2012 China Underwater Robot Games（Nan Jing）2D Simulation Group Rules

V 2.0 Revised 20120101

Intelligent Control Lab of PKU

Jan. 2012

Index

1. Game Platform

1.1. Platform Introduction

The software URWPGSim2D (Underwater Robot Water Polo Game Simulator 2D Edition) is used as the game platform in 2D group of 2012 China Underwater Robot Games. The simulators include URWPGSim2DServer and URWPGSim2DClient. The server simulates underwater game environment, controls and displays the process and result of the game and sends real-time game environment and process information. The semi-distributed client simulates game teams, loads game strategy, processes decision computation and sends decision results to the server.

1.2. Hardware Environment

The platform runs on a PC or workstation with hardware configuration as the following table.

Components	Minimum Configuration	Recommended Configuration
CPU	Intel P4 2.0GHz or similar AMD CPU	Intel E7300 2.66GHz or higher
Memory	256MB	2GB or more
Video Card	Support DirectX 9.0，Pixel Shader 3.0，video memory 128M or more	
Harddisk	10GB	80GB or more

1.3. Software Environment

OS：Windows XP Professional SP3，Windows Vista OR Windows 7.

.Net Framework：.Net Framework 3.5 with SP1.

MRDS：Microsoft Robotics Developer Studio 2008 R3.

Accessory：Microsoft XNA Framework Redistributable 3.1，Microsoft Excel 2003 Com Library.

1.4. Installation and Running

1.4.1. Installing Standard Platform

1. Install Windows XP Professional SP3 on a PC or workstation.
2. Install DotNet3.5SP1 and XNA3.1 with the default setting.
3. Install URWPGSim2D software package, which includes necessary MRDS and Excel component.

1.4.2. Selecting running mode

URWPGSim2D provides two running modes: Local mode and Remote mode.

Local mode is provided to all teams for debugging game strategy roughly. In Local mode, only one process URWPGSim2DServer.exe needs to be launched on the server. The strategy module (dll file) is loaded on the server directly and the computing process of all strategies and simulation cycle process on the server share the same process space.

Remote mode is used for actual games. In remote mode, one process needs to be launched on the server and several processes on the clients. The number of clients is the same as that of teams which participate in current game, i.e., in a 1vs1 game, the number of teams is 2 and 2 client processes need to be launched. The client processes can be launched on several computers or one computer or the server. Whatever topology structure is used, the decision computing process of each team runs in different and independent client process space.

Due to the difference of data exchange methods between Local mode and Remote mode, the strategy which works well in Local mode will not always have expected effect in Remote mode. Thus, all teams must debug their strategies in Local mode preliminarily and then test and modify those in Remote mode.

1.5. Standard Game Configuration

1.5.1. Simulation Field

The 2D model definition, structure and size of simulation field are the same as those of real pool. As shown in figure 1-1, the water surface area is the simulation field with one **goal** on each side. There are two **penalty mark** in the left and right penalty areas. The geometric center of the field is **kick-off spot** and the geometric centers of up-left, up-right, down-left and down-right 1/4 field are **faceoff spots**. Two dotted lines are **offending line** and **defending line**.

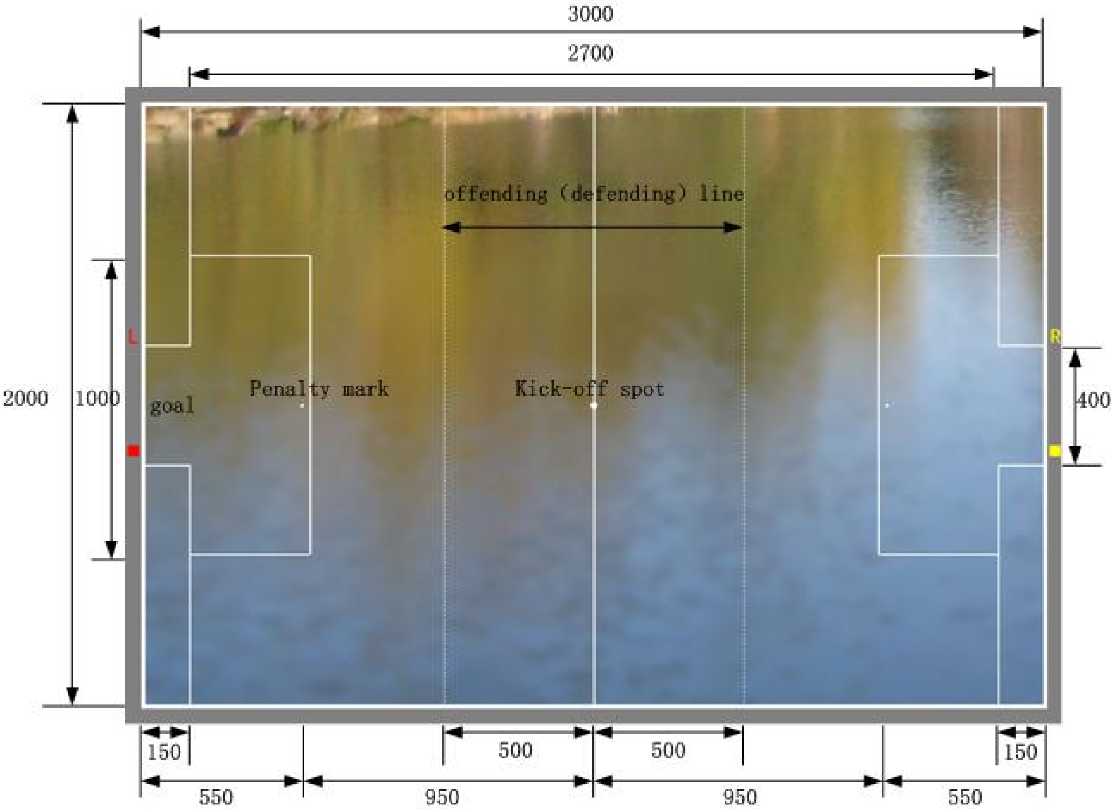

Figure 1-1 Simulation field

1.5.1.1. Size

1. Field: 3000mm×2000mm.
2. Goal: 150mm×400mm.
3. Penalty Area: 400mm×1000mm.

1.5.1.2. Coordinate System

1. The origin is the geometric center of the field. The positive x-axis points to the right while positive z-axis downward.
2. The angle from positive x-axis to negative x-axis varies from 0 to π clockwise and from 0 to – π anti-clockwise.
3. Z coordinates of left and right penalty mark are both 0 while x coordinates is 10 pixels away from the penalty area line. The real size will be different according to different screen resolution and 50mm in standard situation.

1.5.2. Simulation Robotic Fish

The 2D model definition, structure and size of simulation robotic fish are almost the same as those of real robotic fish..

1. Structure: 1 arc-shaped head, 1 rectangular body, 1 tail composed of 3 connected isosceles, 1 crescent caudal fin and 2 fan pectoral fins.
2. Size: radius of head arc is 30mm; the body rectangle is 150mm long and 44mm wide; the straight length of 3 tail joints are70mm, 54mm and 54mm, and the upper lines are 18.59mm, 12.49mm and 6.246mm long; the upper line of caudal fin is 4.858mm long with the bottom line 105mm long (between 2 poles of caudal fin) and 64.8mm high (between the midpoints of upper line an bottom line); the straight line of pectoral fin is 75mm long while the arc is 45mm wide and 50mm long at most.
3. Color: default color of fish body is red and the color of simulation robotic fish in one team is identical; the color of number on the body is black; in antagonistic game, default color of robotic fish in opponent team is yellow.

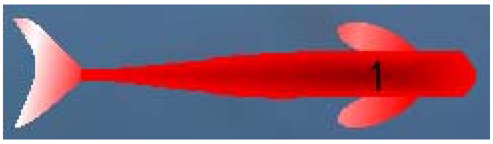

Figure 1-2 Simulation Robotic Fish

1.5.3. Simulation Water Polo

2D model of simulation water polo is a round with 58mm-long radius.

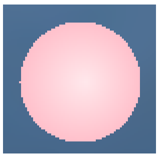

Figure 1-3 Simulation Water Polo

1.5.4. Simulation Robotic Fish Team

The number of team and simulation robotic fish in different game will be defined in specific game rules.

1.5.5. Simulation Game Timing

All 2D simulation games are under countdown system. Total game time can be 10 minutes, 5 minutes or 3 minutes according to specific game rules

The countdown timer on the server of game platform displays time left for current game with 1 second as the decreasing unit. The details are as the following. Total time of the game (e.g. 10 minutes) is divided by simulation cycle period set by the game platform (e.g. 100 milliseconds) and converted to number of simulation cycles gfVFS v V (10×60×1000/100=6000). The game platform will set the number of cycles left with initial value as number of total simulation cycles. As time goes, the cycles left are converted and rounded to the time left which is shown in the form of “mm:ss”.

1.5.6. Simulation Game Score

The scoreboard on the server of game platform shows real-time score of each team in current game with 1 point as increasing unit. By the end of the game, the score shown on the scoreboard will be the final score of each team.

2. Game Organization

2.1. Staff

2.1.1. Team and Members

The member number of each team is not limited with one member as the captain. The team is responsible:

1. to program game strategy according to URWPGSim2D platform criterion;
2. to appoint one member to load game strategy on assigned client and send the instruction “Ready”;
3. to change game strategy during the game in proper time according to the need and resend the instruction “Ready”.

2.1.2. Referee

There is one referee for each game, who is responsible:

1. to operate on the game platform: to launch the programs on the server and corresponding clients as the preparation of the game;
2. to coordinate all teams: to tell all teams to load game strategy on their client and send the instruction “Ready” and decide the proper time for the team that applies for the change of game strategy to make the change;

3. to control the game process: to send the instructions “Pause”, “Continue” and “Replay” during the game according to the need of specific game and handle various situations of the game.

4. to record the game result: to fill in the score sheet with the game result after each game and request the representative of corresponding teams to sign for confirmation.

5. to communicate and settle the game dispute: to settle the game dispute between the teams and report to the committee when the settlement fails or the dispute is between the referee and any team.

2.1.3. Facility Maintainer

During the game, there will be at least one facility maintainer responsible to keep all facilities running regularly.

2.2. Facility Configuration

2.2.1. Hardware

After registration deadline, the committee will inform the sponsor of the number of needed facilities and the sponsor is responsible for the preparation.

Each set of facility include recommended PC or workstation in section 1.2 (at least 1, at most 3), one set of projector and curtain, one 4-access switch, four ultra-five UTP with RJ45 connectors and several electrical outlets.

Standard configuration of one hardware set includes 2 recommended PCs or workstations in section 1.2 and one set of projector and curtain.

2012 CURG 2D simulation group needs 4 sets of standard hardware facility.

2.2.2. Software

The committee is responsible for the installation and test of related software according to the steps in section 1.4.

2.3. Game Flow

The simulation game flow is shown in figure 2-1.

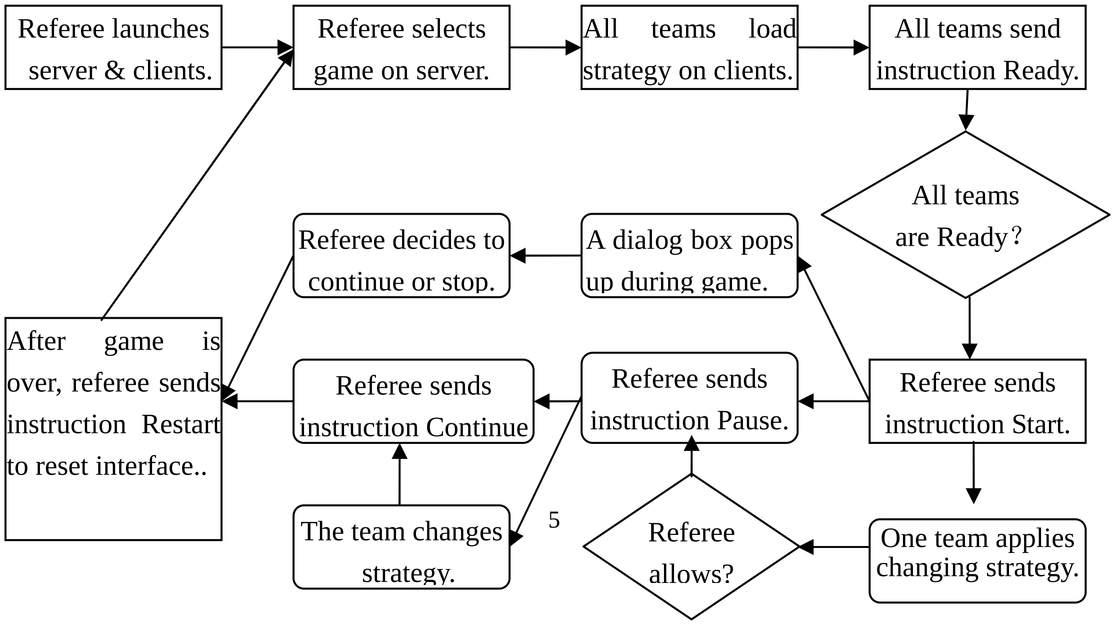
Referee launches server & clients.
Referee selects game on server.
All teams load strategy on clients.
All teams send instruction Ready.
All teams are Ready？
A dialog box pops up during game.
Referee decides to continue or stop.
After game is over, referee sends instruction Restart to reset interface..
Referee sends instruction Continue
Referee sends instruction Pause.
Referee sends instruction Start.
The team changes strategy.
Referee allows?
One team applies changing strategy.

Figure 2-1 Simulation game flow chart

Special Note: time-out for changing strategy is no more than 2 minutes.

3. Game Events

3.1. 2D simulation water polo (5vs5)

3.1.1. Match Content

2D simulation water polo (5vs5) is an antagonistic match between 2 teams with 1 simulation water polo and 5 simulation robotic fish for each team.

3.1.1.1. Match Field

The size of standard simulation field is 3000mm×2000mm. One and a half times of that size (4500mm×3000mm) is used by the game event of 2D simulation water polo as shown in figure 3-1 with ten simulation robotic fish.

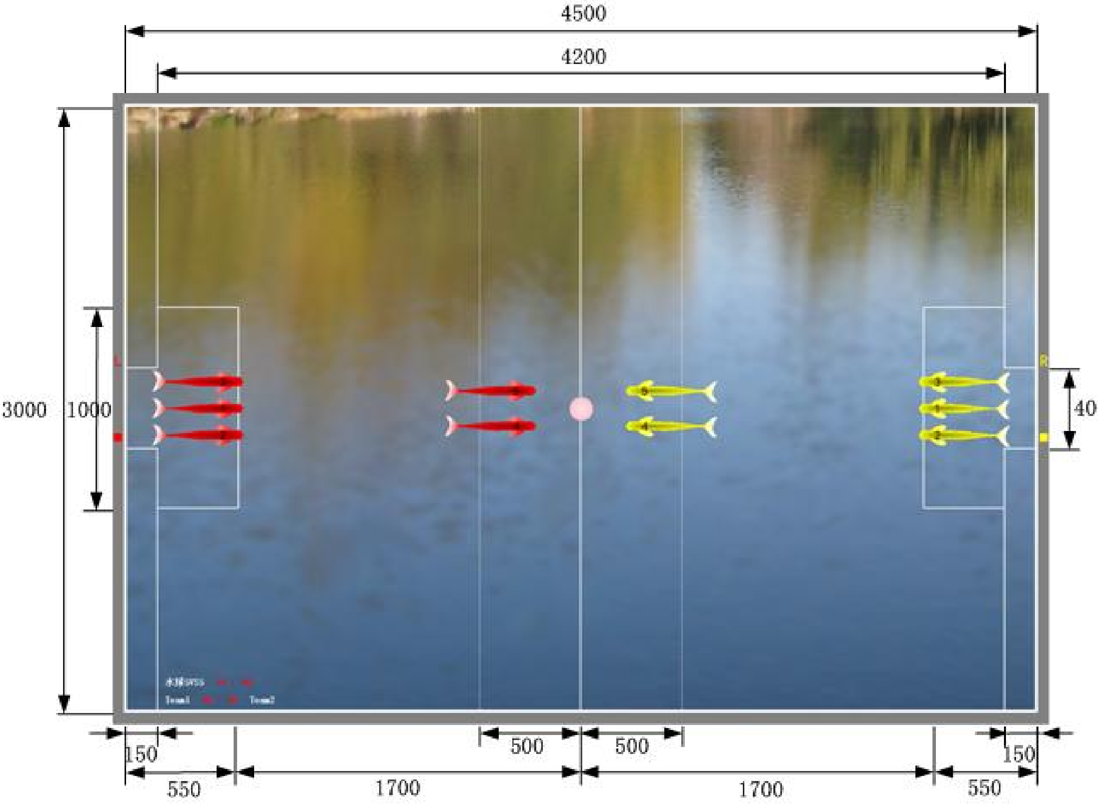

Figure 3-1 2D Simulation water polo (5vs5)

3.1.1.2. Match Process

1. In the initial state, the simulate robotic fish of each team are located symmetrically in the left and right side of the field as shown in figure 3-1 and the simulate water polo is stationary on the center mark.

2. To each team, the dash line in the front court is defined as **offensive line**, while the dash line in the back court as **defensive line**. Two lines are **500 mm** away from the center line symmetrically.

3. **The role distribution of simulate robotic fish:** No.**1** simulate robotic fish of each team is a free team member and can swim in the whole range of the field. No. **2** and **3** simulate robotic fish are defenders that can not pass the **offensive line.** No. **4** and **5** simulate robotic fish are offenders that can not pass the **defensive line.** Otherwise, there is a foul.

4. After the match starts, simulation robotic fish of both teams tackle dribble, shoot, block the ball and make goalkeeping in the drive of strategy with the object of push simulation water polo balls into the goal of the opponent team.

5. When any fish fouls, platform server will pop up a dialogue box and send instruction "Pause" automatically to pause the match. After the referee confirms, the system will give corresponding punishment. The referee sends instruction "Continue" to resume the match.

6. If one team goals, the server will pause the match and pop up a dialog box. After the referee confirms, the score of the team will be added 1 point and all the fish and the water polo reset to the initial state. The referee presses the button "continue" to resume the match.

7. After first half, the server will pause the match and pop up a dialog box. After the referee confirms, the teams will change ends. The referee presses the button "continue" to resume the match.

8. When the match is over, the server will pause the match and pop up a dialog box. After the referee confirms and presses the button "restart", the server will initial and prepare for a new match.

3.1.1.3. Change Strategy

1. After a goal or during the half interval, the teams can change their match strategy.

2. In addition, each team can apply for changing the strategy once per half. The referee will pause the match manually. After the change, the referee will resume the match.

3.1.1.4. Fouls and Punishment

1. The offender which passes the defensive line will be taken out of the field for 10 seconds. The defender which passes the offensive line will be taken out of the field for 10 seconds. The fish which fouls will be put upside the center line of the field.

2. When the whole of the water polo enters the penalty area and 3 fish of the defending team stay in the penalty area for more than 5 seconds, the last fish which enters the penalty area will be taken out of the field for 10 seconds. The fish will be put upside the center line of the field.

3. When the whole of the water polo enters the penalty area and 1 fish of the defending team passes the goal line and stays for more than 5 seconds, the fish will be taken out of the field for 10 seconds. The fish will be put upside the center line of the field.

3.1.1.5. Dealing with Draw

If the match is drawn in the regular match time, the server pops up a dialog box. After the referee confirms, there will be a 5-minute period of golden goal. A goal will end the match at any time immediately. The team which goals wins. If there is no goal in the extra time, the match is drawn. If there must be a winner of the match, the teams can rematch.

3.1.1.6. Dead Ball

If there are at least 1 fish of each team near the water polo which position changes little and the situation lasts more than.10 seconds, the server will consider it as dead ball.

There will be a scrimmage when dead ball occurs.

3.1.1.7. Scrimmage

The water polo will be put at the nearest scrimmage mark of the position where dead ball occurs.

The first fish of the defending team will be put at 400 mm away from the scrimmage mark near its goal line, while the first fish of the offending team will be put at 400 mm away from the scrimmage mark on the other side. The fish both face the scrimmage mark. The other fish are put parallelly with the first fish of each team.

3.1.2. Match Time

Total match time is 10 minutes with each half 5 minutes. At halftime, there can be a time-out no longer than 5 minutes if applied by any team or decided by the referee.

3.1.3. Scoring Rules

1. Winner determination: the team wins the match which scores the great number of goals during match time (10 minutes). If both teams score an equal number of goals, or if no goals are scored, there will be an extra time.

2. Goal and score: The team adds 1 score when simulation water polo enters the goal of the opposite side.

3. The score and winner of the match will be recorded.

3.1.4. Strategy Program

In the 2D water polo simulation match, the strategy parameters provided by the platform include only the flags of the period in the match. The flag is a 32-bit integer. The value 0 represents regular match period, while 3 represents golden-goal period. Other values do not make sense.

The flag can be get and used in the following way.

```
int matchPeriod = Convert.ToInt32(mission.HtMissionVariables["CompetitionPeriod"]);

if (matchPeriod == 0) //it is regular match period now

if (matchPeriod == 3) //it is golden-goal period now
```

3.2. 2D simulation tackling battle

3.2.1. Match Content

2D simulation tackling battle is an antagonistic match between 2 teams with 2 simulation robotic fish for each team, 9 simulation water polo balls and 4 rectangular simulation obstacles.

3.2.1.1. Match Field

Standard simulation field is used by 2D simulation tackling battle as shown in figure 3-2 with simulation robotic fish and other necessary elements.

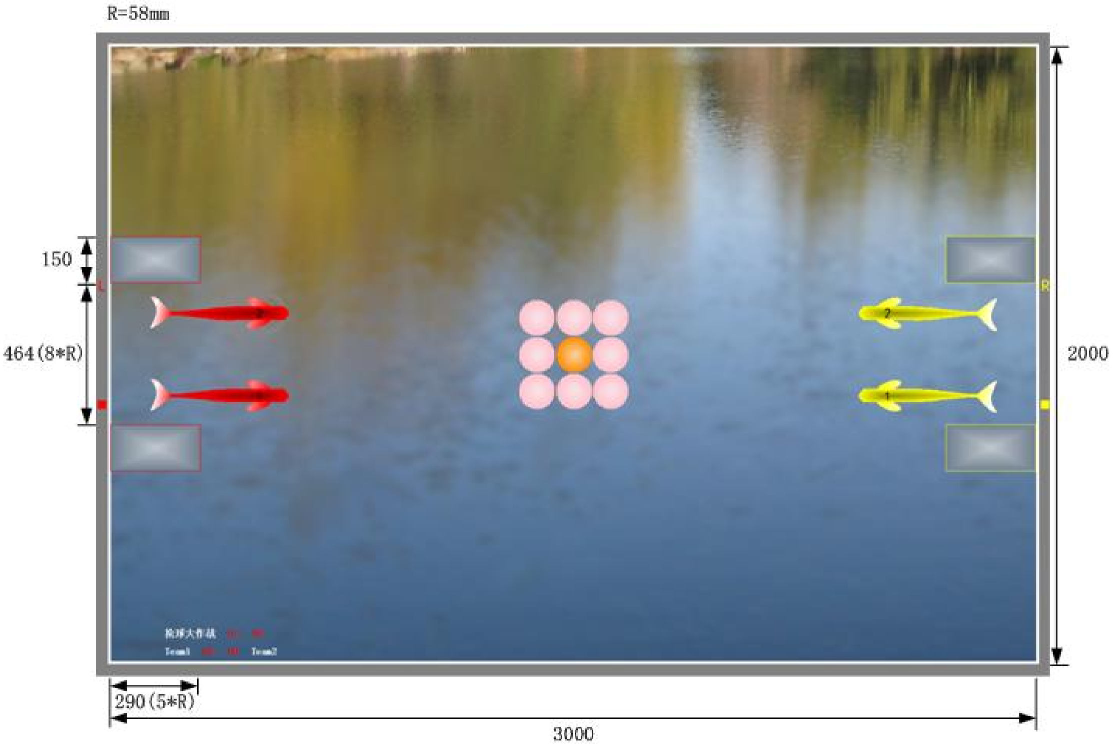

Figure 3-2 2D simulation tackling battle

3.2.1.2. Match Process

1. In initial state, 2 simulation robotic fish of each team are put symmetrically in their penalty area of the field, when 1 water polo ball is put on center kickoff mark with the other 8 balls surrounding next to it symmetrically.

2. The area enclosed by 2 gray obstacles on the left (right) side and left (right) boundary of the field is the object goal of the left (right) team, which is called left (right) goal.

3. After the match starts, simulation robotic fish of both teams tackle dribble, shoot, block and goal keep in the drive of strategy with the object of push simulation water polo balls into their own object goal.

4. At half-time, the platform server will pop up a dialogue box and send instruction “Pause” automatically to make the match pause. After the referee confirms, the system will change ends for 2 teams with all simulation water polo balls rolling over around the center mark o f the field. The referee sends instruction “Continue” to resume the match.

5. When match time decreases to zero, platform server will pop up a dialogue box. After the referee confirms the result, the match is over.

3.2.1.3. Changing Strategy

1. When the match pauses during halftime, both teams can change match strategy.
2. In addition, each team can apply for changing the strategy once per half. The match will pause by the referee manually. After the strategy is changed, the referee will resume the match.

3.2.1.4. Dealing with draw

If the match is drawn in the regular match time, platform server will pop up a dialog box. After the referee confirms, there will be a 5-minute stage of golden goal. A goal will end the match at any time immediately. The team which scores wins the match. If there is no goal in the extra time, the match is drawn. If the match must have one winner, one more extra time is needed. If there is no goal in the first half of golden-goal stage, 2 teams should change ends.

3.2.2. Match Time

Total match time is 10 minutes with each half 5 minutes. At halftime, there can be a time-out no longer than 5 minutes if applied by any team or decided by the referee.

3.2.3. Scoring Rule

1. Score of water polo balls: Score of the orange ball in the center is 3 while the other pink balls are 1.
2. Winner determination: the team wins the match which scores the great number of goals during match time (10 minutes). If both teams score an equal number of goals, or if no goals are scored, the match is a draw.
3. Goal and score: Left (right) team adds corresponding score when simulation water polo enters left (right) goal.
4. Score deduction：Left (right) team loses corresponding score when simulation water polo is get out of left (right) goal.
5. The score and winner of the match will be recorded.

3.2.4. Strategy Program

In the 2D simulation tackling battle, the strategy parameters provided by the platform include only the flags of the period in the match. The flag is a 32-bit integer. The value 0 represents regular match period, while 3 represents golden-goal period. Other values do not make sense.

The flag can be get and used in the following way.

```
int matchPeriod = Convert.ToInt32(mission.HtMissionVariables["CompetitionPeriod"]);

if (matchPeriod == 0) //it is regular match period now

if (matchPeriod == 3) //it is golden-goal period now
```

Id of the orange ball is 4, which can be got by the following parameter in match strategy:

mission.EnvRef.Balls[4]

3.3. 2D simulation water polo snooker

3.3.1. Match Content

2D simulation water polo snooker is a nonantagonistic match played by 1 team with 1 simulation robotic fish, 10 simulation water polo balls and 4 rectangular simulation obstacles.

3.3.1.1. Match Field

The size of standard simulation field is 3000mm×2000mm. One and a half times of that size (4500mm×3000mm) is used by the game event of 2D simulation water polo snooker as shown in figure 3-3 with one simulation robotic fish and other necessary elements.

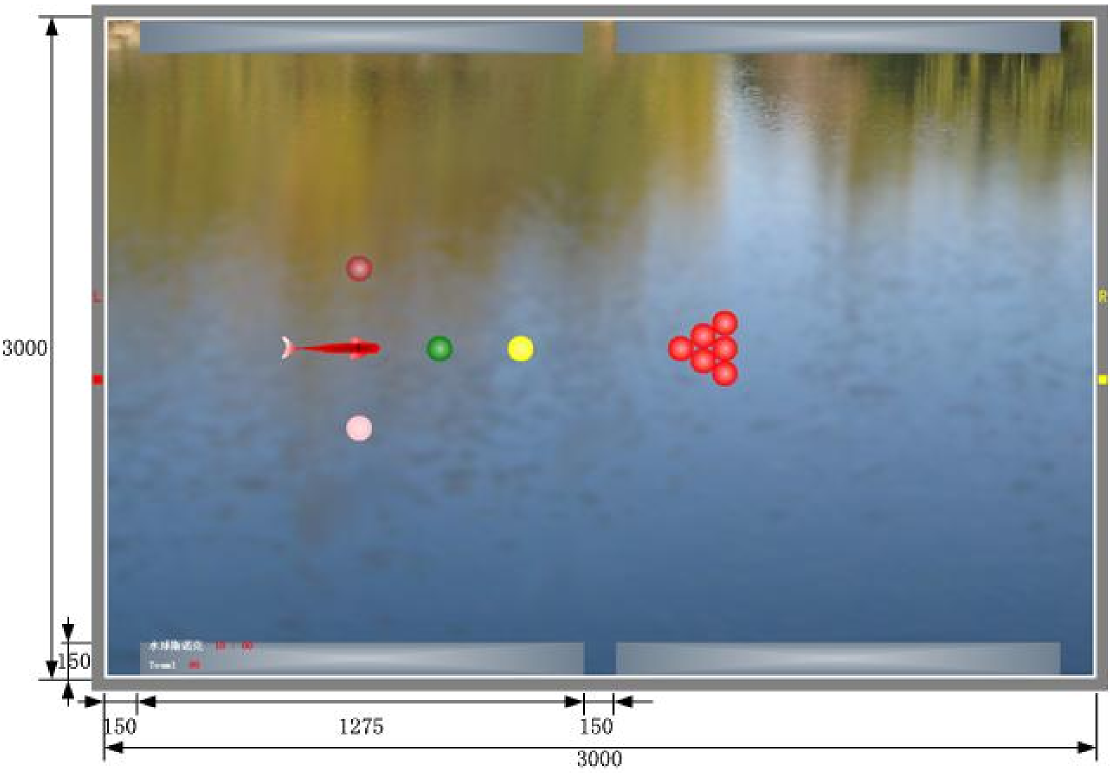

Figure 3-3 2D simulation water polo snooker

There are 2 rectangular obstacles next to both top and bottom boundary and 6 pockets are enclosed. The length of obstacles in Z direction, the length of pockets in X direction and the depth of the pocket are all 150mm. The length of obstacles in X direction and the coordinates of center mark can be got from computation.

Among 10 simulation water polo balls, 6 are red and the other 4 are colorful. Ball IDs are from 0 to 9, among which 0 to 5 are red and 6 to 9 are yellow, green, brown and pink successively.

In the initial state, red ball 0 and yellow ball 6 are put symmetrically on X axis on the left and right side of Z axis and the distance away from the origin equals to one straight fish length. The other 5 red balls are put in a triangle with ball 0 as a vertex. The ball 0 is the first column, while there are 2 and 3 balls in the second and third column. The red balls in each column are put together symmetrically about X axis. Ball 7 (green) is put on the X axis on the left of ball 6 (yellow) with the distance of one straight fish length. Ball 8 (brown) and 9 (pink) are put on the left of ball 7 (green) with the distance of one straight fish length on both X and Z axis. Ball 8 is on top and 9 on bottom.

In the initial state, the simulation robotic fish is on the X axis and oriented to the right with the same X coordinate as ball 8 and 9.

3.3.1.2. Match process

1. Before the match, the simulation robotic fish and balls hold the initial position and orientation.
2. After the match starts, the simulation robotic fish will push the ball into the pocket in the drive of strategy with the object of getting maximal score within match time.
3. When the team gets a full mark before match time runs out, i.e. all balls are pushed into the pockets, platform server will pop up a dialogue box. After the referee confirms the score and time left, the match is over.
4. When match time runs out, platform server will pop up a dialogue box and the referee confirms the score and time left when last ball is pushed into the hole.

3.3.1.3. Match Rules

1. First stage: After the match starts, one red ball shall be pushed into the pocket first, and then one colorful ball. The colorful ball that is pushed into the pocket will be put back at the initial position automatically. The other ball occupying that position will be pushed away. The red ball that is pushed into the pocket will be put out of the field from left to right on top or bottom.
2. Second stage: After all red balls are pushed into the pockets, the colorful balls should be pushed into the pockets according to the sequence of yellow, green, brown and pink. The colorful ball that is pushed into the pocket will be put out of the field from right to left on top or bottom.
3. It is valid to push the ball into the pocket according to aforementioned rule. Lest, it is invalid. When a red ball is pushed into the pocked invalidly, a dialogue box will pop up. After the referee confirms, the red ball will be put back inside the left or right boundary of the field. When a colorful ball is pushed into the pocket invalidly, there will not be any indication. The colorful ball that is pushed into the pocket will be put back at the initial position automatically. The other ball occupying that position will be pushed away.

3.3.2. Match Time

Total match time is 10 minutes and the match can be play only once. During the match, strategy can be changed one time.

3.3.3. Scoring Rules

1. The score of red ball is 1, while yellow, green, brown and pink ball is 2, 3, 4 and 5 successively.

2. When a red ball is pushed into the pocket validly, 1 point will be scored. When a colorful ball is pushed into the pocked validly, corresponding points will be scored. If any ball is pushed into the pocket invalidly, no point will be scored.
3. When all balls are pushed into the pockets, the match is over. The score and time left will be recorded.
4. When time runs out, the match is over. The score and time left when last ball is pushed into the pocket validly will be recorded.

3.3.4. Strategy Program

The only special parameter transmitting to strategy in 2D simulation snooker is the flag whether 10 simulation balls are still in the field. The flag is represented by a 32-bit integer, among which lower 10 bits from 0 to 9 are used to show if simulation balls are still in the field, 1 for in an 0 for not.

The flag can be got by strategy in the following way.

```
int ballsInFieldFlag = Convert.ToInt32(mission.HtMissionVariables["BallsInFieldFlag"]);

if (((ballsInFieldFlag << (31 - i)) >> 31) == 1) //i-th simulation ball being in the field

if (((ballsInFieldFlag << (31 - i)) >> 31) == 0) //i-th simulation ball not in the field
```

3.4. 2D simulation dribbling relay

3.4.1. Match Content

2D simulation dribbling relay is a nonantagonistic match played by 1 team with 2 simulation robotic fish and 1 simulation water polo.

3.4.1.1. Match field

Standard simulation field is used by 2D simulation dribbling relay as shown in figure 3-4 with simulation robotic fish and other necessary elements.

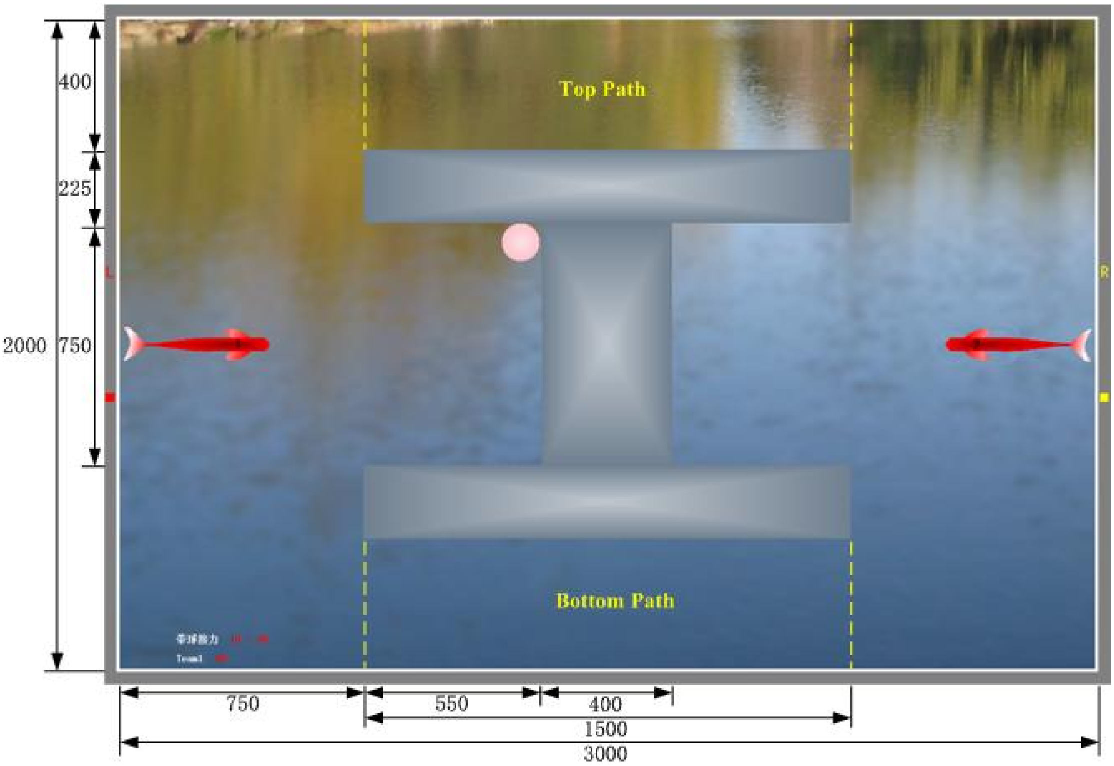

Figure 3-4 2D simulation dribbling relay

3.4.1.2. Match Process

1. In the initial state, 2 simulation robotic fish of 1 team are put symmetrically at the left and right side of the field as shown in figure 3-4.

2. When the match starts, simulation robotic fish will complete the mission of dribbling relay under the rules.

3. When 2 simulation robotic fish complete the mission before match time runs out, platform server will pop up a dialogue box. After the referee confirms time left, the match is over.

4. When match time runs out, platform server will pop up a dialogue box. After the referee confirms whether the mission is completed and time left, the match is over.

3.4.1.3. Match Rules

1. Definition: Member 1/2, the simulation robotic fish at the left/right goal in the initial state;

top path, rectangular area on top above I-shaped obstacle; **bottom path**, rectangular area on bottom below I-shaped obstacle; **stage I**, simulation ball is pushed from initial position to the right groove of I-shaped obstacle through top path; **stage II**, simulation ball is pushed from the right groove of I-shaped obstacle to the left through bottom path; **stage III**, simulation ball is pushed from the left groove to the right through top path; **stage IV**, simulation ball is pushed from the left groove to the right through bottom path.

2. Only member 1 can pass top path, while only member 2 can pass bottom path. If fouled, the robotic fish will be put at its initial position.
3. **Stage I and III must be completed by member 1 and water polo can only pass top path. Stage II and IV must be completed by member 2 and water polo can only pass bottom path.**
4. When simulation ball enters the groove completely, one stage ends and next stage starts. When all 4 stages are completed, the match is over.

3.4.1.4. Foul Handling

1. If simulation robotic fish enters wrong path, it will be put at its initial position.
2. If simulation ball enters wrong path, it will be put at its initial position of different stage in the groove of I-shaped obstacle.

3.4.2. Match Time

Match time is 5 minutes and the match will be played twice. No timeout is allowed during each time of match. The strategy can be the same or different in two times.

3.4.3. Scoring Rules

1. For each completed stage, 1 point is scored.
2. If all stages are completed in one time of match, the score (definitely 4) and time left will be recorded.
3. If time runs out and the mission is not completed in one time of match, the score and time left when last point is scored will be recorded.
4. The score and time left in 2 times of match will be added as total score and time left.

3.4.4. Strategy Program

The only special parameter transmitting to strategy in 2D simulation dribbling relay is the flag representing number of completed stages. The flag is represented by a 32-bit integer, the value of which represents that which stage is not completed. The values of i can be 0, 1, 2 and 3 while other values make no sense. Because the match is over when stage IV is completed, the value of i will not be 4 in the strategy.

The strategy can get and use the flag in the following way.

```
int fstc = Convert.ToInt32(mission.HtMissionVariables["FinishedSingleTripCount"]);

if (fstc == 0) //stage I is not completed
if (fstc == 1) //stage I is completed while stage II is not

if (fstc == 2) //stage II is completed while stage III is not

if (fstc == 3) //stage III is completed while stage IV is not
```

3.5. 2D simulation passing holes in coordination

3.5.1. Match Content

2D simulation passing holes in coordination is a nonantagonistic match played by 1 team with 2 simulation robotic fish, 1 simulation water polo and 3 rectangular simulation obstacles.

3.5.1.1. Match Field

Standard simulation field is used by 2D simulation passing holes in coordination as shown in figure 3-5 with simulation robotic fish and other necessary elements.

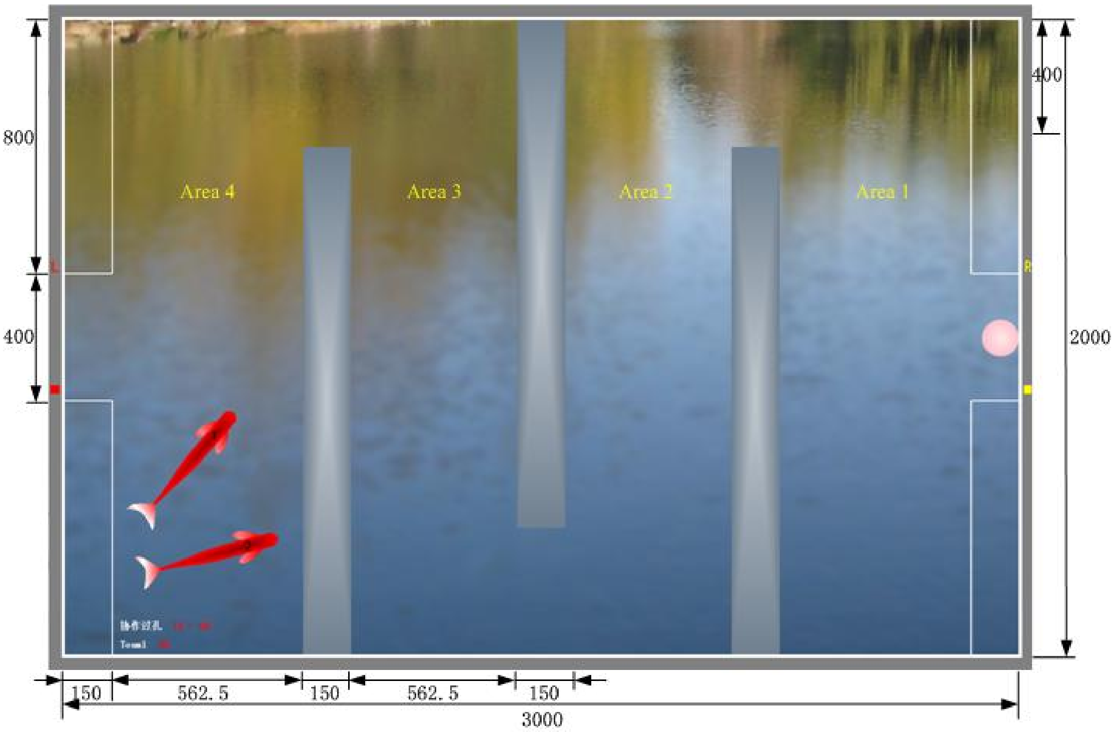

Figure 3-5 2D simulation passing holes in coordination

3.5.1.2. Match Process

1. In the initial state, 2 simulation robotic fish of 1 team occur at bottom left of the field randomly and the simulation water polo is inside the right goal.

2. When the match starts, 2 simulation robotic fish coordinate to push the simulation water polo into the left goal in the drive of strategy under the rules.

3. When 2 simulation robotic fish complete the mission before match time runs out, platform server will pop up a dialogue box. After the referee confirms time left, the match is over.

4. When match time runs out, platform server will pop up a dialogue box. After the referee confirms the score and time left, the match is over.

3.5.1.3. Match Rules

1. Coordinative object I: Both simulation robotic fish have touched the ball inside area 1 before the ball passes the first hole.

2. Coordinative object II: Both simulation robotic fish have touched the ball inside area 2 before the ball passes the second hole.

3. Coordinative object III: Both simulation robotic fish have touched the ball inside area 3 before the ball passes the third hole.

4. Coordinative object IV: Both simulation robotic fish have touched the ball inside area 4 before the ball is pushed into the left goal.

3.5.2. Match Time

Match time is 5 minutes and the match will be played twice. No timeout is allowed during each time of match. The strategy can be the same or different in two times.

3.5.3. Scoring Rules

1. For each object accomplished, 1 point is scored. The full mark is 4.

2. If all coordinative objects stages are accomplished in one time of match, the score and time left will be recorded.

3. If time runs out and the mission is not completed in one time of match, the score and time left (definitely 0) will be recorded.

4. The score and time left in 2 times of match will be added as total score and time left

3.5.4. Strategy Program

There are 9 special parameters of 2D simulation passing holes in coordination transmitting to strategy, including: IsCollidedBallAndFish0AtStage1 to represent whether simulation robotic fish 0 has touched the simulation ball at stage I; IsCollidedBallAndFish1AtStage1 to represent whether simulation robotic fish 1 has touched the simulation ball at stage I; IsCollidedBallAndFish0AtStage2 to represent whether simulation robotic fish 0 has touched the simulation ball at stage II; IsCollidedBallAndFish1AtStage2 to represent whether simulation robotic fish 1 has touched the simulation ball at stage II; IsCollidedBallAndFish0AtStage3 to represent whether simulation robotic fish 0 has touched the simulation ball at stage III; IsCollidedBallAndFish1AtStage3 to represent whether simulation robotic fish 1 has touched the simulation ball at stage III; IsCollidedBallAndFish0AtStage4 to represent whether simulation robotic fish 0 has touched the simulation ball at stage IV; IsCollidedBallAndFish1AtStage4 to represent whether simulation robotic fish 1 has touched the simulation ball at stage IV; ExecutingTaskNo to represent which is current stage.

The data type of first 8 parameters is Boolean with values of true or false. The data type of ninth parameter is integer with values of 1, 2, 3, 4, which represents current stage of the match.

The strategy can get and use aforementioned parameters in the following way.

```
int f01 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish0AtStage1"]);
int f11 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish1AtStage1"]);
int f02 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish0AtStage2"]);
int f12 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish1AtStage2"]);
int f03 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish0AtStage3"]);
int f13 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish1AtStage3"]);
int f04 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish0AtStage4"]);
int f14 = Convert. ToBoolean (mission.HtMissionVariables["IsCollidedBallAndFish1AtStage4"]);
int no = Convert. ToInt32 (mission.HtMissionVariables["ExecutingTaskNo"]);
```

if ((no == 1) && (f01 == true) && (f11 == true)) //when at stage I both simulation robotic fish have touch water polo in area 1, 1 point will be scored if the ball passes the first hole from right to left;

if ((no == 2) && (f02 == true) && (f12 == true)) // when at stage II both simulation robotic fish have touch water polo in area 2, 1 point will be scored if the ball passes the second hole from right to left.

f ((no == 3) && (f03 == true) && (f13 == true)) // when at stage III both simulation robotic fish have touch water polo in area 3, 1 point will be scored if the ball passes the third hole from right to left.

if ((no == 4) && (f04 == true) && (f14 == true)) // when at stage IV both simulation robotic fish have touch water polo in area 4, 1 point will be scored if the ball is pushed into the left goal.

3.6. 2D simulation synchronized swimming

3.6.1. Match Content

2D simulation synchronized swimming is a nonantagonistic match played by 1 team with 10 simulation robotic fish and without simulation water polo and simulation obstacles.

3.6.1.1. Match Field

The size of standard simulation field is 3000mm×2000mm. One and a half times of that size (4500mm×3000mm) is used by the game event of 2D simulation synchronized swimming as shown in figure 3-6 with 10 simulation robotic fish.2D

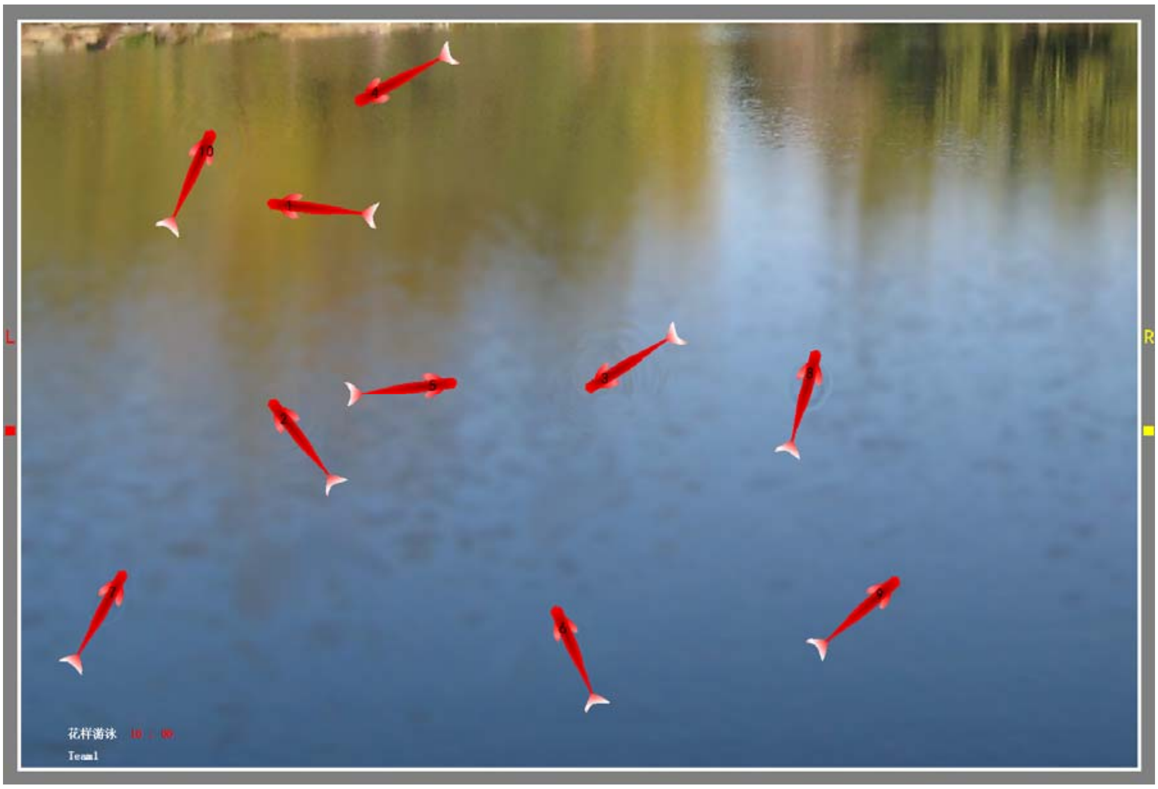

Figure 3-6 2D simulation synchronized swimming

3.6.1.2. Match Process

1. In the initial state, 10 simulation robotic fish of one team are distributed in the simulation field with random positions and orientations.

2. When the match starts, simulation robotic fish will perform in the drive of strategy. The performance can be designed by each team in strategy based on given information without any restriction of the platform.

3. When the performance ends within match time, the team should inform the referee that the match is over. The referee can give the score according to the appreciation and coordination.

4. When time runs out, the match is over regardless of whether the performance ends or not. The referee can give the score according to the appreciation and coordination.

3.6.1.3. Match Rules

The object of the match is to examine the appreciation and coordination of the operating result of the strategy. There is no restriction, such as foul, which provides sufficient free space to each time.

The platform provides the function of loading background music. Background music can be loaded through the interface of the server.

3.6.2. Match Time

Total match time is 5 minutes and the match is played only once. There is no timeout during the match.

3.6.3. Scoring Rules

1. The full mark of the match is 100. The referee can give the score according to the appreciation and coordination.

2. Every team sends one referee to form a referee team to score all teams. Among the scores of all referees, the highest one and lowest one will be eliminated and final score will be the mean of the rest.