XML Maze Generator

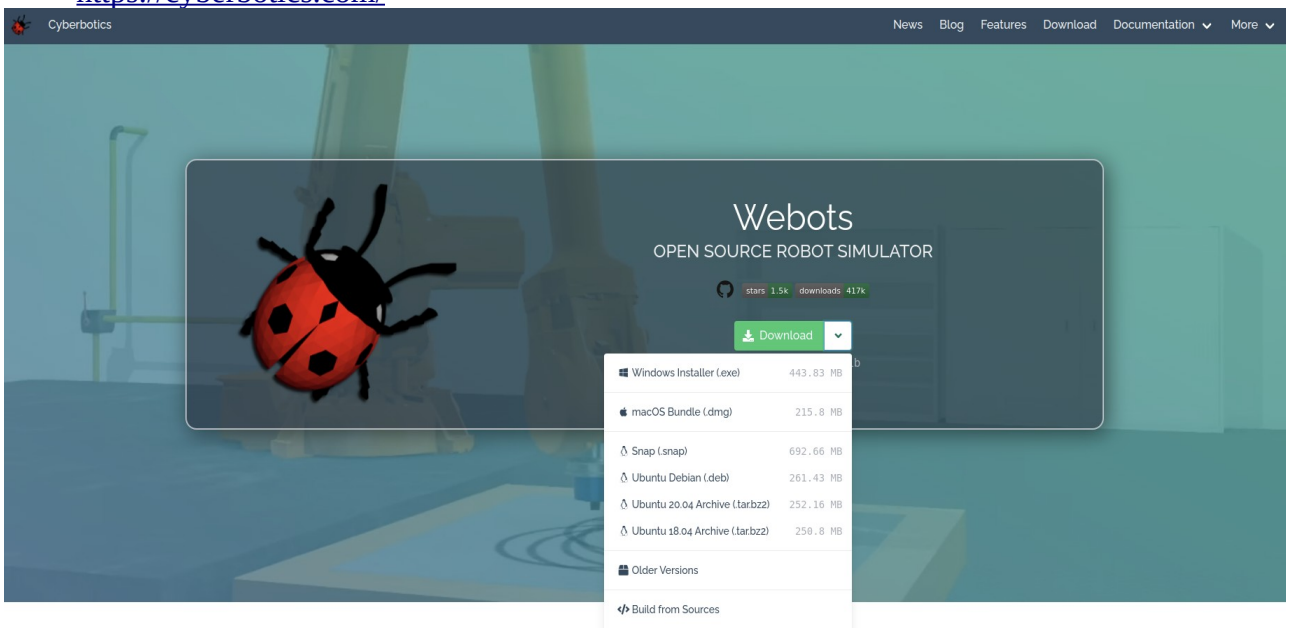
Description

The project is designed to take a XML file and translate the coordinate table into a wbt world maze file. The program is designed to use the maze parser file and translate the x, y and z values into a Pandas table. The supervisor file then iterates through the Pandas table and writes the x, y and z values into the world file

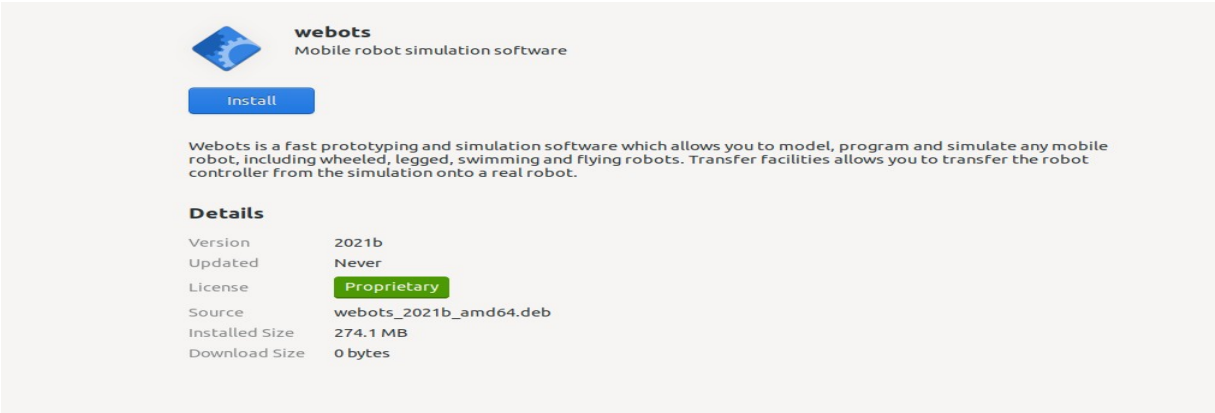
INSTALLATION AND BUILD INSTRUCTIONS AND RUN INSTRUCTIONS

1. Make sure Ubuntu 20.04 is installed. Use either of the instructions below:
   - USB Installation - https://www.youtube.com/watch?v=CFI1Jt8kVUk
   - Dual Boot Installation - https://www.youtube.com/watch?v=-iSAyiicyQY
   - Ubuntu Virtual Machine - https://www.youtube.com/watch?v=x5MhydijWmc

2. Install Webots 20.04 – Make sure the installation is not Snap.
   https://cyberbotics.com/

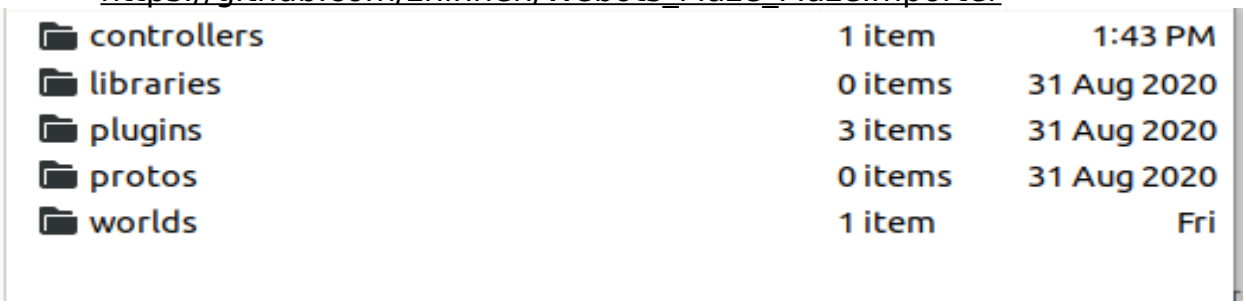


   Make sure that you install the webots Debian package when using Ubuntu Click on the Webots download at the bottom left part of the page and select software installer.

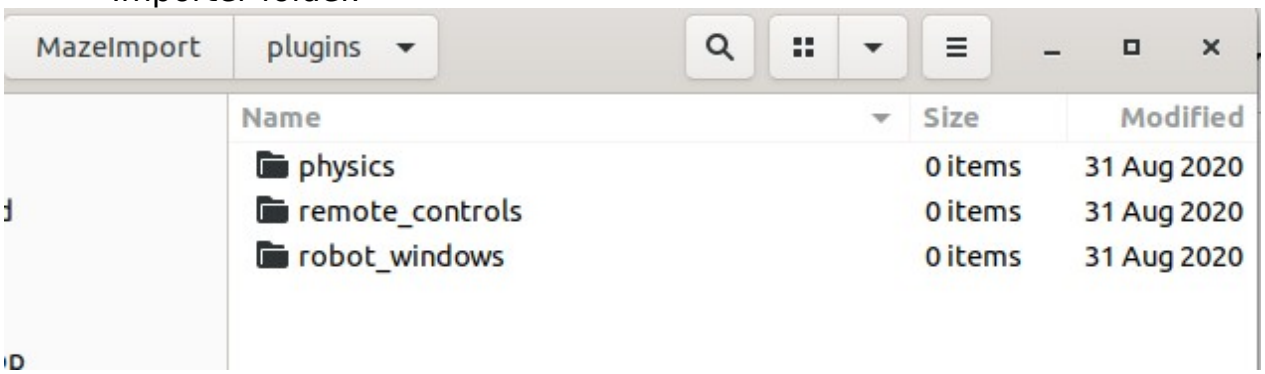


Click install and the webots app simulation will be installed on your system.

3 Download the files from github and make sure all the correct folders are built:
   - Github Repository: https://github.com/zhinnen/Webots_Maze_MazeImporter

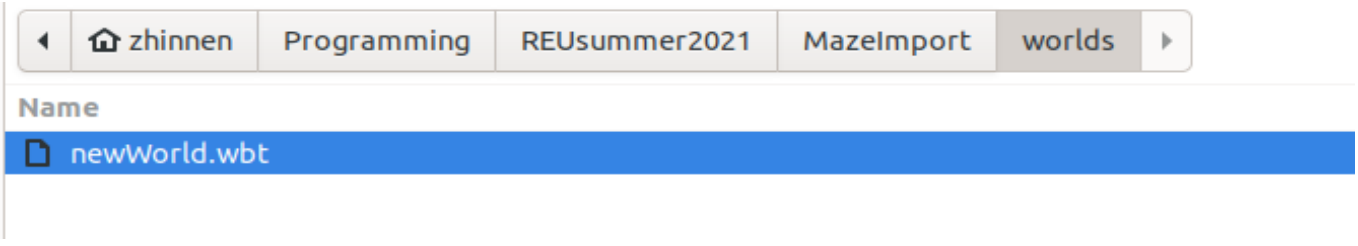


   - Make sure that the above folders are correctly setup up in the Maze Importer folder.

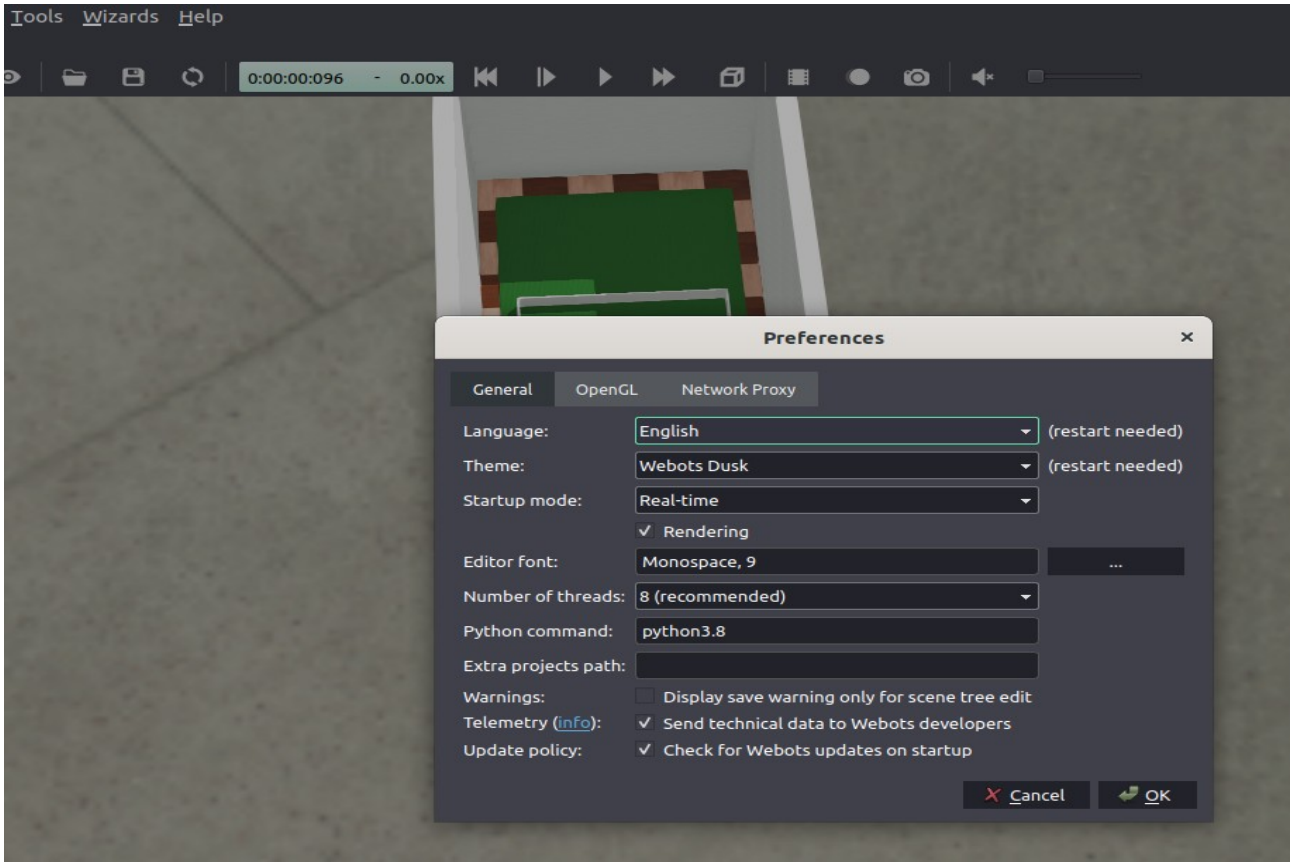


   - These are the necessary plugin folders that need to be set up
4 Open Webots and Launch the base world located in the world file
   - Go to file and sele ct open world. Go to the location of your github installation and select the world file that you want to open.

5. Make sure to adjust preferences when the world is open to run python3.8 instead of the default python.



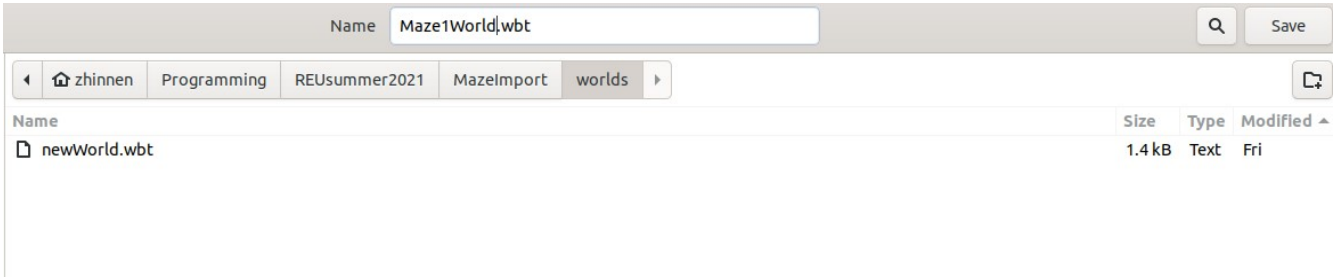
6. Make sure that the XML file is located within the controller folder. Based on the maze
you want generated, change the XML file name within the .parse_maze() location. Once that is set up, check the press the play button and let the algorithm run. Once a maze has generated, save the newly generated world under a new file name with the appropriate maze.



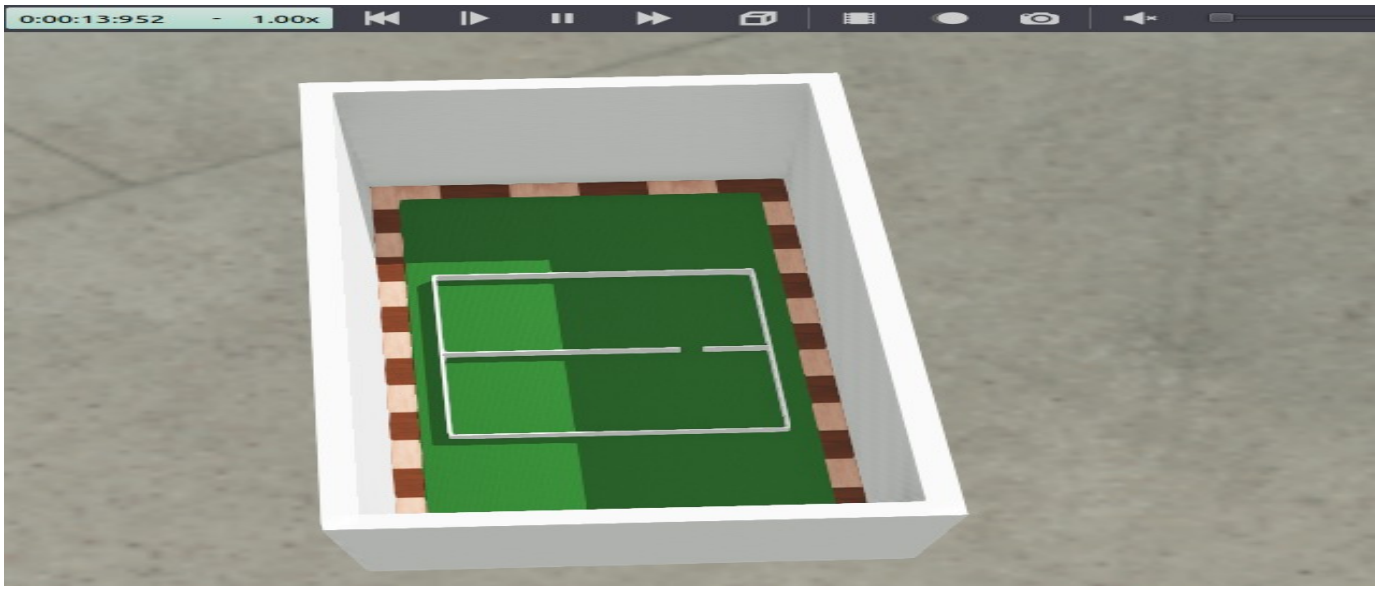

```
# ================ MAIN FUNCTION ===================================
i=1
walls, feeders, start_positions = MazeParser.parse_maze("M1.xml")
for index, row in walls.iterrows():
      add_maze_wall(row,i)
      i = i+1

while (supervisor.step(timeStep) != -1):
  pass
```

Listed above is where to change the XML file name.

Listed below is where to run the program and what the file should look like when generated.



Set the save file to the new maze



7. Open the newly generated world file and insert any robot into the maze. Build a controller and set the robot to the python controller.

Final Notes:

The code works through running the dummy.py file through webots. It will make changes to the base world (which is fashioned to match similar dimensions to the robotics lab). Additional changes to the code can include automating the XML files that will be converted. The idea for the change is to make it so that the user does not have to change the XML input manually, but can simply iterate through a folder and output mazes with matching conditions. This will also require the user to find a way to save the current maze file under a different name every time, and resetting the base world file to its original format to prevent mazes from being generated on top of each other.