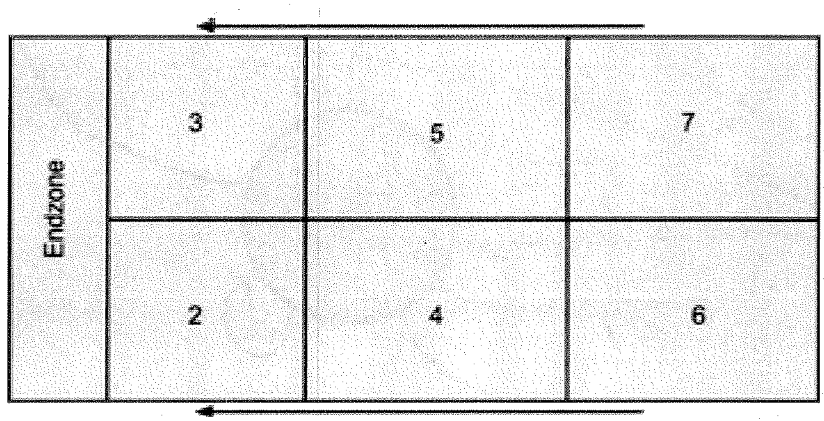
Josh Levin

Math 350 Final Project

Ultimate Frisbee

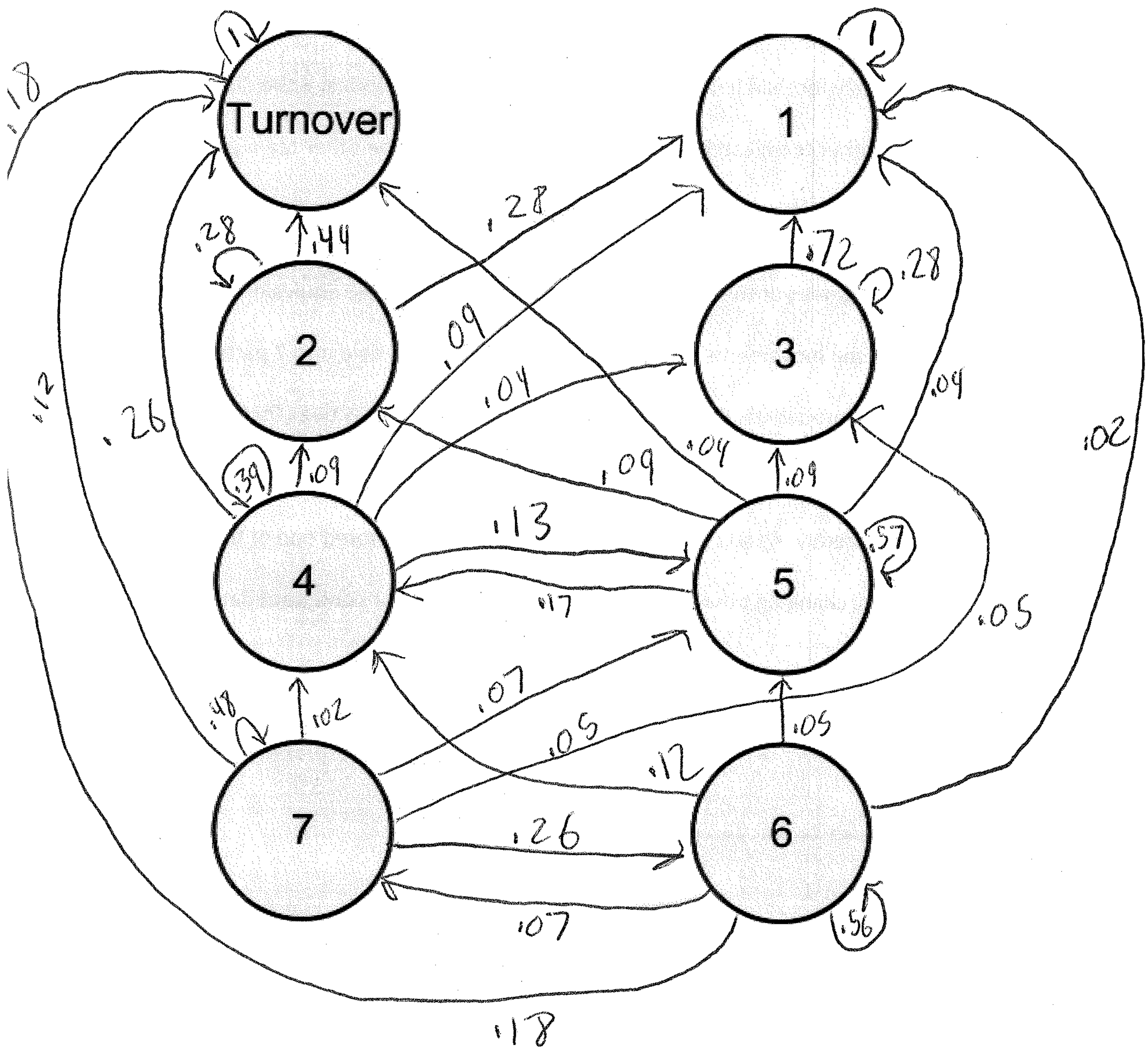
Brief background: The way the game is played is that the disc is thrown from person to person on an approximately football field sized field. There are two end zones. If the disc touches the ground or the other team catches it then it is a turnover and the other team gets to try to score. A team scores by catching the disc in the end zone.

Objective: I watched games of the highest level of ultimate frisbee (the equivalent of professional) and recorded the movements of the disc. By breaking up the field into 7 parts I was able to record data for a probability matrix. Ignoring the defensive perspective I focussed on the possibility of a player throwing the disc to another node or turning it over based on which node they were in currently. As you can see in the diagram, I have ignored one of the end zones because upon the disc changing possessions, I reoriented the field for consistent data.



Hypothesis: My hypothesis is that by testing the probability of a team scoring when they have the disc in a certain location on the field, I could then know where it is on the field that my team wants to have the disc.

Data: After collecting data on the teams I was able to create a transitions probability graph below:



From the graph I was able to create this **Probability Matrix**:

```
[1 0 0 0 0 0 0 0;
 0 1 0 0 0 0 0 0;
.44 .28 .28 0 0 0 0 0;
 0 .72 0 .28 0 0 0 0;
.26 .09 .09 .04 .39 .13 0 0;
.04 .04 .09 .09 .17 .57 0 0;
.18 .02 0 0 .12 .05 .56 .07;
.12 0 0 .05 .02 .07 .26 .48]
```

Analysis: I then plugged in this matrix to the function we created earlier in the semester for the cat and mouse problem in homework 6. Altering the code to give me results if the disc were to start in sections 2, 3, 4, 5, 6, and 7. Of course players can not start with a turnover that is the dead state, and state one is if the team scored. Below I have posted the results when I did the random walk 20,000 times.

Starting location:	7	6	5	4	3	2
Chance of Scoring:	. 3115	. 2590	. 5400	. 3879	1	.3894
Avg. # of throws:	4.8716	4.2772	4.0119	2.7822	1.3926	1.3943

What to take from this?

While the results are certainly limited, one of the trends that I noticed is that it was beneficial for teams to keep the disc on the right side of the field. It is possible that the data I drew from had a particular direction to the wind, but over all, teams the kept the disc on the right side of the field were far more successful than ones that moved the disc up the field on the left side. An easy way to see this is to look at the odds verse the even locations. Odds are on the right side of the field, where the evens were on the left.

What else I could do:

1. Due to limited data the probabilities are certainly not completely accurate. For instance, while no team turned it over when throwing from sector 3, it is certainly not a guarantee that if a team gets the disc that area that they will always score. I can always get more data.

2. The data is harvested from the best teams in the world so it may not apply to my team. Although video is rare, I could collect data on my team. Unfortunately, playing and recording are two exclusive actions.

3. I was unable to adjust for changes in defensive strategy. Not sure how I would do this, but it may require completely different probability matrices.

4. I did not adjust for the difference between when a player received the disc after playing defense, or if they just started on offense at the beginning of the point. Those are two very different circumstances that could be adjusted for.

5. There is certainly more that I could do if I was given more time.

Code:

```
function [a b] = ultimatef(j)
P=[1 0 0 0 0 0 0 0; 0 1 0 0 0 0 0 0; .44 .28 .28 0 0 0 0 0; 0 .72 0 .28 0
0 0 0; .26 .09 .09 .04 .39 .13 0 0; .04 .04 .09 .09 .17 .57 0 0; .18 .02 0
0 .12 .05 .56 .07; .12 0 0 .05 .02 .07 .26 .48];
numberOfThrows=0;
```

```
scorevector=zeros(1,j);

q=1;

w=1;

% Play game j times:

for r=1:j

   statechain=[];

%initial state vector:

svector=[0 0 1 0 0 0 0 0];

statechain(1)=3;

stepcounter=0;

f=1;


while statechain(f)~=1 & statechain(f)~=2

    %move rooms, update statechain


    for i=1:length(svector);

    AccumA(i)=sum(svector(1:i));

    end

    x=rand;

    nextstate=1;

    while AccumA(nextroom)<x

        nextstate=nextstate+1;

    end


   if nextstate==1

      numberOfThrows(q)=length(statechain)-1;

      q=q+1;
```

```
    end
    if nextstate==2
        numberOfThrows(q)=length(statechain)-1;
        scorevector(1,w)=1;
        w=w+1;
        q=q+1;
    end
     statechain(f+1)=nextstate;
     stepcounter=stepcounter+1;
     svector= P(nextroom,:);
     f=f+1;
end
b=sum(numberOfThrows)/length(numberOfThrows);% average number of steps
a=sum(scorevector)/length(scorevector);% frequency of mouse living
end
```