**Környezet mobil robotok csapatainak megerősítéses tanulásának**

**megvalósításához**

*Követelményelemzés*:

**Elképzelésem**: Az alapfeladat mindenképpen az lenne, hogy egy ilyen komplex mesterséges intelligencia programot, mint mobil robot megerősítéses tanítása, annak adjak egy olyat keretet amely segítségével a legtöbb nem hozzáértő ember is tudja használni probléma nélkül. Korábbi kutatásaim alapján, úgy véltem hogy linuxon sokkal több segítség elérhető mint a többi operációs rendszeren ezért mindenképpen abban a környezetben szeretném megoldani a feladatot. Mivel egy fontos része a feladatnak egy környezet létrehozása, amely sok embernek eljuthat, ezért ha lehetséges, szerintem mindenféleképpen megpróbálni a cross-platform alkalmazás kialakítását. A program alapvetően úgy működik, mint mondjuk egy videojáték menüje, különböző gombjai különböző funkciókat lát el. A legfontosabb része a tanítás lesz, ahol különböző Gazebo pályák (világok?) kiválasztásával eldönthető milyen világban szeretnénk megvalósítani a tanítást. Minden különböző pálya másfajta jutalom rendszerrel fog működni, erről mindenképpen tájékoztatást kell adni a felhasználónak, mivel a legfontosabb része a programnak a felhasználóbarát környezet kialakítása. A következő gomb szerepe egy olyan menü lenne, ahol a számítógépen található betanított modelleket összegyűjti és különböző statisztikák alapján sorrendbe rendezi őket. Lehetőséget szeretnék még adni arra hogy iteráció számot és epizód számot is tudjon átalakítani a felhasználó, ebben a menüpontban még legyen lehetőség a helyszín/pálya megváltoztatására is.. A másik legfontosabb a megjelenítés lesz, a tanítás közben lesz lehetősége a felhasználónak grafikus megjelenítésére, de úgy gondolom hogy nem lesz alapvetően bekapcsolva, hogy minél több erőforrás maradjon a tanításra, viszont a megjelenítés gombnál az előre mentett modellekből ki lehet választani a számunkra megfelelőt és azt megjeleníti a program. A program utolsó gombja egy segítség gomb lenne, ahol mint egy wikit szolgálna, amivel segítséget lehet a felhasználónak nyújtani, mivel és hogyan tudja a legtöbbet kihozni a programból a saját javára.

A programot a Linux környezeten kívül (16.04 Ubuntu), Python programozási nyelvben szeretném megvalósítani. Kutatásaim alapján arra jutottam hogy a linux plusz python a legnépszerűbb környezet mesterséges intelligencia alkalmazások megvalósítására. A grafikai megvalósítást a tkinter osztály fogja biztosítani. A mesterséges intelligencia résznél pedig a Turtlebot oldalán található útmutatót fogom használni, annak segítségével szeretném összekötni azt a projektet a sajátommal, esetlegesen módosítani azon ha a program megköveteli. Az előző féléves munkámat is át fogom nézni, mi szállítható át belőle ebbe az új projektbe.

Az alkalmazás célja hogy gyors, megbízható és környezetbarát legyen. Célja hogy minél kisebb mérete miatt könnyen szállítható legyen minden féle rendszerre, remélhetőleg minél több platformon.

**Programozási Terv:**

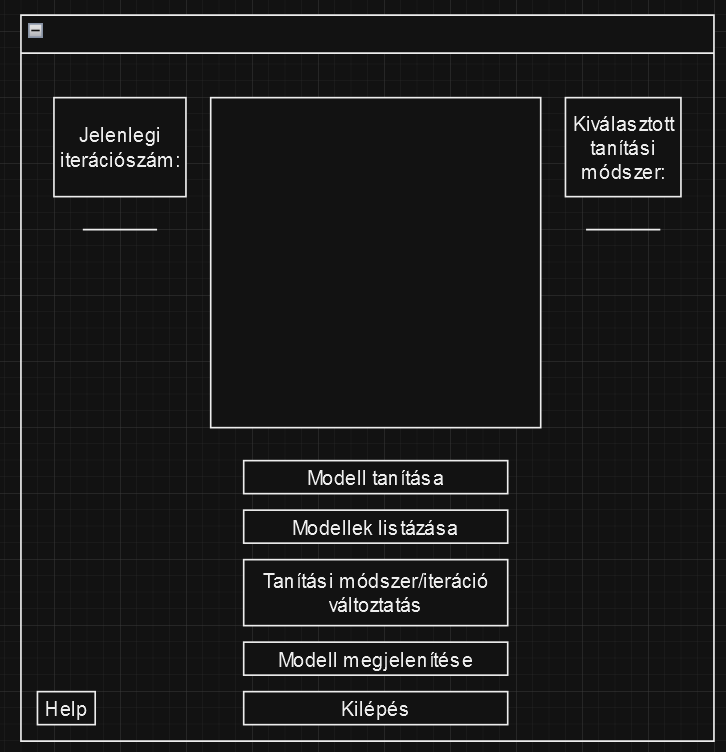
* Követelményelemzés megírása, elfogadása
* Alapvető funkciók megcsinálása, működjenek gombbal
* Előző program átemelése a jelenlegi programba
* Első verzió elkészítése, bemutatása
* Javítása, bővítése az első verziónak
* Második verzió (végleges?) bemutatása

**Megvalósíthatósági terv:**

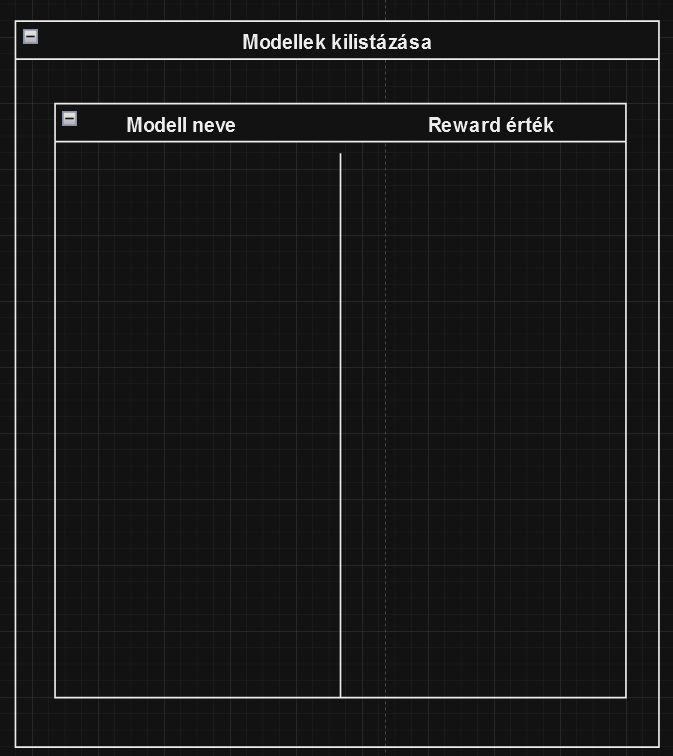
* Humán erőforrás: 1 tervező/fejlesztő (250 emberóra)
* Hardver erőforrás: 1 fejlesztői, tesztelői számítógép (Ubuntu 16.04)
* Szoftver erőforrások: fejlesztőkörnyezet (Visual Studio), verziókövető (Github)
* Üzemeltetés: nem kell biztosítani
* Karbantartás: nem kell biztosítani
* Megvalósítás időtartama: Összesen 250 emberóra

**UI Wireframe:**

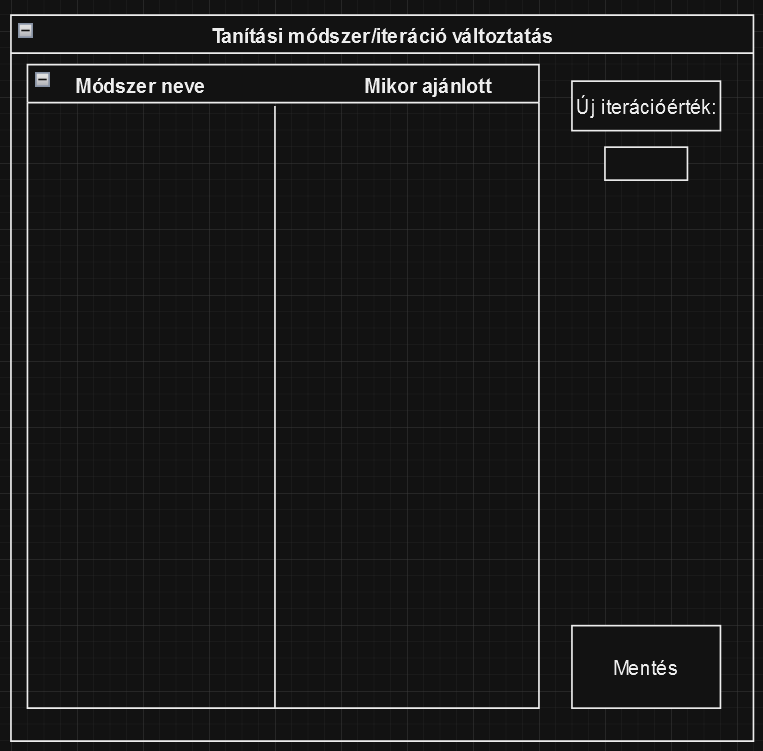
Main Page:



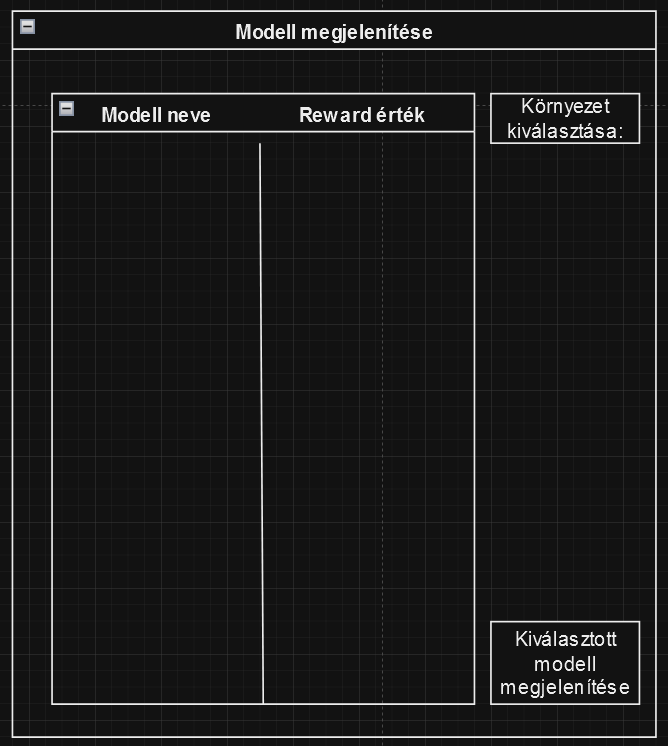
List models:



Change Iterations/Level:



Display Model:



**Nem Funkcionális Követelmények:**

* Hatékonyság: A program terhelés jelent a processzorra, memóriára és videókártyára, mentés esetén a háttértárat is terhelheti. A program nem igényel hálózati kapcsolatot. A menüpontok gyorsan reagálnak, a tanítás és megjelenítés részei a programnak időigényesek lehetnek. A program hatékony működéséhez egy jól felszerelt rendszer ajánlott.
* Megbízhatóság: Szabványos használat esetén nem fordul elő hibajelenség, és nem jelenik meg hibaüzenet. Ha valamilyen adat megsérül arról a program szólni fog a felhasználónak, akármilyen hiba esetén szól a felhasználónak, mit tegyen.
* Biztonság: A program offline működése miatt az adatok biztonságban vannak.
* Működési: Hosszabb használati idő, akár 5-6 óra is, gyakori használat várható
* Fejlesztési: Python nyelv, Linux rendszerben megvalósítva, Cross-Platform céllal

**Funkcionális követelmények:**

Alapprogram:

* Tanítás lehetősége
* Tanított modellek kiírása, statisztikájának megjelenítése
* Iterációszám/Helyszín megváltoztatása
* Kiválasztott modell megjelenítése a környezetben
* Középen egy kijelző amelyen ki vannak írva milyen folyamatok zajlanak éppen
* Egy segítség gomb, ahol lennének magyarázatok azok számára akik jobban szeretnék megérteni a működését a programnak

Iterációszám megváltoztatása:

* Iterációszámnak üres mező ahol meg lehet adni milyen új hossz legyen (ajánlott megadása itt is)
* Helyszínváltoztatás

Tanított modellek kiírása, statisztikájának megjelenítése:

* Modellek nevének, hosszának kiírása
* Reward alapján rendszerezett modellek

Kiválasztott modell megjelenítése a környezetben:

* Modellek kiírása reward alapján csökkenő sorrendben
* A kiválasztott modell megjelenítése
* Környezet kiválasztása menüpont

Tanítás lehetősége:

* Egy külön menü, ahol ki lenne írva éppen hol tart a tanítás jelenleg
* Csak a fontos, könnyen értelmezhető értékek megjelenítése