**3v3basketballParkaudience.bat：**

As to the Audience part, audiences have three behaviors: ‘Excited’, if the host team wins 3 scores consecutively in their 3 attacking rounds; ’Angry’, if the host team cannot get any score consecutively in their 10 attacking rounds; else, the behavior is ‘normal’. Some of them will decide to leave the park if they are angry. The total amount of audiences who still want to watch the game will be shown as an output. Another output is the audience’s behavior.

One of the input is the audience, that should be an integer. The other input is the current attacking round result which should be ‘0’ or ‘1’. Accordingly, one of the output is the current number of audiences who still want to watch the game ( they may leave if they are angry!) and the audience’s behavior(i.e., 0:angry, 1:excited, 2:normal).

**3v3basketballParkticketwindow.bat ：**

Ticket Window is the entrance to the park, through which audiences and workers can get into the park. Workers will come first to prepare for the game. After they finish preparing, audiences will be able to come in. There will be no fee for workers, bur each Audience should be charged 3 dollars as entrance fee. It will output the total amount of money earned as well as the current audience number.

Both inputs of the ticketwindow are supposed to a positive integer. Workers will come first to prepare for the game. Then, the audiences will come. Due to the limitate seats of the park, the total number of audiences should be less than 500. In this case, if the ‘audienceIn’ is larger than 500, after the 500th audience comes in, the ticketwindow will be passive. The ta() of workers is 1 second less than that of audiences because workers can come through the gate without paying , which is quicker.

**3v3basketballParkbasketballcourt.bat，3v3basketballParkbasketballcourt2.bat，3v3basketballParkbasketballcourt3.bat：**

The Basketball Court component will model how the game goes on. We are mainly focus on the host team’s performance. The host team will have a total attacking round initially. When their current attacking round equals the total attacking round, the game will be over. The host team includes 3 players (Guard, Forward, and Center). Assuming that the grand is the one who controls the pace of the game and its skill decides whether the team is able to find a suitable chance to shot before losing the possession of the ball(lose one attacking round). The forward will mainly focus on shot and its skill plays a dominate role in the ratio of successful shot. The Center, based on its rebound ability, decides whether its team is able to get another attack opportunity if the ball is not in(gain another attacking round). Also, the defend skill of the visit team can reduce the ratio of successful shot of the host team. Game is based on the 24 second rule, which means within 24 seconds, the possession of ball will change between the host team and visit team. That is to say the internal transaction time is 48 seconds for the Basketball Court (the host team is able to get a score during a time period of 48 seconds : the host team attack within 24 seconds, then the visiting team attack as well).With the game going on, the Basketball Court will output the score of the host team’s current attacking round and as well, when game is over, it will output the final score of the host team.

The only input of basketballcourt is a 6 or 7 bit number which contains many information.Take number n=5005350 as an example: n/10000 is the total round number; (the fourth number from the right \*20)/100

is the opportunity to lose one attacking round;(the third number from the right \*20-the first number from the right\*10)/100 is the ratio of successful shot;( the second number from the right\*4)/100 is the opportunity to gain another attacking round. The ta() is a random number from 0 second to 48 seconds.

Several *.ev* files will be set to test as follow:

Example 1: **3v3basketballParkbasketballcourt.bat**

00:00:00:00 totalRoundNumIn 5000350

Result:

……

04:23:41:271 scoreout 0

04:23:56:674 scoreout 0

04:24:33:441 scoreout 0

04:24:46:109 scoreout 0

**04:24:46:109 totalout 0**

This is expected because (the fourth number from the right \*20)/100=0, which means PG misses all attack opportunities.

Example 2: **3v3basketballParkbasketballcourt2.bat**

00:00:00:00 totalRoundNumIn 5005550

Result:

……

04:23:25:207 scoreout 1

04:23:41:271 scoreout 1

04:23:56:674 scoreout 1

04:24:33:441 scoreout 1

**04:24:33:441 totalout 500**

This is expected because: 1>(the fourth number from the right \*20)/100=100. The PG will never make a mistake in this case. 2>(the third number from the right \*20-the first number from the right\*10)/100=100.

The SF ‘s successful shot ratio is 100%.

Example 3: **3v3basketballParkbasketballcourt3.bat**

00:00:00:00 totalRoundNumIn 5004442

Result:

……

04:36:27:376 scoreout 0

04:36:35:049 scoreout 0

04:37:03:069 scoreout 0

04:37:45:071 scoreout 0

**04:37:45:071 totalout 259**

That is an expected result.

**3v3basketballParkscore.bat：**

The only input of Score is the totalScoreIn which is an integer.While receiving the input, it will output the result.

**3v3basketballParkcourt.bat：**

The input is the same as atomic model basketballcourt. The output is current attacking round result and total score.

**3v3basketballParkbasketballpark.bat，3v3basketballParkbasketballpark2.bat，3v3basketballParkbasketballpark3.bat：**

out : audienceOut(output current audience number)

moneyOut(output total money)

out (output audience behavior)

totalScoreOut(output final score)

in : totalRoundNumIn (total round, PG,SF,C,Defence’ value)

audienceIn(audience number)

workerIn(worker number)

**3v3basketballParkbasketballpark.bat:**

Host team will win all attacking rounds ( their players’ value is the full value),audiences are excited.

**3v3basketballParkbasketballpark2.bat:**

Host team will lose all attacking rounds ( one of their player’s value is 0),audiences are angry and leave.

**3v3basketballParkbasketballpark3.bat:**

Host team is a good one( their players’ value is more than average).