FreeCiv Python Wrapper

— Yash Bonde

It is important to have strategy game based RL environment, since all the major ones that exist as of now are FPS or Atari or action games where immediate results can be determined relatively easily by looking at the frame. The other end of the spectrum has robotics research environments. A Freeciv environment will allow for a strategy games that have text based, image based information to be processed for long term reward gathering. Which is more similar to real world environment than Atari game, though in no way am I putting down the earlier work. We are standing on the shoulders of a giant.

For now we are simply combining the backend to the python and make an MVP (minimum viable product). This MVP should allow the agent to play the game. Initial release needs to have small mini-games much like Deepmind Starcraft-II environment, **Config files** are important as it will tell the game structure. [see how to do it]

```
freeciv: This is the main folder that has all the files

world.py: This file has the World class that which is what we call for running the
      |    environment
      |
      |- __init__(): setup the configurations of the game by reading the
      |              config file
      |
      |- init(): Initialise the environment by doing the following:
      |          1) connect to the game server using sockets in cpp binder
      |          2) send the config information
      |          3) start the game
      |
      |- start_game(): Start the game by doing the following:
      |                1) check player stack
      |                2) let the player above in stack do their turns
      |                3) if all other players have played their turns return True
      |                   so the agent can play the game
      |
      |- get_states(): return three different maps, resource_map, units_map and
      |                holding_map (control). [More needs to be clarified here]
      |
      |- get_[units/global_score](): As the name suggests
      |
      |- update(): update the world by sending the action information to the
      |            backend binder
      |
      |- [more functions need to be added]


unit.py: Unit class has information about the unit and some basic functions related
        to it.

standard_utils.py: has basic (non-game) utility functions

game_utils.py: has game utility functions
```

```
Binder files: These are the files that connect to the required backend [need to
start work on those]
```

Future Work

Need to combine the freeciv-web with this so we can perform easy visualisation and get simpler and more easier training data. This opens up possibilities for larger training methods like imitation learning, semi-supervised methods, generative methods. Visualisation of the models will also make debugging faster and will bring more attention to the game.