Pannon Egyetem

Műszaki Informatikai Kar

[Tanszék neve]

[Szak neve]

**SZAK-/DIPLOMADOLGOZAT**

**[Dolgozat címe]**

**[Név]**

Témavezető: [Témavezető neve]

Külső/belső konzulens: [Konzulens neve]

[évszám]

Témakiírás

A szkennelt formában megkapott témakiírás beillesztése a dolgozatba.



Hallgatói nyilatkozat

Alulírott <<név>> hallgató (Neptun kód: <<SAJÁT NEPTUN KÓD>>) kijelentem, és a dolgozat feltöltésével egyidejűleg nyilatkozom, hogy a <<dolgozatcím>> című <<záródolgozatot/szakdolgozatot/diplomadolgozatot>> (a továbbiakban: dolgozat) a Pannon Egyetem Műszaki Informatikai Kar <<szervezeti egységében (tanszékén)>> készítettem a <<végzettség>> oklevél megszerzése érdekében.

Kijelentem, hogy a dolgozatban csak a megadott és hivatkozott forrásokat használtam fel, és ezekre a vonatkozó idézési szabályok szerint hivatkoztam.

Nyilatkozom, hogy a dolgozat érdemi része saját szellemi alkotásom eredménye, és azt más intézményben, szakon, vagy felsőfokú képesítés megszerzésére nem nyújtottam be. Tudomásul veszem, hogy a plágium vagy szerzői jogsértés esetén a dolgozatom elutasításra kerülhet, és ellenem fegyelmi eljárás indulhat. Tudomásul veszem továbbá, hogy szerzői jogsértés esetén az Egyetem jogosult a dolgozat elérhetőségét korlátozni, valamint eltávolítani a dokumentumot a dolgozatok tárolására szolgáló, a témát vezető szervezeti egység által meghatározott elektronikus zárt rendszerből.

Tudomásul veszem továbbá, hogy a Pannon Egyetem a dolgozat eredményeit saját céljaira eltérő írásbeli megállapodás hiányában a Pannon Egyetem Szellemi Tulajdon Kezelési Szabályzatában foglaltaknak megfelelően szabadon felhasználhatja.

Nyilatkozom, hogy a dolgozat elkészítése során mesterséges intelligencia eszközöket *használtam /nem használtam[[1]](#footnote-1).*

Nyilatkozom, hogy a dolgozat elkészítése során az alábbi táblázatban feltüntetett mesterséges intelligencia eszközöket kizárólag a kutatási, illetve fejlesztési feladat támogatására használtam fel, az érdemi munka, elemzés és következtetések teljes mértékben saját szellemi alkotásomat képezik.

Példa a táblázat kitöltésére:

|  |  |  |  |
| --- | --- | --- | --- |
| **Alkalmazott technológia** | **Alkalmazás módja** | **Előállított tartalom** | **MI használat aránya** |
| GPT-4o (OpenAI) | szöveges összefoglaló generálása | 2.2 fejezet | 80% |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Dátum: Veszprém/Nagykanizsa/Zalaegerszeg, 20.. <<hónap neve>> <<nap>>.

*<<hallgató neve>>*

Témavezetői nyilatkozat

Alulírott <<témavezető neve>> témavezető kijelentem, hogy a <<dolgozatcím>> című dolgozatot *<<*hallgató neve*>>* a Pannon Egyetem <<tanszék neve>>én készítette <<végzettség>> végzettség megszerzése érdekében.

Kijelentem, hogy a dolgozat védésre bocsátását engedélyezem.

Dátum: Veszprém/Nagykanizsa/Zalaegerszeg, 20.. <<hónap neve>> <<nap>>.

*<<témavezető neve>>*

Köszönetnyilvánítás

A hallgató köszönetet nyilvánít mindazoknak, akiktől (elméleti, gyakorlati, erkölcsi stb.) segítséget kapott.

Tartalmi összefoglaló

Tartalmi összefoglaló magyarul. Az összefoglalónak tartalmaznia kell (rövid, velős és összefüggő megfogalmazásban) a következőket:

* téma megnevezése,
* megoldott feladat megfogalmazása,
* megoldási mód,
* elért eredmények,
* kulcsszavak (4-6 darab)
* terjedelme nem lehet több 1 A4-es oldalnál.

Az összefoglalót magyar és angol nyelven kell készíteni. Sorrendben a dolgozat nyelvével megegyező kerül előrébb. A cím Title stílusú, formázása: Times New Roman, nagybetű, 14 pt, félkövér, középre igazított; az összefoglaló Normál stílusú, formázása: Times New Roman, 12 pt, sorkizárt, 1.5-ös sortávolság.

**Kulcsszavak:** [4-6 kulcsszó felsorolása, vesszővel elválasztva]

Abstract

Abstract in English

**Keywords:** [list 4-6 keywords]

Tartalomjegyzék

[Jelölésjegyzék 8](#_Toc99107193)

[1. Fejezet 9](#_Toc99107194)

[1.1. Alfejezet 9](#_Toc99107195)

[1.1.1. Alfejezet 9](#_Toc99107196)

[2. Új fejezet 10](#_Toc99107197)

[2.1. Új alfejezet 10](#_Toc99107198)

[3. Instrukciók 11](#_Toc99107199)

[Irodalomjegyzék 12](#_Toc99107200)

[Mellékletek 13](#_Toc99107201)

[Ábrajegyzék 14](#_Toc99107202)

[Táblázatjegyzék 15](#_Toc99107203)

Jelölésjegyzék

|  |  |
| --- | --- |
| AI: | Artificial Intelligence (Mesterséges Intelligencia) |
| GPU: | Graphical Processing Unit (Grafikus Processzor / Grafikus Feldolgozó Egység) |
| API: | Application Programming Interface (Alkalmazásprogramozási Felület) |
| CPU: | Central Processing Unit (Központi Feldolgozó Egység / Processzor) |
| GUI: | Graphical User Interface (Grafikus Felhasználói Felület) |
| HCI: | Human Computer Interaction (Ember-gép kapcsolat) |
| CIS: | Cognitive Information System (Kognitív információs rendszer) |

# Webots szimulációs környezet

A projekt megvalósítása a Webots szimulációs környezetben történik, az úthálózat megépítése előtt elengedhetetlen fontosságú annak megismerése.

A Webots egy nyílt forráskódú szimulátor, melyet kifejezetten robotikai kísérletekhez fejlesztettek, nagy hangsúlyt fektetve a valósághű fizikai modellezésre és részletes 3D-s környezetre. A program lehetőséget nyújt különböző robotok és járművek, illetve ezek szenzorjainak szimulációjára, valamint saját virtuális világ elkészítésére.

A környezet megismerése során különösen nagy figyelmet kapnak a következő területek:

* kezelőfelület
* Scene Tree hierarchia
* objektumok elhelyezése és azok paraméterezése
* pályaépítéséhez szükséges útobjektumok (Road, Crossroad, RoadLine)

A képen képernyőkép, szöveg, Multimédiás szoftver, Grafikai szoftver látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

1. ábra: ismerkedés a Webots környezettel

## Úthálózat megvalósításának módszerei

Az úthálózat megvalósítására három módszer került feltárásra. Ezek a következők:

* utak elhelyezése és szerkesztése manuálisan Webots környezetbe
* valódi városrészlet beimportálása OpenStreetMap segítségével
* úthálózat megrajzolása JOSM programban, majd importálása Webots környezetbe

A következő alfejezetekben mindhárom módszer bemutatásra kerül, lényegi leírásukkal, illetve a problémáikkal, amik miatt a módszer cserélve lett a soron következőre.

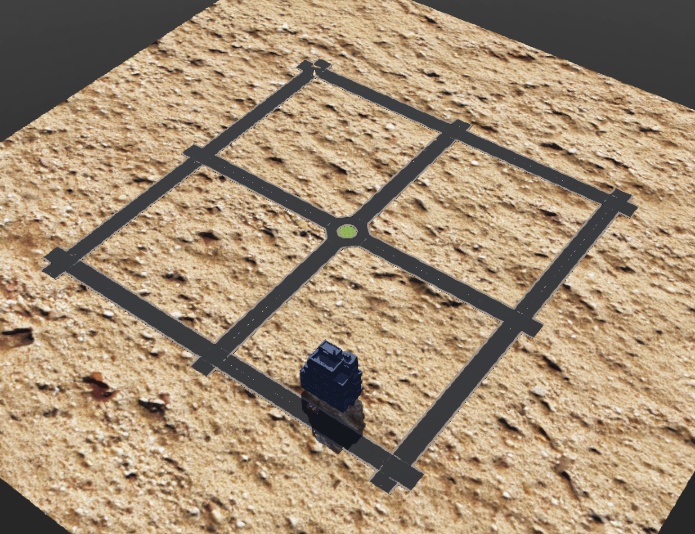
### Manuális útépítés

A legkezdetlegesebb megoldás a három módszer közül a manuális útépítés Webots környezetben. Ebben az esetben az úthálózat a „Road” és „Crossroad” típusú objektumok felhasználásával, valamint azok attribútumainak módosításával hozható létre. Az útépítés szempontjából legfontosabb paraméterek a „translation”, a „rotation” és a „wayPoints” attribútumok:

* translation: az útobjektum kezdő- vagy középpontjának elhelyezésére szolgál a világ globális koordinátarendszerében, egy koordinátapár megadásával
* rotation: az objektum elforgatását teszi lehetővé
* wayPoints: a „Road” típusú objektumoknál használható, azon koordináták listáját tartalmazza, amelyeken az útnak mindenképpen át kell haladnia

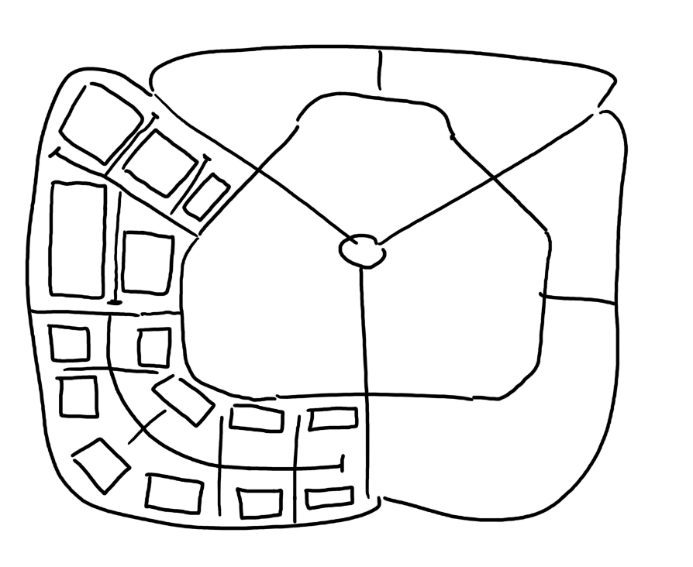
A probléma ott mutatkozik meg ebben a megoldás, hogy kimondottan időigényes

egy realisztikus és a feladat leírásának megfelelő szintű úthálózatot megalkotni, illetve amennyiben azt később bővíteni kellene, teljes utakat kell módosítani vagy legrosszabb esetben a nulláról létrehozni.



2. ábra: manuálisan összerakott úthálózat

A nehézségek közé tartozik még emellett az is, hogy a csapatban nem áll rendelkezésre olyan személy, aki megfelelő ismeretekkel rendelkezik közlekedési infrastruktúra tervezésében, ezáltal az ezzel a módszerrel létrehozott úthálózat realisztikussága korlátozott.



3. ábra: kezdeti tervrajz

### Valódi városrészlet beimportálása

A www.cyberbotics.com által nyújtott dokumentációk között található egy „OpenStreetMap Importer” nevű leírás, amely egy olyan szkritpről szól, aminek a segítségével a www.openstreetmap.org/export oldalon keresztül a világ bármely pontjáról letölthető annak virtuális mása. A letöltött részlet tartalmazza az adott területen található utakat, épületeket és egyéb objektumokat.

Az importáló szkript célja, hogy megszűntesse a szimulációs környezet manuális létrehozásával járó nehézségeket, illetve lehetővé teszi, hogy akár négyzetkilométeres területeken is valós időben lehessen szimulációt futtatni.

Az OpenStreetMap oldalról történő exportáláshoz meg kell adni a szélességi és hosszúsági fokok alsó és felső korlátait, ezt követően pedig a kijelölt terület osm (OpenStreetMap) fájlformátumban exportálható. Az így kapott fájlt ezután a Webots beépített szkriptje segítségével alakítható át egy olyan világgá, amelyet a szimulációs program képes megnyitni.



4. ábra: Budapest, IX. kerület a szimulátorban

A probléma ezzel a megközelítéssel az volt, hogy a létrehozott világ túlságosan realisztikus lett, különösen az úthálózat tekintetében. A robot működését figyelembe véve nincs szükség a többsávos utakra, sőt ezek akár az önvezető jármű hibás működését is okozhatják. Emellett előfordulhatnak olyan útszakaszok, amelyekről hiányzok a felfestések vagy színük nem megfelelő, így a robot számára az útvonal követése vagy a sávok felismerése nehézséget jelenthet. Ezen hibák kijavítása rendkívül időigényes lehet, különösen egy több négyzetkilométeres terület esetén.

### Rajz készítése JOSM-ban, majd importálás

A korábban említett dokumentációban az OpenStreetMap oldal mellett szerepel egy JOSM (Java OpenStreetMap Editor) nevű Java nyelven készült program is.

A JOSM célja, hogy az OpenStreetMap-ről exportált úthálózatok könnyen szerkeszthetők, bővíthetők legyenek, illetve lehetőséget biztosítson teljesen új úthálózatok létrehozására is.

A program használata a következő: a felhasználó egy kétdimenziós, grafikus felületen rajzolhatja meg az utakat és épületeket, hasonló módon, mint egy rajzolóprogramban.

Az így létrehozott objektumokhoz ezután különböző paraméterek rendelhetők, mint például az út típusa (főút, autópálya, mellékút), így az úthálózat gyorsan testre szabható.

A képen térkép, kereszteződés, csomópont, Városépítészet látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

5. ábra: Budapest, IX. kerület részlet, JOSM-ban megvalósítva

Az előző megoldásokban tapasztalt problémák ezzel a megközelítéssel nagyrészt megszűnnek: a program felületén egyszerűen hozzáadhatók vagy eltávolíthatók a hálózatból a nem kívánt útszakaszok, és a robot működéséhez szükséges feltételek is könnyedén biztosíthatók, anélkül, hogy a szimulációs környezet beállításait módosítani kellene.

## Úthálózat megvalósítása

Az úthálózat megvalósítására tehát a JOSM program használata bizonyult a legideálisabb megoldásnak, mivel ez felelt meg leginkább a projekt igényeinek és korlátjainak.

A hálózat elkészítésének folyamata három részre osztható:

* a JOSM programmal való megismerkedés: a kezelőfelület, eszközök és funkciók feltérképezése
* a hálózat szerkezetének megrajzolása: az utak, csomópontok és egyéb infrastruktúra-elemek vizuális felépítése a 2D-s felületen
* importálás a Webots környezetbe: a megrajzolt úthálózat exportálása megfelelő formátumban, majd annak átalakítása és tesztelése a szimulációban

### Ismerkedés a JOSM programmal

A program működését bemutató videó az OpenStreetMap Importer című Cyberbotics-dokumentációban található.

A felvételen a JOSM legfontosabb, alapvető funkciói kerülnek ismertetésre, különös tekintettel arra, hogyan módosítható egy már meglévő .osm fájl, majd hogyan importálható az a Webots környezetébe.

Bár a videó elsősorban a módosítási folyamatra fókuszál, a bemutatott eszközök segítségével a teljes úthálózat manuálisan is létrehozható, így a projekt szempontjából is jól hasznosítható ismereteket tartalmazott.

A projekt megvalósításához a következő három szerkesztőeszköz bizonyult a legfontosabbnak:

* kijelölés mód: az útszakaszok, csomópontok vagy épületek kiválasztására szolgál
* rajzolás: új útszakaszok létrehozására használható, a felhasználó egérkattintásokkal pontokat helyez el a síkon, amelyeket a program automatikusan összeköt, így határozva meg az útvonalat (később ezek a pontok fognak megjelenni „wayPoints” attribútumként a szimulációban)
* kettévágási mód: már meglévő útszakaszok felosztására szolgál

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

6. ábra: JOSM szerkeszőteszközök

### Úthálózat szerkezetének megrajzolása

A programot magyarázó videóban egy „my\_world.osm” nevű világgal mutatják be a program funkcióit és szerkesztőeszközeit.

Ez a világ szerkezeti felépítését és méretarányait tekintve is ideálisnak bizonyult a projekt igényeihez: elrendezése egyszerű, átlátható, ugyanakkor elegendő részletet tartalmaz ahhoz, hogy valósághű környezetet biztosítson a robot működésének teszteléséhez.



7. ábra: "my\_world.osm" a Webots környezetben

Az útszakaszok kialakítása kevésbé realisztikus, mint egy valódi városi úthálózaté, ami előnyös a projekt szempontjából, mivel így a robot navigációját nem befolyásolják a felesleges komplexitások, például a többsávos utak.

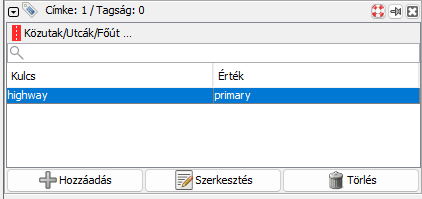
Ez a világ azonban nem érhető el semmilyen nyilvános forrásban, ezért a rekonstrukciója az előző fejezetben ismertetett módszer alapján történt. A JOSM-ben történő rajzolás során különös figyelmet kellett fordítani az arányok helyes megválasztására, mivel a méretek hibás beállítása esetén a létrehozott úthálózat a valóságosnál lényegesen nagyobb, akár több kilométeres, kiterjedésűvé válhat.

A képen művészet látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

8. ábra: az úthálózat rekreációja

Az így megrajzolt útszakaszokat ezt követően címkékkel (kell ellátni, amelyek meg határozzák az utak különböző tulajdonságait a szimuláció során. A címkék segítségével adhatók meg többek között az útburkolati jelek, a sávok száma, az út típusa (például főút, mellékút, gyalogút), valamint a forgalmi irány vagy a kötelező haladási irány.



9. ábra: „főút” címkével megjelölt útszakasz

Ezt követően az elkészült úthálózat .osm fájlként menthető. A mentett állomány ezután az 1.1.2. alfejezetben ismertetett módszerrel alakítható át a Webots szimulációs környezet által értelmezhető világfájllá (.wbt), így a megtervezett pálya közvetlenül betölthető és tesztelhető a programban.



10. ábra: az elkészült úthálózat a szimulációban

Ezzel az úthálózat azon részei, amelyekkel a robot közvetlenül kapcsolatba kerül, elkészültek. A projekt következő fázisaiban a hangsúly a pálya vizuális és funkcionális kiegészítésére helyeződik, amely magában foglalja a környezet részletgazdagabbá tételét, például épületek, fák és egyéb dekorációs elemek elhelyezését. Igény esetén a meglévő úthálózat is tovább bővíthető, hogy a szimuláció még életszerűbb és változatosabb legyen.

Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | T. De Smedt and W. Daelemans, “Pattern for python.,” *The Journal of Machine Learning Research,* vol. 13, no. 1, pp. 2063-2067, 2012. |
| [2] | „Záróvizsga információk,” [Online]. Available: https://mik.uni-pannon.hu/index.php/hu/oktatas/zarovizsga.html. [Hozzáférés dátuma: 04 03 2022]. |
| [3] | D. J. Wetherall és A. S. Tanenbaum, Computer networks, Pearson Education, 2013. |

Mellékletek

Mappaszerkezet

+chatbot

| backen.bat

| backend.py

| files.doc

| fixedlinks.json

| ipcheck.py

| linkek.json

| linkfix.py

| log.txt

| sqlwriter.py

| textprocessor.py

|

+---backend

| pyvenv.cfg

+---lara

| | entities.py

| | nlp.py

| | parser.py

| | stemmer.py

+---static

| +---css

| | chat.css

| |

| \---js

| chat.js

|

+---templates

| index.html

|

[PÉLDA!!! Megjegyzés: A Python csomagkezelője által telepített fájlok, illetve a különböző cache fájlok a fenti listából kimaradtak, mivel ezekkel indokolatlanul és aránytalanul hosszú lenne a fenti felsorolás. A beadott fájlok között azonban a teljesség kedvéért szerepelnek ezek a fájlok is.]

Ábrajegyzék

[1. ábra: ábrafelirat (ha szükséges, akkor a forrás megjelölésével) [1] 1](#_Toc97890941)

Táblázatjegyzék

[1. táblázat (forrás megjelölésével) [2] 1](#_Toc97890942)

1. A megfelelő rész aláhúzandó. Amennyiben a válasza NEM, akkor a következő bekezdést és táblázatot törölje a dolgozatból. [↑](#footnote-ref-1)