==================================================

Running an Agent in pysc2.env.sc2\_env.SC2Env a

==================================================

At each timestep, the agent.step function is called with an observation in the form of a TimeStep object, ie.

agent.step(obs)

where obs is a TimeStep object.

agent.step(obs) is then expected to return an action with subaction arguments, ie.

return actions.FunctionCall(\_NOOP, [])

obs is a TimeStep object with the following properties

TimeStep.step\_type

step\_type=<StepType.FIRST: 0> if this is the first step

step\_type=<StepType.MID: 1> if this is neither the first nor the last step

step\_type=<StepType.LAST: 2> if this is the last step

TimeStep.reward

a scalar value that denotes the reward for the TimeStep

TimeStep.discount

a scalar value in range [0, 1]

TimeStep.observation

a dictionary with the following key-value pairs

'cargo': (n, 7) tensor, for each unit in a transport, refer to 'single\_select'

'minimap': (7, 64, 64) tensor, in the following order

[0]: height\_map

[1]: visibility\_map

[2]: creep

[3]: camera

[4]: player\_id

[5]: player\_relative

[6]: selected

'game\_loop': (1,) tensor showing game step

'available\_actions': (n,) tensor showing ids of valid actions

'screen': (17, 84, 84) tensor, in the following order

[0]: height\_map

[1]: visibility\_map

[2]: creep

[3]: power

[4]: player\_id

[5]: player\_relative

[6]: unit\_type

[7]: selected

[8]: unit\_hit\_points

[9]: unit\_hit\_points\_ratio

[10]: unit\_energy

[11]: unit\_energy\_ratio

[12]: unit\_shields

[13]: unit\_shields\_ratio

[14]: unit\_density

[15]: unit\_density\_aa

[16]: effects

'control\_groups': (10, 2) tensor, for each of the control groups

[0]: unit leader type

[1]: count

'multi\_select': (n, 7) tensor, for each currently selected unit, refer to 'single\_select'

'cargo\_slots\_available': (1,) tensor showing number of slots available in transport

'player': (11,) tensor, for general information

[0]: player\_id

[1]: minerals

[2]: vespene

[3]: food used / supply

[4]: food cap

[5]: food used by army

[6]: food used by workers

[7]: idle worker count

[8]: army count

[9]: warp gate count

[10]: larva count

'single\_select': (1, 7) tensor, for the currently selected unit

[0]: unit type

[1]: player\_relative

[2]: health

[3]: shields

[4]: energy

[5]: transport slot taken if in transport

[6]: build progress (%) if being built

'build\_queue': (n, 7) tensor, for each unit in a build queue, refer to 'single\_select'

'score\_cumulative': (13,) tensor, showing score information

[0]: overall score

[1]: idle production time

[2]: idle worker time

[3]: total value of own units

[4]: total value of own structures

[5]: total value of killed units

[6]: total value of destroyed structures

[7]: collected minerals

[8]: collected vespene

[9]: mineral collection rate

[10]: vespene collection rate

[11]: spent minerals

[12]: spent vespene