**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

[**Információs Rendszerek**](http://www.inf.elte.hu/karunkrol/szervezet/tanszekek/Lapok/Inform%C3%A1ci%C3%B3sRendszerekTansz%C3%A9k.aspx) **Tanszék**

**Portal 2D**

**Témavezető:**

Dr. Gombos Gergő

adjunktus, Ph.D.

**Szerző:**

Kiss János

Programtervező informatikus BSc.

Budapest, 2020

Tartalomjegyzék

[1. Bevezetés 3](#_Toc39628688)

[1.1 Rövid ismertető 3](#_Toc39628689)

[1.2 Motiváció 3](#_Toc39628690)

[2. Felhasználói dokumentáció 4](#_Toc39628691)

[2.1 A program telepítése és telepítési előfeltételei 4](#_Toc39628692)

[3. Fejlesztői dokumentáció 4](#_Toc39628693)

[Forrásjegyzet 6](#_Toc39628694)

# Bevezetés

## Rövid ismertető

Szakdolgozatom egy a Portal című játékhoz hasonló 2 dimenziós többszemélyes játék megvalósítása webes alkalmazásként. A játék fő eleme, hogy a pályák során különböző fejtörőket kell megoldani, teleportáció segítségével, portálokat lehet falakra helyezni. A játék fizikai rendszere szerint amilyen lendülettel áthalad egy tárgy az egyik portálon, az olyan lendülettel távozik a másikon, ez fontos szerepet fog játszik az egyes feladatok megoldásánál. A játék fizikai rendszerét magam valósítottam meg, ezzel egy egyedi fizikai szimulációt létrehozva. Az alkalmazást javascript nyelvben valósítottam meg, a többszemélyességet pedig TCP alapú web socketet alkalmazva nodejs szerverrel. Többszemélyesség a játékban oly módon nyilvánul meg, hogy együttesen próbálhatják megoldani a fejtörőket a játékosok.

## Motiváció

A 2007-ben megjelent Portal című játékkal még általános iskolában ismerkedtem meg és első pillanattól kezdve el voltam ámulva tőle. Egy számítógépes játékban valódi fizikát láttam méghozzá olyan sci-fi elemekkel, mint a teleportáció egybevetve. Abban a pillanatban tudtam, hogy aki egy ilyen dolgot meg tud valósítani az bármire képes, persze ekkor még nem tanultam programozni, így el sem tudtam képzelni, hogyan lehetne valami hasonlót megalkotni. Most hogy egyetemi tanulmányaim vége felé közeledem egy személyes próbatételként élem meg ezt a témát a többszemélyesség megalkotása pedig csak hab a tortán, hiszen soha nem dolgoztam még sem fizikával sem többszemélyes játékkal.

# Felhasználói dokumentáció

Ez a fejezet bemutatja a játék indításának előfeltételeit, rendszerkövetelményeit, az elindításához szükséges lépéseket és további instrukciókat, majd részletesen leírja a játék működését és használtatát.

## A játék telepítése és telepítési előfeltételei

A program egy webalkalmazás, így nincs szükség külön telepítésre, viszont szükségünk van a kliens oldalon egy modern böngészőre (pl.: google chrome,Microsoft Edge) valamint szerver oldalon Node.js keretrendszerre. A szerver és a kliens természetesen lehet azonos eszközön, viszont ha a két külön eszközt használunk akkor biztosítani kell a hálózaton való kommunikációt a két eszköz között ez megvalósítható *port-forwarding-*gal vagy LAN-os csatlakozással.

Az optimális játékélményért az alábbi rendszerkövetelmények ajánlottak:

* Op. rendszer: Windows 10
* Processzor: 2 Ghz
* Memória: 2 GB RAM
* Tárhely: 50 MB szabad hely

## A szerver

A jatek mukodesehez egz kituntetett szerverre van szukseg, amit futtathaatnuk a sajat gepunkon vagy akar egy tavoli gepen is az interneten

## Konfiguralasa

A szerver konfiguralasa a config.js fajlban tortenik, parameterei:

* server\_host: segitsegevel megadhato, hogy a szerver milyen ip cim alatt fusson, ha helyi szervert akarunk futtatni akkor ez nyugodtan lehet a ’localhost’ ertek
* server\_port: segitsegevel megadhato, hogy a szervert melyik porton szolgaljuk ki. Itt erdemes egy 1024-nel nagyobb szamot megadni, hisz ez alatt a rendszer szamara lefoglalt portok vannak.

## Elinditasa

A szerver elinditasahoz node.js keretrendszerre van szukseg, ami letoltheto innen: LINK-TO-NODE.

Ha mar rendelkezunk node.js-el a gepunkon akkor a szervert indithatjuk manualisan parancssorbol: a szerver fomappajaba navigalunk es beirjuk a kovetkezo parancsot:”node gameserver.js”, vagy windows 10 alatt hasznalhatjuk a local\_server.bat scriptet ami ugyanezt eredmenyezi.

Ha a szerver nem dobott hibat akkor keszen all a kliensek fogadasara

Lehetseges hibak:

* Nincs node : hibauzenet: xzxxzxz
* A port mar foglalt; hibauzenet : asdsadasd

## Kliems

Kliens kiszolgálása

Itt be kell mutatnod az alkalmazásod, olyan szinten, hogy a felhasználó használni tudja. Itt lehet berakni képernyőképeket, amelyek segítik a program használatát. Itt le kell írnod milyen gépigény szükséges, hogyan kell telepíteni / elindítani. Itt gondolkodj úgy, hogy ha te lennél a felhasználó, akkor tudnád-e használni ezek alapján a programodat. Itt lehet use case-eket használni, hogy bemutasd az egyes funkciókat.

# Fejlesztői dokumentáció

Itt egy másik szemszögből kell bemutatnod a programodat. Itt egy olyan fejlesztő vagy egy cégnél, akinek azt mondják, hogy fejlesszen hozzá ehhez a programhoz újabb funkciókat. Ekkor csak a fejlesztői dokumentáció áll rendelkezésedre, hogy megértsd, melyik modul mit csinál, és hogy hova kell nyúlnod, hogy meg tudd valósítani, amit kértek. Itt mutasd be a programod rétegeit, adatbázist, táblákat, osztályokat, modulokat, fontosabb függvényeket, algoritmusokat, felhasználói eseteket, hálózati kommunikációt, fejlesztői környezetet. Itt szerepeljenek osztálydiagramok, UML ábrák. Itt lehet beszélni az üzemeltetésről is. Milyen időzített folyamatok vannak, hol van a logolás, használ-e felhő rendszereket, stb. Ebbe a fejezetbe tartozik a **tesztelés** is. Minden osztályodra legyen tesztelés is. Egység tesztek, felületi tesztek, integrációs tesztek stb. Ha nem tudsz automatikus tesztet írni a programodra, akkor csinálj teszt jegyzőkönyvet. Ebben leírod mi a funkció, amit tesztelsz és annak mi az elvárt kimenete, de nagyon sok tesztelő eszköz elérhető manapság. Olyanok is vannak, amelyek felületet képesek tesztelni.

**Összefoglalás**: Nagyon hasznos tud lenni, ha egy oldalban össze tudod foglalni, hogy miről is szól a dolgozatod. Ez annyiban másabb, mint a bevezetés, hogy itt feltételezheted, hogy már elolvasták a dolgozatodat. Itt újra mutasd be a problémát, amivel foglalkoztál, mutasd be mit és hogyan használtál és végül mutasd be, hogy mire jutottál.

**További fejlesztési lehetőségek**: Egy program sose készül el, csak jön a határidő. Ebbe a fejezetbe írd le, milyen további funkciókat tudsz elképzelni az alkalmazásodhoz. Mit lehetne még hozzáfejleszteni, mi az, amit esetleg máshogy is meg lehetett volna valósítani.

# Forrásjegyzet

|  |  |
| --- | --- |
| [1] | „ELTE EDIT - IK Szakdolgozatok,” [Online]. Available: http://edit.elte.hu/xmlui/handle/10831/27. [Hozzáférés dátuma: 09 07 2019]. |
| [2] | G. Gergő, „Szakdolgozat Word sablon,” 2019. [Online]. Available: http://ggombos.web.elte.hu/szakdoli/szakdoli\_how\_to.docx. [Hozzáférés dátuma: 19 11 2019]. |
| [3] | C. Máté, „ELTE szakdolgozat template (Github),” [Online]. Available: https://github.com/mcserep/elteikthesis. [Hozzáférés dátuma: 10 07 2019]. |
| [4] | „BSc szakdolgozat követelmények,” [Online]. Available: https://www.inf.elte.hu/dstore/document/257/PTI\_BSc\_szakdoli\_uj.pdf. [Hozzáférés dátuma: 19 11 2019]. |
| [5] | „ELTE címer,” [Online]. Available: https://www.inf.elte.hu/dstore/document/203/cimerek.doc. [Hozzáférés dátuma: 19 11 2019]. |
| [6] | „Galaxis útikalauz stopposoknak - wikipédia,” [Online]. Available: https://hu.wikipedia.org/wiki/Galaxis\_%C3%BAtikalauz\_stopposoknak. [Hozzáférés dátuma: 09 07 2019]. |
| [7] | „Szakdolgozat leadáshoz szükséges dokumentumok,” [Online]. Available: https://www.inf.elte.hu/content/adatlapok-formanyomtatvanyok.t.1052?m=129. [Hozzáférés dátuma: 19 11 2019]. |
| [8] | „Murphy törvénye - wikipédia,” [Online]. Available: https://unciklopedia.org/wiki/Murphy\_t%C3%B6rv%C3%A9nyei. [Hozzáférés dátuma: 09 07 2019]. |
| [9] | „ BSc pontrendszer,” [Online]. Available: https://www.inf.elte.hu/dstore/document/1667/Bsc\_b%C3%ADr%C3%A1lat.docx. [Hozzáférés dátuma: 19 11 2019]. |