**OBJECTS:**

Board:

* Holes
* Stores
* Marbles

Person:

Game:

* Quick play
  + Model (Reinforcement learning)
* Animated play

Scenarios:

* 1 marble
  + Hole
    - Empty
      * Yours

1. Get opponent’s marbles and 1 marble and put in store
2. END TURN
   * + - Not Yours
3. END TURN
   * + Not Empty
   * Store
     + Yours
     + Not Yours

* >1 marble

def move\_marbles(self,side\_i,hole\_i,n\_marbles):

if n\_marbles==1:

if hole\_i == 6:

if side\_i==self.player\_up:

self.scoreboard[self.player\_up].add\_marbles()

print("EXTRA TURN!!")

return None

else:

self.master.after(10, self.move\_marbles,flip(side\_i),0,n\_marbles)

return None

else:

n\_marbles += self.game\_board[side\_i][hole\_i].get\_marbles()

print('Marbles in hand {}'.format(n\_marbles))

if n\_marbles == 1:

if side\_i==self.player\_up:

opposite\_hole\_i = self.game\_board[side\_i][hole\_i].get\_opposite\_hole()

self.scoreboard[self.player\_up].add\_marbles(opposite\_hole\_i.get\_marbles() + 1)

self.turn\_over()

return None

else:

hole\_i += 1

if hole\_i == 6:

side\_i, hole\_i, n\_marbles = self.reached\_store\_flip\_sides(side\_i,n\_marbles)

else:

self.game\_board[side\_i][hole\_i].add\_marbles()

n\_marbles -= 1

hole\_i += 1

print('Marbles in hand {}'.format(n\_marbles))