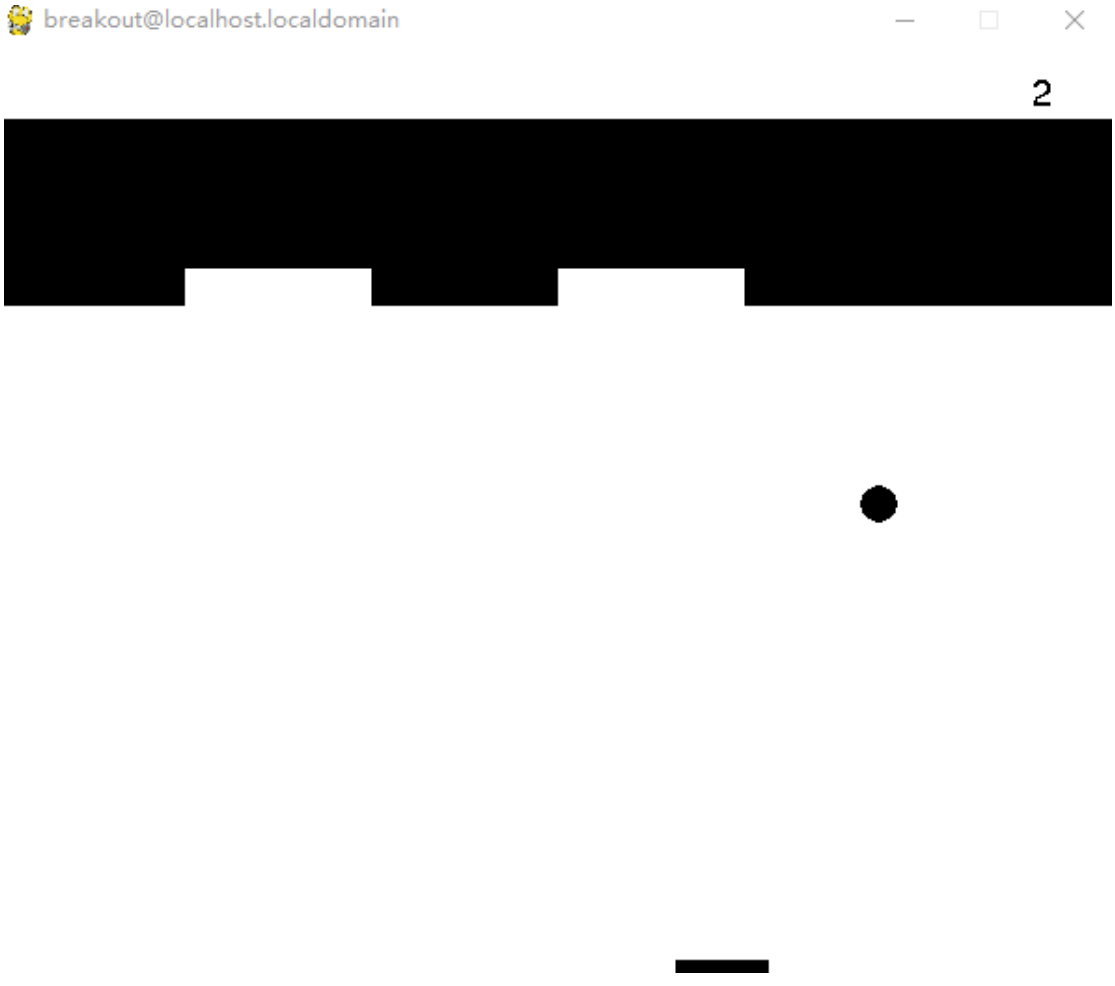
Breakout 实验进展：

游戏方面：

游戏界面：
我去掉了游戏背景、颜色、开始与“Game Over”的界面，最大程度上简化游戏的画面，减少干扰和降低计算量。

游戏流程：
游戏开始时，挡板在底部随机位置。小球打完所有砖块时游戏胜利，小球落到挡板下方时游戏结束。游戏胜利或结束后马上进入一轮新的游戏。

Reward 设置：
最开始设置的是：打掉一块砖块 reward 为 1，游戏结束 reward 为-1，但按照这个设置训练了一段时间之后发现，模型学习速度很慢，每次只能接一个球。所以我加大了游戏胜利的奖励和游戏结束的惩罚，把 reward 设置为：每一步的基础 reward 是 0.1，打掉一块砖块 reward 加 1，挡板每接到一次球 reward 加 1，游戏胜利 reward 加 10，游戏结束 reward 减 5。



模型方面：

基本参数：

训练步数：
训练前观察十万步，一共训练 3 百万步（先看训练 3 百万步的结果）。
ε-greedy 参数：初始 ε 为 1（即完全随机），十万步观察结束后，在接着的一百万步训练中递减至 0.1，而后保持不变。
经验池大小：50000
Minibatch 大小：32

```
GAME = 'breakout'  # the name of the game being played for log files
ACTIONS = 3  # number of valid actions
GAMMA = 0.99  # decay rate of past observations
OBSERVE = 100000.  # timesteps to observe before training
EXPLORE = 1000000.  # frames over which to anneal epsilon
TRAINING = 2000000  # timesteps to training
FINAL_EPSILON = 0.1  # final value of epsilon
INITIAL_EPSILON = 1  # starting value of epsilon
REPLAY_MEMORY = 50000  # number of previous transitions to remember
BATCH = 32  # size of minibatch
FRAME_PER_ACTION = 1
PROBABILITY = 0.5  # probability of human choose
```

人为干预部分：

比较挡板中心点横坐标和小球球心横坐标，得出人为的判断，然后以一定概率（最初用的概率是 0.5）直接采用人为的判断动作，否则采用模型的判断动作。

```
# choosing the human action with PROBABILITY
if random.random() <= PROBABILITY:
    print("----------Human Action----------")
    if ball_x < bat_mid:
        a_t = [1, 0, 0]  # move to left
    elif ball_x > bat_mid:
        a_t = [0, 0, 1]  # move to right
    else:
        a_t = [0, 1, 0]  # do nothing
elif random.random() <= epsilon:
    print("----------Random Action----------")
    action_index = random.randrange(ACTIONS)
    a_t[random.randrange(ACTIONS)] = 1
else:
    action_index = np.argmax(readout_t)
    a_t[action_index] = 1
```

除人为干预的部分外，加入人工干预的模型和不加的模型的其他参数都保持一样。

两个模型同时在服务器上运行，跑完 3 百万步大约需要 78 个小时（3.25 天）。现在还正在跑，结果图还没出来。

等这个结果出来后，我的想法是再多试试其他的几个参数，比如改变选择人为判断动作的概率、改变加入人为判断的时间等，再看看结果如何。除开这些，老师您认为实验接下来还可以怎么做呢？