



Istanbul Technical University Robot Olympics 2018

Simulation Category Rules

Definition of the Task:

- In this category cat tries to catch the rat in the given time interval, rat tries to not be caught in the given time interval.

Success Criterion:

- If a competitor does not send data for 7 consecutive seconds, that competitor is considered defeated in that round

Communication:

- 1) Competitors are going to communicate with the game using websocket.
- 2) Competitors are going to send data in turns.
- 3) The data competitors send are going to be in json format.
- 4) Data of the first player that sends the data to game is going to be processed first
- 5) Competitors are going to send their character type for that round and their password(string) to the game using subprotocols(passwords are going to be given at the start of each round).

Competition:

- 6) Competition will be conducted of 2 rounds of dual match between algorithms.
- 7) Competitors can use any language they want to program their algorithm.
- 8) Both players are going to play one cat round and one rat round where the round is going to be cat versus rat.
- 9) Time limit for each round is 5 minutes.
- 10) If the cat comes within 2.5 units of distance of the rat, cat wins.
- 11) If only one player does not send data for 7 consecutive seconds, that player is considered defeated for that round.
- 12) If both players do not send data for 7 consecutive seconds in a round, the result of the match is going to be based on the second round.
- 13) If there is no winner in both of the rounds, the match is canceled and both players are disqualified.



- 14) If in both rounds cat fails to catch the rat, result of the match is going to be based on the distances between cat and rat at the end of each round.
- 15) If in both rounds cat catches the rat, result of the match is going to be based on the remaining time.
- 16) Game map is going to be 300 units by 300 units(left_motor = 10, right_motor=10 moves the character by 1 unit in the forward direction), x and z axis are in the interval from -150 units to 150 units in coordinate system while the y axis is the height.
- 17) Game map is going to be surrounded by walls and there will be a labyrinth in the inside.
- 18) Competitors are controlling spheres whose radiuses are 5 units.
- 19) Competitors move by sending data for right and left motors(Like line follower).
- 20) If a competitor sends a second data before the other competitor sends one, that data is not going to be processed. After the second competitor sends their data, last data sent by the first competitor excluding the first one is going to be processed.

Datas:

- 21) Game starts with both competitors connecting to the websocket and receiving "game started" data.
- 22) "Game started" data consists of title(string) value.
- 23) Competitors respond to "game started" data by sending "move" data.
- 24) "Move" data consists of title(string), left_motor(int/float) and right_motor(int/float) values.
- 25) Game, sends "move response" data after receiving and processing the data sent by each competitor.
- 26) "Move response" data consists of title(string), x(float), y(float), z(float) and view(distances).
- 27) View(distances) consists lines every 1 degree for 360 degrees around y axis. Lines start from the competitor's sphere and go until it hits a shape. View value consists of length of these lines(distance)(float), the shape it hits(pickedMesh)(string) and angles(int).
- 28) Move and Move response loop keeps going until time of the round runs out.



- 29) If a competitor sends a variable type that is different than float or int for right or left motor value, game sends them “wrong velocity type” data.
- 30) “Wrong velocity type” data consists of title(string) and content(string).
- 31) If competitor sends extra data(Article 19), game sends “wait for move” data to the competitor.
- 32) “Wait for move” data consists of title(string) and content(string).
- 33) Once the time is up, game sends “game over” data to the competitors and winner of the round is declared.

Data Examples:

- 34) Example websocket connection url(for cat)(url will be different):

```
ws://foo.bar.org/game/ituro-2018/match/74/cat
```

- 35) Example subprotocols(python example for cat):

```
subprotocols=["cat", cat_password]
```

- 36) “Game started” data example:

```
{'title': 'game_started'}
```

- 37) “Move” data example:

```
{'left_velocity': 10, 'right_velocity': 10, 'title': 'move'}
```

- 38) “Move response” data example(each angle until 5 degree instead of all 360 degrees):

```
{'distances': [{'distance': 19.170658286879345, 'angle': 0, 'pickedMesh': 'ground'}, {'distance': 19.173578515756606, 'angle': 1, 'pickedMesh': 'ground'}, {'distance': 19.182343652342773, 'angle': 2, 'pickedMesh': 'ground'}, {'distance': 19.196967063056853, 'angle': 3, 'pickedMesh': 'ground'}, {'distance': 19.21747108056488, 'angle': 4, 'pickedMesh': 'ground'}, {'distance':
```



```
19.24388708713425, 'angle': 5, 'pickedMesh': 'ground'}], 'position': {'y': 0.499998539686203,  
'x': 125.00000762939453, 'z': -108.17236328125}, 'title': 'move_response'}
```

39) “Wait for move” data example:

```
{'title': 'wait_for_move', 'content': 'Wait for opponent'}
```

40) “Wrong velocity type” data example:

```
{'title': 'wrong_velocity_type', 'content': 'Velocity values must be float or integers'}
```

41) “Game ended” data example:

```
{'content': 'Time is up. Rat won!', 'title': 'game_ended'}
```