Documentation of the code

Team WatchDogs

Devanshu Jain, Miten Shah, Bharat Kumar Garg

November 19, 2014

1 Introduction

We have made the following functions in the file ***SamplePlayer.cpp***:

- *createState()* // This creates the state for the agent looking at his current world model
- *writeState()* // This writes the state to the appropriate file of the agent, specified by '*state_[unum].txt*'
- *chooseAction()* // This chooses the appropriate action for the agent to do in current state
- *findQ()* // This finds the q-value of the state-action pair
- *obtainReward()* // This gives a reward to the agent looking at its previous and current state
- *SampleMove()* // This moves the agent according to the algorithm described in the term paper
- *SamplePass()* // This does a direct pass to the agent
- *GiveThrough()* // This does a through pass to the agent
- *RunThrough()* // This does a run through so that the nearest player to the ball runs through to possess the ball
- *impl_dribble()* // This does dribbling of the ball
- *getUnum()* // This returns the uniform number of the agent

2 Flow of the Program

The state is created as position of all playing entities and the ball. To narrow down the possibility of the number of states, we have divided the field in a grid of 18 X 24, with the origin at the center of the field.
The action space is described as an enumerator in *SamplePlayer.h*, with possible actions as: Pass, Hold, Dribble, Goal, Move and Intercept.
The actual flow of the program is illustrated in the figure 1.

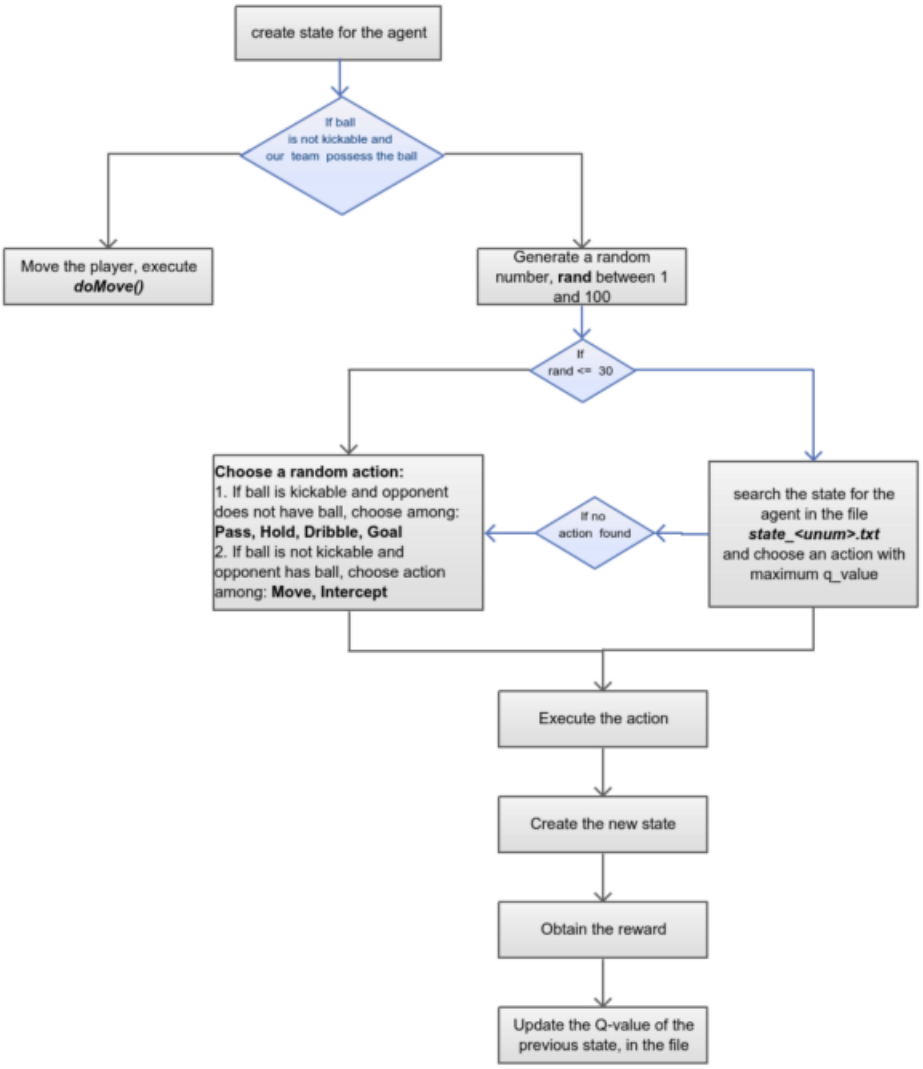

Figure 1: Flow of the program in *SamplePlayer.cpp*