**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Média- és Oktatásinformatika Tanszék**

**Labirintus verseny a mesterséges intelligenciával**

**Témavezető:**

Bende Imre

tanársegéd, MA

**Szerző:**

Lukács Dávid István

Programtervező informatikus BSc.

Budapest, 2022

Tartalomjegyzék

[1. Bevezetés 3](#_Toc25048151)

[2. Témaválasztás 3](#_Toc25048152)

[3. Témabejelentő 3](#_Toc25048153)

[4. Konzultáció 3](#_Toc25048154)

[5. Program készítése 3](#_Toc25048155)

[6. Szakdolgozat felépítése 4](#_Toc25048156)

[Sablon 4](#_Toc25048157)

[Fejezetek 4](#_Toc25048158)

[Formai segítségek 5](#_Toc25048159)

[DVD melléklet 6](#_Toc25048160)

[Leadás 6](#_Toc25048161)

[7. Védés 7](#_Toc25048162)

[8. Értékelés 7](#_Toc25048163)

[9. Tipikus hibák 7](#_Toc25048164)

[Forrásjegyzet 8](#_Toc25048165)

# Bevezetés

A szakdolgozatom témájának kiválasztása, ezáltal alapvető koncepciója az volt, hogy nem tartottam túl valóságos, gyakran előforduló esetnek azt, hogy labirintusba kerüljek valaha, de mindig úgy éreztem, hogy kicsit tartok, félek a megoldandó problémától, hogy nem lennék képes csak a józan eszemre és szerencsémre hivatkozva kijutni. Ez mellé jött az a tulajdonságom, hogy sokkal könnyebben szokott egy új dolog megtanulása, elsajátítása menni akkor, ha valaki azt a cselekvést éppen előttem végzi, és mindig érdeklődéssel töltött el az, ahogy próbáltam rájönni a cselekvés közben hozott döntések miértjére. Ezt a gondolatot és fejlődési vágyat ragadtam meg. Tehát a programom szeretné úgy szórakoztatni a felhasználóját, hogy gondolkodásra, gyorsaságra, döntéshozatalra készteti, illetve ad is neki egy példát, akitől akár tanulhat is, hiszen kijutása során egy ellenfélt állít vele szembe, aki közben mesterséges intelligenciát, gráfkereső és labirintus generáló algoritmusokat használ ellene. Azonban ezt még kevésnek tartottam, szerettem volna azt, hogyha a felhasználók egymással is interakcióba léphetnek valamilyen módon, de azt éreztem, hogy túl sok terhelést jelentene minden éppen online felhasználót megjeleníteni a labirintusban. A felhasználói élmény, és a szociális jelleg növelése érdekében egy közös adatbázisban láttam a megoldást, ahova a felhasználók kedvük szerint készíthetnek labirintusokat, ezáltal adva egy végtelenségig szórakoztató, el nem laposodó jelleget a játéknak, hisz bármely pillanatban bővülhet az adatbázis néhány új labirintussal, amivel aztán a többi felhasználó, vagy akár saját maga is küzdhet. A szociális jelleg egyfajta kompetitív tulajdonságot is hoz magával, ugyanis a játék bár elméletileg a végtelenségig mehet, a közben érkező folyamatos nehezítések, amely leginkább a mesterséges intelligencia fokozatos gyorsítását jelenti, egy idő után lehetetlenné teszik a túl jó pontszám elérését. Kicsit elvesz a kompetitív jellegből viszont a sok véletlenszerűség, hogy véletlenszerű pályát kapunk, ami abból kiindulva, hogy a labirintus kezdő és végpontja között nincs meghatározva sem minimális, sem átlagos távolság, végül változó nehézséget jelent a kijutást illetően. A nyelvek, környezetek választásánál az volt a szempontom, hogy az itteni tanulmányaim során megismert technológiákat, algoritmusokat vegyítsem olyan újdonságokkal, amik a témából kifolyólag érdekelnek engem.

# Felhasználói dokumentáció

Az alapvető probléma, hogy nem volt még szerencsém olyan játékszoftverhez, amelyben volt lehetőség labirintusok létrehozására, illetve a kijutáson kívül valamilyen extra nehézség is került a játékmenetbe. Jelen esetben, a program futása során egyfajta mesterséges intelligenciát személyesít meg, ami az egyik hatékonyabb labirintus útkereső algoritmust használja, ezzel próbálkozik az emberi versenytársa szerencséje, gyorsasága és tudása ellen. A probléma másik fele , hogy bár mindenképp szerettem volna a játékot még ennél is izgalmasabbá tenni azzal, hogy a felhasználókat az egymással történő versengésre, akadályoztatásra buzdítom, nem szerettem volna kiszúrni azon játékosokkal, akik infrastruktúrájuk, vagy pillanatnyi helyzetük miatt nem rendelkeznek aktív internetkapcsolattal. Így a játék rendelkezik egy offline móddal is, ami az eddigi tapasztalataim miatt kicsit nehezebbre is sikerült, mint az online mód, tekintettel arra, hogy az online módban a játékosok hajlamosak egyszerűbb labirintust csinálni, mint egy, a bonyolultságra összpontosító algoritmus.

A program futtatásához egy Windows 10 rendszerű, 64-bites processzoros számítógép szükséges. A pontos, minimum követelmények:

* Processzor: Legalább 1 gigahertz (GHz) órajelű processzor.
* RAM: 2 gigabyte (GB).
* Tárhely: Az operációs rendszeren, és a futtatókörnyezeten felül 2 GB szabad tárhely szükséges.
* Grafikus kártya: DirectX 9 kompatibilis grafikus kártya, legalább WDDM 1.0 illesztőprogrammal.
* Kijelző: Legalább 1280 x 720 pixel felbontású
* Operációs rendszer: A játék Windows és Ubuntu alatt is tud futni, a megfelelő játékélményhez a Windows 10/11 operációs rendszer ajánlott.
* Szoftver: A szoftver futtatásához egy legalább Java 17 Runtime verziót futtatni képes Java verzió szükséges.

A játékhoz szükséges java verzió (Windows és linux esetén) a következő weboldalról érhető el: <https://www.oracle.com/java/technologies/downloads/#jdk18-windows> . Windows esetén a weblapot megnyitva a Windows fülön az „X64 Installer” sorban szereplő linken lehet letölteni a telepítő fájlt.  
A képen szöveg látható

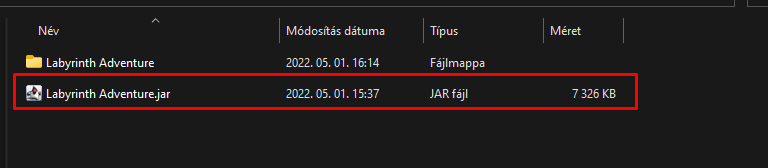
Automatikusan generált leírás

1. ábra : a letöltési link lokációja a weblapon.

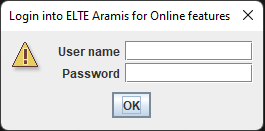
Példaképpen, a Java 18 telepítője 420 megabyte (MB) helyet igényel a merevlemezünkön. A telepítő megnyitásánál, ha szeretnénk eltérni az alapvető telepítési helytől, válasszuk ki a számunkra szimpatikus útvonalat, majd a tovább gombbal fel is települ a futtatási környezet. A képen szöveg látható

Automatikusan generált leírás

2. ábra : a Java futtatási környezet telepítő ablaka.

A Java futtatási környezet feltelepülése után már csak a játékszoftver futtatása maradt hátra. Ezt a csatolmányban mellékelt tömörített állomány kicsomagolásával, majd a kicsomagolt verzióban, a főkönyvtárban elhelyezett „Labyrinth Adventure.jar” fájl megnyitásával érhetjük el.  


3. ábra: Futtatható állomány a kicsomagolt mellékletből

A futtatás során (több képernyős rendszerek esetén az operációs rendszerünkben megadott alapértelmezett képernyőn) megnyílik egy kis ablak: 

4. ábra: Bejelentkező képernyő a program megnyitásakor

Ebben az ablakban szükséges az ELTE Aramis adatbázisához rendelkezésünkre álló felhasználónév-jelszó párt beírnunk, amivel később elérjük saját labirintusainkat, vagy megtekinthetjük, esetleg megdönthetjük rekordunkat. Amennyiben ilyennel nem rendelkezünk, a következő felhasználók valamelyikével szintén be tudunk lépni:

* Adminisztrátori felhasználó:
  + felhasználónév: IDU27K
  + jelszó: almafa
* Felhasználói jogokkal rendelkező felhasználók:
  + felhasználónév: HF3LBG
  + jelszó: HF3LBG
  + felhasználónév: RNYR2F
  + jelszó: RNYR2F
  + felhasználónév: GF4SDH
  + jelszó: GF4SDH

Ha nem rendelkezünk, vagy nem szeretnénk a fenti felhasználók valamelyikét használni, a bemeneti mezőket hagyhatjuk üresen is, ezzel viszont csak az offline funkciók maradnak használhatóak. Ha ezt a döntést meghoztuk, az „OK” feliratú gombra kattintva a főmenübe jutunk.  
A képen szöveg, áramkör, elektronika látható

Automatikusan generált leírás

5. ábra: A főmenü kinézete, ha nem elérhetőek az online funkciók

Sikertelen bejelentkezés, vagy szándékosan üresen hagyott bejelentkezési adatok esetén azt tapasztalhatjuk, hogy csak a „Play Offline” és „Exit Game” feliratú gombok elérhetőek. Amennyiben ez nem az elvárt eredmény, ellenőrizzük internetkapcsolatunkat és a program újra futtatásával, majd gondosan ellenőrzött bejelentkezési adatok megadásával próbálkozzunk ismét. Ha viszont a bejelentkezés sikeres, a következő főmenü jelenik meg:  
  
A képen szöveg látható

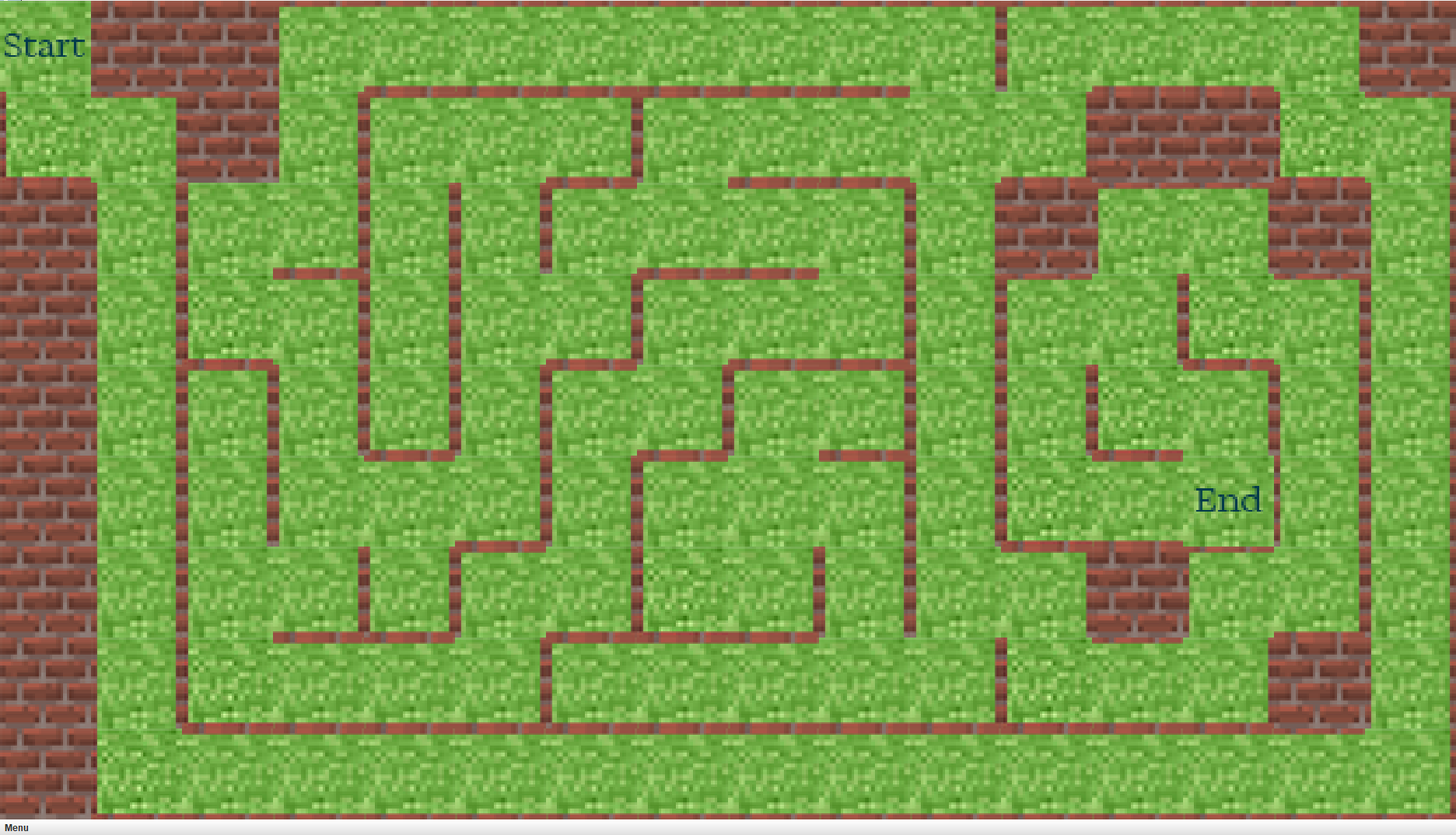
Automatikusan generált leírás

6. ábra: A főmenü kinézete, sikeres bejelentkezés, ezáltal elérhető online funkciók mellett

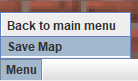
A főmenüből az elérhető menüpontok, és funkciójuk balról jobbra:  
„Map Builder” (csak online elérhető) : Ezen menüpont alatt van lehetőségünk a pályakészítő funckiót elérni. A képen szöveg, kültéri, tégla, építőanyag látható

Automatikusan generált leírás

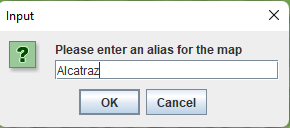
7. ábra: A pályakészítő funkció kinézete a menüben lévő gombra kattintás után

A pályakészítőben egy labirintust úgy tudunk megrajzolni, először az egérmutatót az általunk kiválasztott kezdőcellára helyezzük, majd a bal egérgombot lenyomva, és a kurzort valamilyen irányba húzva elkezdjük kialakítani a labirintusunkat. A labirintusunk kezdőhelye egy „Start” felirattal lesz jelölve, a labirintusból kivezető utat pedig teljesen a saját elképzelésünkre bízhatjuk. Arra viszont figyelnünk kell, hogy ha a labirintus kezdőpontjából a végpontjába nem vezet bejárható út (például úgy, hogy a labirintusunk végpontja és kezdőpontja közti részt minden esetben legalább egy fal elválasztja), akkor egy későbbi pontban az elmentést, az adatbázisba feltöltést a program nem fogja engedni. Egy példa egy elkészített labirintusról:  


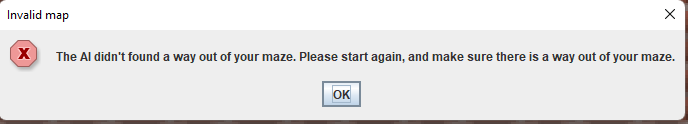
8. ábra: Egy mentésre kész labirintus

Ha elégedettek vagyunk az elkészített labirintussal, a képernyő bal alsó sarkában látható „Menu” feliratú gombra kattintva, majd a „Save Map” feliratú gombra kattintva a játék arra kér minket, adjunk egy becenevet a labirintusunknak, amiről később majd felismerjük. Természetesen, ha mégsem szeretnénk elmenteni a labirintusunkat, a „Back to main menu” gombot megnyomva ismét a főmenüben találjuk magunkat.  


9. ábra: Képernyő bal alsó sarkában található "Menu" gomb, amiből aztán a "Save Map" menüpont lesz elérhető.



10. ábra: Becenév adása a labirintusnak

Ha az OK gombra kattintottunk, a szoftver a háttérben ellenőrzi, hogy ténylegesen van-e kiút a labirintusból, amit készítettünk, és ha talált, azt úgy jelzi, hogy ki lép az aktuális ablakból. Ha viszont olyan labirintust szeretnénk elmenteni, ami nem rendelkezik kiúttal, a következő hibaüzenet fogad minket:   
  


11. ábra: Hibaüzenet, ha olyan labirintust készítenénk, amiből nincs kiút

Ha nem kaptunk hibaüzenetet, és semmilyen más internetkapcsolati probléma nem avatkozott közbe, a „My Maps” menüpont alatt megtekinthető az új labirintusunk, de ezt a menüpontot még részletesen bemutatom.

//////////vége

# Témabejelentő

A témabejelentőbe 0.5-1 oldalban kell összefoglalnod, hogy mit is fogsz csinálni. Itt pár mondatban mutasd be a probléma területet, majd írd le, hogy mit fogsz te megcsinálni és hogyan. Meg lehet adni milyen nyelveket, eszközöket használsz. Célszerű nem teljesen konkrét leírást írni, mert ha menetközben kiderül, hogy az adott nyelv nem támogatja azt, amit szeretnél, vagy az csak újabb verziókban elérhető, akkor már nem tudsz módosítani. Például, érdemesebb azt írni, hogy mobil alkalmazás, mint Android vagy iOS program. A témádat egyeztesd el a témavezetőddel. Ha sikerült elkészíteni a leírást, akkor azt neptunban egy kérvény formájában kell beadnod, amit majd a témavezetőd elfogad.

# Konzultáció

Konzultációra járni most már kötelező, de érdemes is. Viszont akkor érdemes konzultálni, ha van miről beszélni. Úgy érkezz a konzultációra, hogy már vannak terved a dolgozattal kapcsolatban, már van működő alkalmazás, amit meg lehet beszélni, vagy épp kérdések merültek fel, amiket szeretnél megbeszélni. A konzultáció nem ér véget a dolgozat leadásával. A védésre ugyanúgy fel kell készülni. Érdemes a védésedet megmutatni a témavezetőnek. Ő még el tudja mondani, mit kellene még javítani. Érdemes az egész dolgozatnak nekiállni, hogy csinálsz egy ütemtervet, hogy milyen határidőre meddig kell, hogy elkészülj a programoddal, hogy le tudd adni. Ha ezt betartod, akkor nem lesz gond.

# Program készítése

A program elkészítése tűnik a legkönnyebb feladatnak, de nem az. Ha egy jó dolgozatot szeretnél, akkor igényesen, jól megtervezett, átgondolt alkalmazást kell készítened. A készítésnél sokan nem mernek olyan nyelvekhez, eszközökhöz nyúlni, amit nem ismernek. Mai világban inkább vannak mobil, vagy webes alkalmazások, mint asztali alkalmazás. Nem kell megijedni ezektől. Programozó vagy. Egy programozó bármilyen nyelven tud programozni. Csak rá kell szánni az időt, hogy megismerd a nyelvet. A programodban legyen help, ami segíti a felhasználót az alkalmazásod használatában.

# Szakdolgozat felépítése

A szokásos kérdés hogy épüljön fel a szakdolgozat. Most leírom, hogy mi az, ami célszerű egy szakdolgozatban.

Sablon

Kezdjük azzal, hogy milyen sablont használj a dolgozat elkészítéséhez. Ha Word-ben írod a dolgozatodat, akkor használhatod ezt a fájlt [2], amit most olvasol. Persze olvasd el a formai követelményeket, hogy minden teljesüljön (1,5-es sorköz), illetve a megfelelő részeket a borítón írd át. A Latex-ben írnád dolgozatodat, akkor használhatod Cserép Máté sablonját [3].

Fejezetek

A szakdolgozat alap fejezetei: a bevezetés, felhasználói dokumentáció és a fejlesztői dokumentáció. Ezen kívül célszerű még írni, egy összefoglalást és egy további tervek részt.

Témabejelentő: A témabejelentő kérvényt csatolni kell a dolgozat mellé. Ezt a témabejelentőt a neptunból kell letölteni és a feltöltendő „.zip” csomagba betenni!

Tartalomjegyzék: A dolgozat elején legyen egy tartalomjegyzék, ahol a főbb fejezetek látszódnak. Célszerű kerülni a túl „mély” fejezet szinteket, pl: 1.2.4.2.5.3-as fejezet. Ehhez persze mindenképp szükséges, hogy oldalszámozva legyen a dolgozat.

Köszönetnyilvánítás: Ha esetleg pályázati támogatást kaptál a dolgozatodhoz, akkor azt se felejtsd el megköszönni.

Bevezetés: Itt le kell írni, miről is fog szólni a szakdolgozatot, leírod mi a motiváció. Itt kell felkelteni az olvasó érdeklődését, hogy el akarja olvasni a dolgozatodat. Itt nem kell belemenni a részletekbe, itt lehet felületesen fogalmazni, hogy egy egyszerű halandó is megértse, miről szól a dolgozat. Itt veted fel a problémát, amivel elkezdtél foglalkozni. Bemutatod, hogyan akarod ezt megoldani. Lényegében egy előzetes áttekintést adsz a dolgozatról.

Felhasználói dokumentáció: 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.

Formai segítségek

A hivatalos formai követelményeket ellenőrizd le a kari honlapon [4], de a fontosabbakat összegyűjtöttem neked. Az elte hivatalos címerét pedig innen tudod letölteni [5] .

Követelmények:

A szakdolgozatot kemény kötésben kell leadni, 1 példányban.

Fedőlapjának színe fekete, aranyszínű feliratokkal.

Lap: A4-es méret, színe fehér

Betűméret: 12 pont

Sorok: sorkizárt igazítás, 1,5-es sortávolság

Margó:

belső: 3,5 cm

külső: 2,5 cm

alsó: 2,5 cm

felső: 2,5 cm

Oldalszám: az oldalszámozást a tartalomjegyzéktől kezdve az irodalomjegyzékkel bezárólag folyamatosan kell végezni. Jelölése arab számokkal történik (a tartalomjegyzék oldalainak az oldalszámát nem szokás feltüntetni).

A dolgozat fő fejezetei (1. szintű címsorok) új oldalon kezdődjenek.

Képek: Ha képet raksz a dolgozatodba, akkor annak mindig legyen képaláírása és sorszáma. Ha raksz be képet, akkor a szövegben hivatkozz is rá. Az 1. ábrán láthatjuk például a Galaxis útikalauz stopposoknak című könyv bortóját. A könyvből elhíresült mondatot („Ne ess pánikba!”) Elon Musk is felhasználta, amikor egy Teslát lőtt ki az űrbe. Ezt láthatjuk a 2. ábrán. Másik oldalról a képaláírás olyan legyen, hogy meg tudjuk állapítani, mi van a képen és ne kelljen a szövegben keresgélni.

|  |  |
| --- | --- |
| https://upload.wikimedia.org/wikipedia/hu/thumb/3/32/Galaxis_%C3%BAtikalauz_gy%C5%B1jtem%C3%A9nyes.jpg/200px-Galaxis_%C3%BAtikalauz_gy%C5%B1jtem%C3%A9nyes.jpg  12. ábra Galaxis útikalauz stopposoknak könyv[[1]](#footnote-1) | https://upload.wikimedia.org/wikipedia/commons/1/17/Elon_Musk%27s_Tesla_Roadster_%2840110298232%29.jpg  13. ábra Elon Musk Teslája az űrben[[2]](#footnote-2) |
|  |  |

Hivatkozások: Ha felhasználsz bármilyen forrást akkor azt mindig hivatkoznod kell. Ez fontos, hogy semmilyen plágium ne merüljön fel. Erről egy papírt is ki kell töltened, hogy büntetőjogi felelősséged van ezzel kapcsolatban. Célszerű a szövegben is berakni egy hivatkozást, hogy kapcsolható legyen forrásjegyzékhez. Az előbbi képeket például a wikipédián találtam [6].

DVD melléklet

A DVD mellékletre mindent tegyél rá, hisz úgy is elfér. Legyen rajta a forráskód, legyen rajta futtatható formában az alkalmazásod, ha lehetséges. Legyen rajta a dolgozat .pdf formátumban, ill. a leadáshoz szükséges papírok is rákerülhetnek. Ha használtál külső csomagokat, adatokat, akkor azokat is tedd rá.

Leadás

Az ELTE IK-n megszünt a papír alapú szakdolgozat leadás! A szakdolgozatodat a neptunba kell feltölteni ”.zip” formátumba.

A becsomagolt állományban a következőknek kell hogy szerepeljenek [7]:

Témabejelentő, amit a neptunból töltesz le

Szakdolgozatod .pdf formátumban

A programod forráskódja

EDIT Hallgatói dolgozat űrlap

EDIT - Elhelyezési megállapodás - Hallgatói dolgozat

Szakdolgozat/Diplomamunka leadási és eredetiség nyilatkozat

EDIT - Hallgatói dolgozat titkosítása (ha szükséges)

Védés

Az elkészült dolgozatodat meg is kell védened. A védésre 10-15 perced van. Ennyi idő alatt kell bemutatnod az alkalmazásod működését. Szakdolgozatnál nem prezentációt várunk el, hanem működő, jól kinéző, használható alkalmazást. Ha van az alkalmazásodban olyan rész, amelyet nem tudsz megmutatni működés közben, vagy épp a nagy eredmény a háttérben futó motor működése, akkor készíts róla prezentációt. Ekkor is célszerű minimális mennyiséget csinálni, max. 3-4 dia. A védésre célszerű egy jól végiggondolt, jól felépített és begyakorolt bemutatással jönni. Ha jó a védésed, az akár a dolgozatod jegyén is tud javítani. A védésre mindig legyen egy ’B’ terv. Az ördög nem alszik, és ami el tud romlani, az el is fog [8]. Éppen nem lesz internet a védés alatt, nem elérhető a külső forrás, amit használsz, stb. Csinálj egy videót, ahol működik a rendszer, vagy készíts diasort a funkciókról. Ha az alkalmazás védés során nem működik, vagy hibát dob, akkor sajnos nem fogják elfogadni a dolgozatot. A védésen a forráskódot is megnézzük, így az légyen megfelelően kommentezve. Védés előtt próbáld ki, hogy működik-e az alkalmazásod az egyetemi gépeken, mert ha nem akkor hozz laptopot és mutasd be azon. Ha laptopon szeretnéd bemutatni, akkor mindenképp jelezd a bizottság elnökének emailben.

Értékelés

A szakdolgozatra 3 jegyet fogsz kapni. Egyet a dolgozatra, egyet a védésedre, egyet pedig a húzott tételre. A három átlaga lesz a végső jegyed, emiatt is fontos, hogy a dolgozatod rendben legyen, és jól meg tudd védeni. A dolgozatodat a témavezetőd is értékelni fogja. Ennek az értékelésnek a pontrendszere elérhető a kari honlapon [9].

Tipikus hibák

Rossz a dolgozat felépítése. Ha több részből tevődik össze a dolgozatod, akkor előbb kezd kisebb elemek bemutatásával, majd az abból épülő nagyobbakkal, és így tovább. Ha lapozgatni kell a dolgozatban, mert olyat említesz, ami később lesz bevezetve, akkor az nem jó.

Sok kép, kevés szöveg, vagy fordítva. A dolgozatba lehet rakni képet, de minden képhez tartozzon szöveg is. Ha csak bemásolod a képernyőmentéseket és nem írsz róluk, akkor az olvasó nem biztos, hogy tudja értelmezni azokat. Ha viszont csak szöveget írsz és nem ábrázolod képekkel, akkor az megint nem segíti az olvasót.

A dolgozat nem éri el a 40 oldalt. Ha kevesebb oldalad van, akkor valami még kimaradt, vagy nem elég részletesen írtad le.

A témavezető nem látja a végső verziót. Mindig küld el neki mielőtt bekötteted a dolgozatot, hogy hátha észrevesz még hibát benne.

Védés közben derül ki valami hiba a programban. Ez a legrosszabb forgatókönyv. A védésre legyen egy begyakorolt bemutatód, hogy mikor mit fogsz bemutatni. Ezt gyakorold be nyugodtan otthon, nézd meg hogy beleférsz-e az időbe, és hogy minden működik. Olyat ne mutass, amiben nem vagy 100%-ig biztos!

Frissítve: 2022-04-08

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]. |

1. [https://hu.wikipedia.org/wiki/Galaxis\_%C3%BAtikalauz\_stopposoknak#/media/F%C3%A1jl:Galaxis\_%C3%BAtikalauz\_gy%C5%B1jtem%C3%A9nyes.jpg](https://hu.wikipedia.org/wiki/Galaxis_%C3%BAtikalauz_stopposoknak%23/media/F%C3%A1jl:Galaxis_%C3%BAtikalauz_gy%C5%B1jtem%C3%A9nyes.jpg) [↑](#footnote-ref-1)
2. <https://hu.wikipedia.org/wiki/Galaxis_%C3%BAtikalauz_stopposoknak#/media/F%C3%A1jl:Elon_Musk's_Tesla_Roadster_(40110298232).jpg> [↑](#footnote-ref-2)