Használni tervezett eszközök, technikák 5](#_Toc465167977)

[Létrejött és leendő produktumok és létrejöttük ideje 6](#_Toc465167978)

[Tesztelési követelmények 7](#_Toc465167979)

[Bemutatás (demó) és szállítás módja, ideje 7](#_Toc465167980)

[Információ források 8](#_Toc465167981)

# Fejlesztési terv: mérföldkövek

* Szoftver követelmény specifikáció
* Analízis modell
* Rendszerterv
* Adatbázisok létrehozása
* Szoftver implementálása
* Szoftver belső tesztelése
* Béta verzió
* Visszajelzések értékelése, hibák kijavítása
* Végleges verzió

# 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. |
| H3 prezentáció | ppt | 2016.10.25. |
| Szótár 1.verzió | docx | 2016.10.25. |
| Munkanapló 1.verzió | docx | 2016.10.25. |
| SRS dokumentum 1.verzió | docx | 2016.10.25. |
| Felhasználói kézikönyv | docx | 2016.10.25. |
| H4 prezentáció | ppt | 2016.11.15. |
| Analízis modell | docx | 2016.11.15. |
| Rendszertervezés | docx | 2016.11.15. |
| Szótár 2.verzió | docx | 2016.11.15. |
| Munkanapló 2.verzió | docx | 2016.11.15. |
| SRS dokumentum 2.verzió | docx | 2016.11.15. |
| H5 prezentáció | ppt | 2016.11.29. |
| Tesztelési terv | docx | 2016.11.29. |
| Implementáció |  |  |

# 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.

# 

# 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