Project Proposal

My project is aimed at creating a simple soccer game using Tkinter.

Game rules

Since football game is a really complicated game, with the limitation of time and technology, I plan to simplify the rules of my game: there are two teams in different color. Each team contains 4 field players and one goalkeeper. The objective of the game is to score as many as possible.

Game environment

The game environment has the following objects:

1. Players
2. Teams
3. Field
4. Ball

   I will define a class for each of them. I will treat field players and goalkeepers as the subclass of players.

   The whole game will be encapsulated by the class soccerField, which will be instantiated at the beginning of main function. This class will hold all the other objects including two teams, a ball and 10 players. I will also treat the game canvas as a 2d board just as what we do in snake and tetris. Each cell is a position. At the beginning of the game, each player will be at a certain cell, this cell will be there home. If they don't have other tasks, they will come back this cell. As the strategy of a team change, players home cell will also be changed. For example, when attacking, players home cell will be more close to the goal of

the opponent.

I will also have the ball class. It stores the ball's position standing by x and y axes, ball's velocity vx, and vy and a Boolean variable isCaught standing for if the ball is caught by player. It will also have a method to move the ball.

Player class is the most important class in the game. Since I just change my project to this one, I really don't have a clear organization of it. It will definitely have some basic attribute such as position, home cell, speed and so on. It will also have some state methods control its behavior including chasing the ball, kicking the ball, passing the ball and maybe also dribbling.

User's behaviour

User can use the keyboard to control one player in the team. There is a state variable in team class to indicate if the team has the ball and another variable indicates which player on the team are driving the ball. Actually, the value of the variable is the player's index in playerList, an attribute of team class.

When the control keys are called, we will call the method of this under-control player to act some behavior. Some of this methods are, move up, move down, move left, move right, pass the ball, shoot the ball…

User can use direction keys to move the current player. When user press this keys, I just modify the

When attacking, the player user used has the following move alternatives:

1. Drive the ball:

   My program will wait for direction keys stroke to modify the position of current player in a certain direction. If no key is pressed, no move will appear.

2. Pass the ball:

   When the direction key and the pass key 's' are both pressed, the ball will be kicked off the current player. The direction will from this player to the nearest

player in this direction, if there is no player in this direction, the ball will move forward until reaching the boundary. For example, if the player pass the ball by pressing passing key and UP and Right, the ball will go to the guy above and right to this player who have the nearest distance to him. If the receiver is moving, my program will based on receiver's current speed and direction and let the ball move to the front of his trajectory. That is to say, the direction of the ball will not accurate point to the receiver, it will point to the area a little ahead of the receiver. So the user should control the kicking force to make the pass accurately. If no direction key is pressed, the ball will go in the player's heading direction. I vary the force by considering the time passed when key is held down.

3. Shooting the ball

   Since in the 2d vision, it is really hard to simulate shooting. I make it in a simplified way. To make the game easy to play, I let the user use only one key to control shooting. That means, by pressing the shooting key 'd' with different strength, the user can control the ball's direction and speed. If no direction key is pressed, the ball will go strait to the middle of the goal. If a up or down key is pressed, the ball will move to the right and the left half of the goal. The magnitude of the offset and the ball's speed is proportional to the strength or the time that shooting key is kept down.

When defending, the under-control player also follows commands of directions keys. If the ball is moving without opponent protecting it, the player can catching it by moving forward it. If the ball is controlled under an opponent's foot, the player can only reget the ball by run into this player face-to-face.

AI's behavior

If an AI is in the defensive side, what he should do is run to the ball and try to make a

steal when ball get close to it. In my program, all the player will sit in their home cell at first. They are all in charge of a circle area, when the ball goes in, these AI chase the ball and try to catch it. Once they move out of the home cell, their circle area will also move with them. Only when the ball is out of their circle area they will come back to the home cell.

When attacking, AI can be separated into two groups: forward and guards. Forward will try to move in front of the ball and wait for a pass and make a shot. The guard will stay in the back field to protect goal from attacking in the future.

Other features

- Highlight replay

  When a team score, the program will go back to replay this process. To doing this, we should only know all the objects' position in the former moment and redraw every screen from now back. So each visible object should have a variable which is a list storing maybe 1000 position tuple.
- If I still have time in next two weeks, I will try to use socket to achieve multiple player play against each other using different PC through local area network.