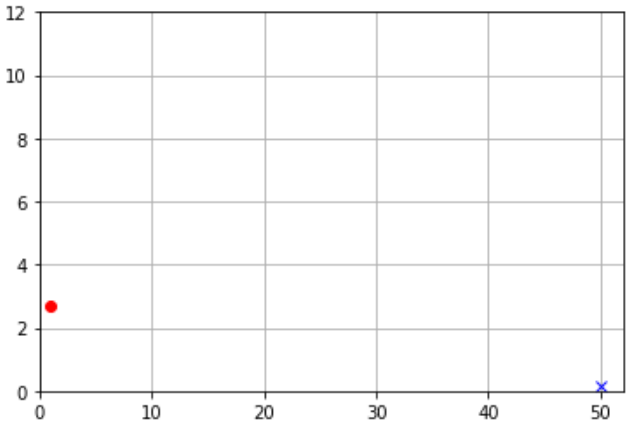
Environment v0

The bigger picture

- 0 – (realistic) toy environment. Gen random velocity profiles. Render
- 1 – Implement RL algo (DQN). Obtain optimal policy
- 2 – Gen expert trajectories. Recover Ped utility (reward fn) through IRL



(realistic) toy environment

Environment

- **Env** = Init + Reset + Step
- **Init** – Env dimension [dxW], Velocity and acceleration ranges, Safe-box dimensions (i.e. danger zone), Fear-box dimensions, time resolution
- **Reset** – Pedestrian starts at (d,0). Car starts at a random y-position, with random velocity and acceleration
- **Step** – C_step + P_step
  - C_step: random action, irrespective of pedestrian behavior
  - P_step: assign reward based on situation (goal / intent + fear / death)

Simulation

- **State space**: C_state, P_state
- **Action space**: V_ped
- **Reward**: goal = 5000, death = -10,000, intent = -time, fear = f (V_ped, C_pos - P_pos)
- **Initial Conditions**: d = 50, W = 9, V_ped = [0, 1.5], V_car = [0, 8], A_car = [-1.5, 1.5], Safe-box = 1.5, Fear-box = 2
- **Termination**: Death, Goal, Time > 200, C_pos > 1.5*d. Considered 1 episode
- **Result:** Number of episodes, and the fraction of times Ped crossed safely (1-death toll)
- **Render:** Animation of the env. Image on next slide

RL Algo - DQN

- 2-layer NN, with 24 ReLU activations. Linear layer output with Adam
- Action space – V_ped = {0, 0.75, 1.5}
- Result – Completely avoids death by walking at max speed. Need to modify env accordingly