Experiment pitch for behavioral modelling using ViZDoom

Sean Mulready

2024-01-19

1 The basics

- Participants play ViZDoom in the basic scenario, where the goal of an episode[1] is, to move left or right to shoot a monster at the opposite side of the map with a gun
- There is a reward-system: a living reward of -1 (so each tic[2] get's a reward of -1 no matter what), a shooting reward of -5 and a killing reward of +106. So in consequence the reward is -1 for moving or standing, -6 for shooting and missing (-1+(-5)) and 100 for killing the monster (-1+(-5)+ 106).

[1] An episode is one runthrough of the game, so from start(loading of the screen) to finish(goal achieved, i.e. monster is dead)

[2] tics are the ingame-time, one tic is one runthrough of the ingame-loop (getting state(positions, ammo, health,…), action and reward). By default it is set to 35 tics per second, so one tic = 1/35 of a second ($\approx$ 28 ms). The ticrate can be set higher or lower

2 The idea

- recording of the game episode-wise, afterwards analyzing the episodes tic-wise and store data in a .csv-format for further processing
- goal for players is to maximize their reward
- introducing an exploitation-exploration-dilemma: for each block (with let's say 300 Episodes) a random choice will give a parameter we call μ a value out of $0, 1$. In relation to the value of μ the probabilities of a normal or inverted[3] movement-control in relation to the side on which the monster spawns change. The movement to the wrong direction[4] is rewarded with at least -2 or even less. To maximize their reward participants have to decide to exploit or explore to get to the monster as fast as possible without risking to loose reward by moving in the wrong direction. The following table should illustrate this principle. I am using example-probabilities for the movement of 0.7 and 0.3:

μ	monster left	monster right	p_μ(movement = normal)	p_μ(movement=inverted)
0	yes	no	0.7	0.3
0	no	yes	0.3	0.7
1	yes	no	0.3	0.7
1	no	yes	0.7	0.3

- After gathering data from participants the goal is to use different models and let agents play the scenario following these models when making decisions to see, which modells recreates the measured data the best.

3 Things have to be thought about/solved

- how to realize the ingame-coordination of different movements and corresponding rewards without making the game slower (maybe having the probabilty-based decision on movement for each episode made before the block starts so game can follow simple instructions)
- aesthetics (mouse is still shown in the game-window)
- implementing the scripts into a code for the whole Experiment (including on-screen-instructions, testruns, ….)
- …

[3] so keys for left and right are swapped

[4] Further away from the monster, to the left when monster is right and vice versa.