

## Chess\_game\_Auto

Sunday, December 29, 2024 6:16 PM

## for ROS2 gazebo11

If you face the problem that your gazebo stuck at some point that can't be open

Try pkill -9 gz

# Our robot model is "Franka Emika Panda robot"

For updating ur workspace code

You need to first: cd ros2\_ws(or your workspace path)

Then do these:

1. colcon build --packages-select chess\_manipulator (or use colcon build will build the whole branch)
2. .install/setup.bash (or use source install/setup.bash ##recommend)
3. ros2 launch chess\_manipulator simulation.launch.py
4. ros2 run chess\_manipulator example\_game

Then you are good to go

## ## For Reinforcement learning

In the conda env (chess)

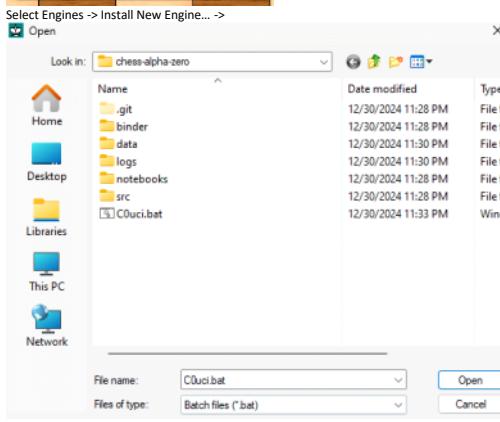
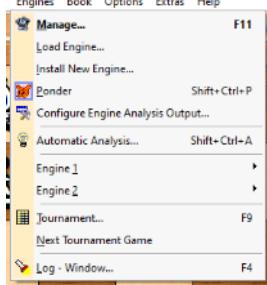
Cd to where you clone github repository

Then use command below to do

1. Make new best  
python src/chess\_zero/run.py --cmd self
2. Trainer  
python src/chess\_zero/run.py --cmd opt
3. Evaluator  
python src/chess\_zero/run.py --cmd eval

#chess-alpha-zero  
#Arena download <http://www.playwitharena.de/> - for windows

Inside Arena:



-> select you COuci.bat file (open it) then load the Engine.

#change COuci.bat file to below: so that our anaconda env can get the data

```
@echo off
call C:\Users\jack2\anaconda3\Scripts\activate.bat chess
cd /d "%dp0"
python src/chess_zero/run.py --cmd uci --type mini
```

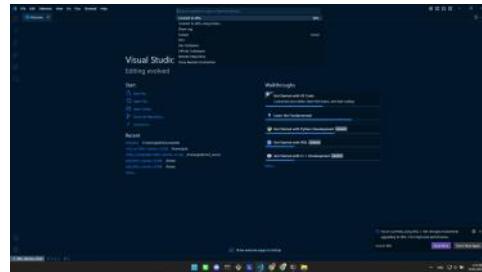
```
1 @echo off
2 call C:\Users\jack2\anaconda3\Scripts\activate.bat chess
3 cd /d "%dp0"
4 python src/chess_zero/run.py --cmd uci --type mini
```

```
2024-12-30 23:10:56,671@chess_zero.agent.model_chess DEBUG # loaded model digest = 05872dc36fbdb8138698cee172ab9a613f2bf5
eca792a77491a9fa62427e46857
Using TensorFlow backend.
2024-12-30 23:10:57.081737: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_100.dll'; dlsym: cudart64_100.dll not found
2024-12-30 23:10:57.082133: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
game 1 time= 81.1s halfmoves=158 Winner.draw
game 2 time= 42.8s halfmoves= 74 Winner.black
game 3 time= 15.7s halfmoves= 29 Winner.white
```

## Packages you'll need in this project

1. Chess\_manipulator(panda)
2. Universal\_Robots\_ROS2\_Gazebo\_Simulation(UR3, 3e, 5, 5e, etc.)
3. chess-alpha-zero (for sending reinforcement learning)
4. Chess\_robot\_ai (for receiving reinforcement learning model)
5. Ur\_chess\_controller (for controlling and testing UR)

You can choose the first one ↓



Bottom right we can change our python env.

Bottom left we got an small icon that can change the terminal from windows to WSL(windows subsystem for linux) then we can select and switch to the Ubuntu system.

## Movelt2:

A motion planning framework that helps you:

1. Plan robot arm movements while avoiding collisions
2. Calculate how to move joints to reach specific positions (inverse kinematics)
3. Create grasping motions for picking up objects
4. Visualize and test movements before executing them

For your chess project, Movelt2 would help:

- Plan safe paths to reach chess pieces
- Calculate exact joint positions to grab pieces
- Avoid colliding with other pieces on the board
- Execute smooth movements

Movelt2 integrates with Gazebo, so you can:

1. Test movements in simulation first
2. Use the same code for both simulation and real robots
3. Visualize planned paths in both RViz and Gazebo

## How to copy one file/ whole directory using command line?

For a single file:

```
cp /path/to/source_package/file.py /path/to/target_package/file.py
```

entire directory and its contents:

```
cp -rp /path/to/source_package/directory/* /path/to/target_package/directory
```

```

game 1 time= 56.2s halfmoves= 84 Winner.black
game 2 time= 50.7s halfmoves=102 Winner.black
game 3 time= 82.8s halfmoves= 98 Winner.black
game 4 time= 5.5s halfmoves= 13 Winner.white
game 5 time= 24.7s halfmoves= 44 Winner.black
game 6 time= 56.2s halfmoves= 64 Winner.black

```

Then we tried to make connection between window system Anaconda and WSL system Ubuntu,  
Basically we are trying to implement:

1. Chess\_model\_server.py in anaconda env: inside src, inorder to give the model info to ROS2
2. We create whole new directory for client (ROS2), then we make a file call chess\_robot\_node.py and we setup setup.py and package.xml to be compatible to the system

```

Model loaded successfully!
Starting chess model server on http://localhost:5000
* Debugger is active!
* Debugger PIN: 105-800-352
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [31/Dec/2024 21:21:16] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:21] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:26] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:31] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:36] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:41] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:46] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:51] "GET /health HTTP/1.1" 200 -
127.0.0.1 - - [31/Dec/2024 21:21:56] "GET /health HTTP/1.1" 200 -

```

This is Anaconda prompt while we use command "python chess\_model\_server.py"

```

[INFO] [1735651276.766110800] [chess_robot_node]: Robot initialized successfully
[INFO] [1735651276.766953200] [chess_robot_node]: Waiting for model server...
[INFO] [1735651276.774352300] [chess_robot_node]: Connected to model server!

```

This is Ubuntu while we run "ros2 run chess\_robot\_ai chess\_robot\_node"  
After connect to the anaconda env, self play chess game will start, the robot will predict the move and send it to gazebo env, then the manipulator will receive the info and move. (# notice here if you didn't close the server, and rerun the client code, model will be continued, which means it won't restart the board, don't forget to restart the server.)

#### Current problem:

Can't reach to the end of the chess board, let's say a8a6, it not possible for panda robot(the one we are using now) move to that position.  
So instead, I tried to implement UR10(Universal Robot) as an alternative manipulator arm to solve it.

Accompany with MoveIt: What is it?

- Plan collision-free paths for your robot arm
  - Calculate inverse kinematics (converting chess board positions to joint angles)
  - Handle object manipulation (grasping chess pieces)
  - Avoid collisions with the chess board and pieces
- In order to download it, go to your ROS2 workspace and use the command below(Here we use ROS2 humble env)
- ```

cd ~/{your_workspace}
sudo apt-get update
sudo apt-get install ros-humble-moveit
sudo apt-get install ros-humble-ur-msgs ros-humble-ur-client-library

```

The clone the repository of UR drivers( If you face problem while colcon this repository, it probably mean your ROS2 is not compatible with it, go to github repo to see whether u choose the compatible brach for your project like below. For example(not main but humble)

**Universal\_Robots\_ROS2\_Driver** Public

humble 7 Branches 55 Tags

Go to file + Code

This branch is 126 commits ahead of 232 commits behind main.

mergify[bot] and bmagyar Remove unused include (backport of #1220) 1cd4e05 · 4 days ago 435 Commits

- .github Remove dependency to docker.io (backport of #985) ... 8 months ago
- ur 2.5.1 last week
- ur\_bringup 2.5.1 last week
- ur\_calibration 2.5.1 last week
- ur\_controllers 2.5.1 last week

Universal Robots ROS2 driver supporting CB3 and e-Series

Robotics ros-industrial ros2 ros2-foxy  
ros2-galactic ros2-rolling ros2-humble

About

Readme  
BSD-3-Clause license  
Activity  
Custom properties  
450 stars  
17 watching

**Universal\_Robots\_ROS2\_Gazebo\_Simulation** Public

humble 4 Branches 0 Tags

Go to file Add file Code

This branch is 9 commits ahead of 14 commits behind main.

mergify[bot] and fmauch Update README (backport #80) (81) 9bfb1c2 · 7 months ago 34 Commits

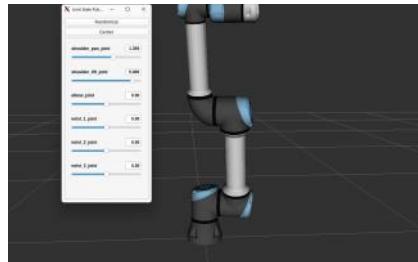
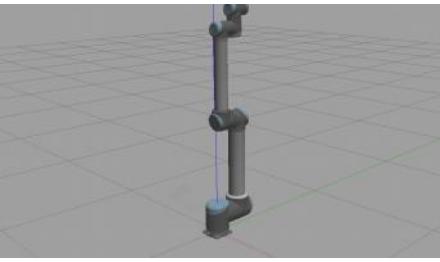
- .github/workflows Remove all non-humble CI and update README (#40) 2 years ago
- ur\_simulation\_gazebo Add an if-condition to RViz delay (#77) 8 months ago
- .clang-format Initial CI setup and configuration. 4 years ago
- .pre-commit-config.yaml Remove galactic workflows and updated upstream repos (#2...) 2 years ago
- CONTRIBUTING.md Make workflows run again (#32) 2 years ago

No description, website, or topics provided.

Readme  
BSD-3-Clause license  
Activity  
Custom properties  
76 stars  
11 watching

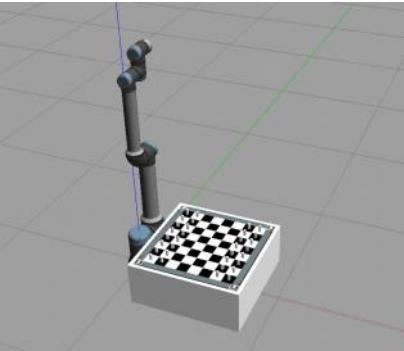
Clone it and run it then you can have the result below.





We can finally use it UR10 and other version on gazebo!!! 😊😊 very nice  
 ros2 launch ur\_simulation\_gazebo ur\_sim\_control.launch.py ur\_type:=ur10e (3, 5, 5e, 10, 10e, etc...)

Use this command then you can test the robot's every joints to see how it will move, this is inside this repo below  
[https://github.com/UniversalRobots/Universal\\_Robots\\_ROS2\\_Description](https://github.com/UniversalRobots/Universal_Robots_ROS2_Description)



So I add the chesset world inside as well,  
 Right now use the command above we can see UR10e with chessboard, perfect!! 🎉

#### Problem solve

`touch: cannot touch 'URClass.py': Permission denied`  
`ls -l ~/ros2_ws/src/Universal_Robots_ROS2_Gazebo_Simulation/ur_simulation_gazebo/ur_simulation_gazebo`

Look inside:  
`-rw-r--r-- 1 root root 0 Jan 4 12:18 __init__.py`

Which means it's not currently under the permission of ur host,  
 Change it by using

`sudo chown -R jack:jack ~/ros2_ws/src/Universal_Robots_ROS2_Gazebo_Simulation(change the path to the path you want)`  
`-rw-r--r-- 1 jack jack 0 Jan 4 12:18 __init__.py`

Then you can use touch to create a python file

ros2 run ur\_chess\_controller example\_game use it to test UR.

It's can move to the particular position right now, but it not stable, so I tried a lot of way to make it better

1. Reduce the tolerance(so at least we can move)
2. Make the loop, only execute it when the plan path is acceptable(doesn't know which function control that, hard to implement)
3. Set a height constraint, make robot can not go over some height, so it won't collide with chessboard, doesn't work

So , nvm, change it in the future!! 😊😊

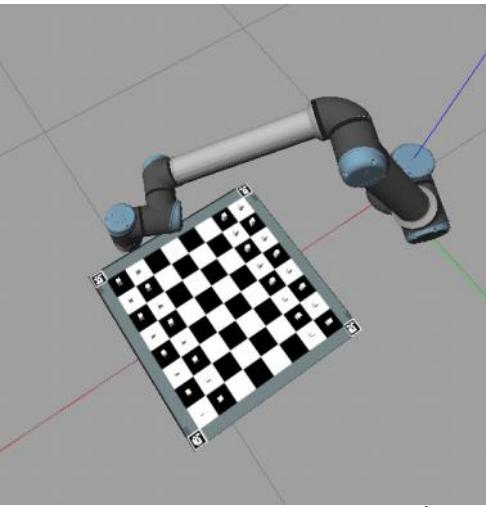
# problem found:

I try to use Moveit2, then I put on my bashrc file, it will causing our example game not working functionally, just command it when you are not using it.

`# source ~/ws_moveit2/install/setup.bash`

Also , the path planning sucks, we need to optimize it or it will crashes each time we control the robot, and if we modify the threshold to lower value.  
 The robot will just don't move at all, so it still unfix, I tried to use pathplanner class to control it, but still have some problem, solve it in the future perhaps.

Right now I'll try to move the chessboard to a little bit far away to see whether distance between manipulator arm will be more stable? I hope it work



Ya it works, sometimes still crash tho, but it has more stability right now, perfect 🎉

To use reinforcement learning with UR10 or other version of UR  
 Use this command: `ros2 run chess_robot_ai chess_robot_node_ur`