RLBench Setup for HPC3

Overview

We translate the following github commands to be HPC3 compatible.

- https://github.com/stepjam/RLBench/tree/master?tab=readme-ov-file#install

1. We have to use the Ubuntu18_04 version of CoppeliaSim since 20_04 requires GLIBC=2.29 (for pyrep) and Rocky 8.10 Green Obsidian only has GLIBC=2.28.
2. Our remote machine can't git clone for some reason so manually download the compressed repository and then pip install directly.
3. Run headless by starting an xvfb server similar to x server from
    1. https://github.com/stepjam/PyRep?tab=readme-ov-file#running-headless

Python Venv

Create the virtual environment

```
mkdir project
cd project
python –m venv venv
```

Activate the virtual environment

```
source venv/bin/activate
```

Install CoppeliaSim

```
export COPPELIASIM_ROOT=${HOME}/CoppeliaSim
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$COPPELIASIM_ROOT
export QT_QPA_PLATFORM_PLUGIN_PATH=$COPPELIASIM_ROOT

wget
https://downloads.coppeliarobotics.com/V4_1_0/CoppeliaSim_Edu
_V4_1_0_Ubuntu18_04.tar.xz
mkdir -p $COPPELIASIM_ROOT && tar -xf
CoppeliaSim_Edu_V4_1_0_Ubuntu18_04.tar.xz -C
$COPPELIASIM_ROOT --strip-components 1
rm -rf CoppeliaSim_Edu_V4_1_0_Ubuntu18_04.tar.xz
```

Manually Install PyRep

```
wget
https://github.com/stepjam/PyRep/archive/refs/heads/master.zi
p
unzip master.zip
cd PyRep-master
pip install .
cd ..
```

Manually Install RLBench

```
wget
https://github.com/stepjam/RLBench/archive/refs/heads/master.
zip
unzip master.zip
cd RLBench-master
```

Change only the pyrep requirement in `setup.py` to

```
core_requirements = [
    "pyrep",
```

```
    ...
```

```
pip install .
cd ..
```

Cleanup

```
rm -rf PyRep-master RLBench-master master.zip
```

Install Packages

Enable using OpenAI gym API

```
pip install gym
pip install gymnasium
```

For generating video

```
pip install imageio[ffmpeg]
```

Test Script

Create `test.py`

```
import gymnasium as gym
from gymnasium.utils.performance import benchmark_step
import rlbench

import imageio
import time

env = gym.make('rlbench/reach_target-vision-v0',
render_mode='rgb_array')
```

```
fps_env = benchmark_step(env, target_duration=10)
print(f"Environment FPS: {fps_env:.2f}")

frames = []

start = time.perf_counter()
training_steps = 120
episode_length = 40
for i in range(training_steps):
    if i % episode_length == 0:
        print(f'Reset Episode, Time: {time.perf_counter() -
start:.5f}s')
        obs = env.reset()
    obs, reward, terminate, _, _ =
env.step(env.action_space.sample())

    frame = env.render()
    frames.append(frame)
print('Done')

output_filename = 'robot_simulation.mp4'
fps = 30
imageio.mimwrite(output_filename, frames, fps=fps)
print(f"Video saved as {output_filename}")

env.close(
```

Create a corresponding `submit_test.sh`

```
#!/bin/bash
#SBATCH -A cs175_class_gpu    ## Account to charge
#SBATCH --time=04:00:00       ## Maximum running time of
program
#SBATCH --nodes=1             ## Number of nodes.
```

```
                                ## Set to 1 if you are using
GPU.
#SBATCH --partition=gpu         ## Partition name
#SBATCH --mem=30GB              ## Allocated Memory
#SBATCH --cpus-per-task 8       ## Number of CPU cores
#SBATCH --gres=gpu:V100:1       ## Type and the number of GPUs

# Start xvfb server
DISPLAY_NUM=99
Xvfb :${DISPLAY_NUM} -screen 0 1280x1024x24 +render -noreset
&
export DISPLAY=:${DISPLAY_NUM}
sleep 2

# Run script
python -u test.py

killall Xvf
```

Run the script

```
sbatch submit_test.sh
```

Environment Variables

Insert the environment variable definitions into `~/.bashrc` so it is setup on start.

```
export COPPELIASIM_ROOT=${HOME}/CoppeliaSim
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$COPPELIASIM_ROOT
export QT_QPA_PLATFORM_PLUGIN_PATH=$COPPELIASIM_ROOT
```