CTF2009 Specification

Dennis Peters,
Jonathan Anderson, Lihong Zhang, Pouria Shaker, Andrew Vardy

Fall 2009
2009-10-01(#924)

Note: Changes in this document since the previous revision are indicated using strikeout and colour.

1 Overview

1. CTF2009 shall be a computer system for playing Capture the Flag. It will consist of two modules:

   Simulator Simulates a game of two-team Capture the Flag. Enforces the rules in this specification, represents the state of the game through a User Interface and presents a control interface to Controllers.

   Controller Connects to the Simulator to control a CTF team.

2. The Simulator and Controllers shall communicate over TCP/IP (Transmission Control Protocol/Internet Protocol), using a language called the STEAL (Simulator - TEAm Link) protocol.

2 Rules

1. Capture the Flag is a team game, in which each team attempts to find a flag that has been hidden on a field by their opponents, to capture it and bring it to their own side of the field. In the process, players "in enemy territory" may be captured by opposing players.

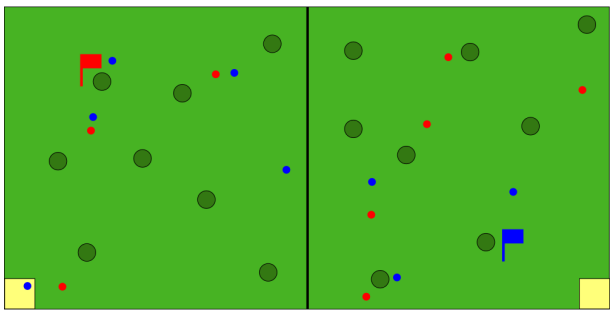

Figure 1: Visualization of a Capture the Flag Game

2.1 Basic Rules

2.1.1 Game Time

1. Each game of Capture the Flag will be ten minutes long.
2. In order to facilitate tournament-style play, each Simulator shall be able to simulate a game at double time (i.e. two seconds of game time are played in one second of real time) without violating the Correctness Requirements given in Section 3.2.1.
3. The "clock multiplier" shall be selectable via the User Interface.

2.1.2 Games and Rounds

1. A game is composed of a number of rounds (at least one).
2. There may be as many rounds as will fit into a 10-minute game.
3. A round is played until one of two events occurs:
   - A team captures its opponent's flag and brings it back to its own side of the field
   - A team captures all of its opponent's players
4. When a round ends, both teams shall re-place their flags and game play will resume.
5. When a new round begins, the game clock shall continue from its value at the end of the previous round (in the case of the first round, 0).

2.1.3 Scoring

1. Teams shall score points by capturing flags, capturing enemy players and releasing friendly players.
2. Points earned are shown in Table 1.

Action	Section	Points Earned
Capturing a Player	2.3.3	1
Capturing an Entire Team (bonus)	2.3.3	2
Releasing a Captured Player	2.3.3	1
Capturing the Flag	2.3.4	5
Reaching Home Territory with the Flag	2.3.4	10

Table 1: Points Earned

2.2 Physical Descriptions

2.2.1 Field

1. The field on which this game of Capture the Flag will be played is 500m long and 250m wide.
2. The field shall be divided by a centre line into two sides, left and right, each 250m square.

3. No part of any Capture the Flag object may move or be placed outside these boundaries.
4. All game activity shall occur within the plane of the field.
5. The directions "Up" and "North" shall be understood to mean "the direction of increasing y"
6. The orientation 0^o points straight North

2.2.2 Jails

1. There are two jails on the field - one per side - where captured players are kept (player capture is described in Section 2.3.3).
2. Each jail is 25x25 m in size.
3. Each jail is located in a corner of the field.
   (a) Left-side jail: (0,0)–(25,25).
   (b) Right-side jail: (475,0)–(500,25).
4. The boundaries of a jail do not obstruct player movement or vision, except where otherwise specified in Section 2.3.2.

2.2.3 Trees

1. There are ten trees on each side of the field.
2. Each tree is 2m in diameter.
3. In normal play mode trees are placed randomly at the beginning of the game, with the following constraints:
   (a) No part of any tree may extend past the boundaries of the field or cross the centre line.
   (b) No part of any tree may be in a jail.
   (c) Trees may not be placed less than 3m (centre to centre) apart.
   (d) Trees may not overlap with any player's starting position (see Table 2).
   (e) A team may only place a tree on its own side of the field ($x < 249$ or $x > 251$).
4. In Test Mode (see Section 3.1) trees are placed at the following standard locations:

(125, 35)	(375, 55)
(125, 55)	(375, 55)
(125, 75)	(375, 75)
(125, 95)	(375, 95)
(125, 115)	(375, 115)
(125, 135)	(375, 135)
(125, 155)	(375, 155)
(125, 175)	(375, 175)
(125, 195)	(375, 195)
(125, 215)	(375, 215)

5. Trees, once placed, shall remain fixed until the end of the game.

2.2.4 Flags

1. Each team has one flag.
2. A flag has no physical radius.
3. Each flag is placed by its team before the beginning of each round.
4. Restrictions on flag placement:
   (a) Each flag must be placed at least 1m away from the edge of the nearest object (field boundary, jail, tree or player).
   (b) A flag may not be placed within 25m of a player's starting position (see Table 2).
   (c) A flag may not be placed inside a jail.
   (d) A team may only place its flag on its side of the field (i.e. $x < 249$ or $x > 251$).

2.2.5 Players

1. There are eight players on each team.
2. Players are represented by circles, 40cm in diameter.
3. Player behaviour is governed by their current state, as given in Section 2.3.2.
4. The following constraints always apply to players (though further constraints may be imposed by current state):
   (a) A player's velocity is in the direction they are currently oriented.
   (b) A player's velocity may not exceed 5 m/s in magnitude.
   (c) A player may not move backwards.
   (d) A player's acceleration may not fall outside the range [-5,2] m/s^2.
   (e) A player may only turn to the left or to the right.
   (f) A player's angular velocity may not exceed 90 degrees per second in magnitude.
   (g) A player may not occupy the same space as another object.
5. Players begin each round located at the positions given in Table 2.

Player	Left-Side Team	Right-Side Team
0	(200,195)	(300,195)
1	(200,175)	(300,175)
2	(200,155)	(300,155)
3	(200,135)	(300,135)
4	(200,115)	(300,115)
5	(200,95)	(300,95)
6	(200,75)	(300,75)
7	(200,55)	(300,55)

Table 2: Player Start Positions

6. Players begin each round facing each other: those on the left-side team with an orientation of 90 degrees and those on the right-side team with an orientation of 270 degrees.

7. Players begin each round in a Defending state.

2.3 Game Play

2.3.1 Visibility

1. Whether or not a player or flag is visible to a team depends on:

   (a) The orientation of its players: each player can see objects within a 120 degree arc in from of him/her ($|\theta| < 60^o$ of his/her orientation)

   (b) The distance between the viewing player and the object: a player can see objects up to 75m away

   (c) The presence of other objects to block the player's view: a player's view is blocked by other players, trees and - sometimes - jail walls (see Section 2.3.2)

2. An object is considered visible if any part of that object can be seen. For example, in Figure 2:

   (a) Blue 1 is **visible**

   (b) Blue 2 is **invisible**, as the entire player is hidden by the tree

   (c) Blue 3 is **visible**, as part of the player is within red's visual range

   (d) Blue 4 is **visible**, as only part of the player is hidden by the tree

   (e) Blue 5 is **invisible**, as the entire player is outside the range $|\theta| < 60^o$

   (f) The Blue flag is **invisible**, as it is hidden by the tree

3. A team's own flag and players are always visible to that Controller.

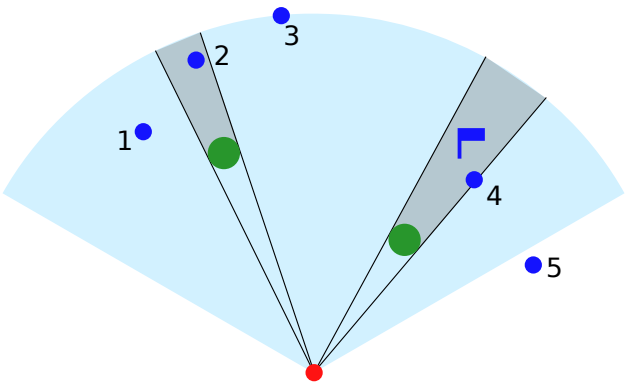

Figure 2: Field of View

2.3.2 Player States and Transitions

In addition to the constraints given in Section 2.2.5, player behaviour is governed by their current state.

1. Defending
   (a) A Defending player is one that is, even in part, located on their team's side of the field
   (b) Defending players cannot be captured.
   (c) If a Defending player crosses the centre line completely (i.e. no part of the player is on home territory), that player's state changes to Attacking
   (d) Defending players cannot enter within their own flag's Exclusion Zone - a circle around the flag having a radius equal to the smaller of:
      i. The distance from the flag to the closest enemy player
      ii. 25m
   (e) If the Exclusion Zone grows - due to enemy movement or capture - to include a Defending player, that player's state becomes TCTF (Too Close To Flag)
2. Attacking
   (a) If an Attacking player touches an enemy Defending player, the Attacking player becomes a Prisoner
   (b) If an Attacking player touches the enemy jail, s/he becomes an Escapee (whether or not there were prisoners to set free)
   (c) If an Attacking player touches the enemy flag, that player becomes a Flag Carrier
3. Prisoner
   (a) A Prisoner may not move through jail walls.
   (b) A Prisoner may not see through jail walls.
   (c) If a friendly player touches the jail, one Prisoner becomes an Escapee

4. Escapee

   (a) Escapees cannot be captured.

   (b) Escapees cannot capture the flag.

   (c) If an Escapee returns to home territory (i.e. any part of the player crosses the centre line), that player's state changes to Defending

   (d) If an Escapee does not return to home territory within 60s, that Escapee is automatically re-captured.

5. Flag Carrier

   (a) The maximum speed of a flag carrier is 4 m/s.

   (b) A Flag Carrier cannot release Prisoners.

   (c) If a Flag Carrier returns to home territory, the current round ends (see Section 2.3.4).

6. TCTF (Too Close To Flag)

   (a) TCTF players, upon exiting the Exclusion Zone around their flag:

      i. Change to the Defending state if on home territory

      ii. Change to the Attacking state if entirely on enemy territory

   (b) TCTF players, if they do not exit the Exclusion Zone within 7s, change to the Prisoner state.

2.3.3 Player Capture and Release

1. Player capture occurs when:

   (a) A Defending player touches an Attacking player or Flag Carrier (see Attacking in Section 2.3.2)

   (b) An Escapee does not return to home territory within 60s (see Escapee in Section 2.3.2)

   (c) A TCTF player does not exit the Exclusion Zone within 7s (see TCTF in Section 2.3.2)

2. When a player is captured:

   (a) The player's state changes to Prisoner (see Attacking in Section 2.3.2)

   (b) The player is immediately placed inside their enemy's jail

   (c) Their enemy's score is incremented by the amount given in Section 2.1.3

3. Prisoner release occurs when an Attacking player touches their enemy's jail (see Attacking in Section 2.3.2)

4. When a player is released:

   (a) The player's state changes to Escapee (see Prisoner in Section 2.3.2)

   (b) The player's team's score is incremented by the amount given in Section 2.1.3

2.3.4 Flag Captures

1. Flag capture occurs if an Attacking player touches their enemy's flag (see Attacking in Section 2.3.2).
2. When a flag is captured:
   (a) The capturing team scores points, as given in Section 2.1.3
   (b) The flag's location is exactly equal to that of the Flag Carrier
3. When a Flag Carrier is captured:
   (a) The flag is dropped where the capture occurred
   (b) The flag remains in that position unless captured again
4. When a Flag Carrier returns to home territory:
   (a) The Flag Carrier's team scores points, as given in Section 2.1.3
   (b) The current round ends

2.4 Collisions

1. When a player collides with an object that cannot be captured (e.g. tree, field boundary, player on same team):
   (a) The player's speed is set to 0
   (b) The player's linear acceleration is set to 0
   (c) The player's orientation and angular velocity are unchanged

3 Simulator

3.1 Test Mode

1. The Simulator shall incorporate a Test Mode which permits "Test Mode Only" commands (e.g. place player)
2. Unless in Test Mode, "Test Mode Only" commands shall be considered protocol errors
3. The Simulator's default shall be for Test Mode to be off
4. The Simulator, on startup, shall accept user input to turn Test Mode on
5. Once a Controller connects, the Simulator shall not switch Test Mode on or off

3.2 Functional Requirements

3.2.1 Correctness

1. The Simulator shall be implemented in Java, using no Operating System-specific extensions.

2. The Simulator shall act as a TCP/IP server, and will accept connections from STEAL controllers on a TCP port determined by the Simulator's colour, as given in Table 3.

Simulator Colour	TCP Port
Red	23450
Green	23451
Blue	23452
Purple	23453
Orange	23454
Black	23455

Table 3: Simulator Ports

3. Each simulator shall accept two TCP/IP connections per game. Once two connections have been accepted, no further connections may be accepted until that game is complete.

4. The language of Simulator-Controller interaction is given by the STEAL protocol, described in the document "STEAL Specification".

5. Any errors in STEAL protocol compliance shall immediately result in:

   (a) The current game ending

   (b) An error message being transmitted to each Controller:

      i. One to the offending Controller, which specifies the protocol error

      ii. One to the other Controller, which states that its opponent made a protocol error

   (c) The error message from 5(b)i being displayed on-screen

   (d) All TCP/IP connections being closed

6. At the end of a game, whether because of game time expiring or STEAL errors, the Simulator shall close both TCP connections (after sending error messages, if appropriate) and begin accepting new connections.

7. Error messages displayed by the Simulator shall not prevent new games from starting (i.e. they may be logged or displayed in a panel, but they must not be displayed using modal dialogs).

3.2.2 Performance Requirements

These requirements must be met on standard lab computers (EN2048 or EN3000). Teams are encouraged to ensure that their work exceeds requirements.

1. The Simulator must be able to service and respond to Controller requests within 0.1 seconds.
2. The Simulator must be able to service at least 10 requests per Controller per second.
3. STEAL Commands may not be acted on before the requested time.
4. Numerical error may cause some drift in player position. The Simulator shall not allow drift of more than 25cm per second.

4 Controller

Each Controller will control the actions of one team of virtual players. By giving simple commands (e.g. Accelerate, Turn) to the Simulator via STEAL, the Controller will move its players around the field and attempt to find the opposing controller's flag while defending its own flag.

4.1 Functional Requirements

1. The Controller shall be implemented in Java, using no Operating System-Specific extensions.
2. The Controller shall accept user input, indicating the IP host and TCP port of the Simulator.
3. The language of Simulator-Controller interaction is given by the STEAL protocol, described in the document "STEAL Specification v6.0".
4. At the end of the game, the Controller shall inform the user of the game's outcome (win, loss, draw).
5. The Controller shall only play one game (i.e. it shall not re-connect to the Simulator) without prompting the user.
6. Upon detecting a protocol error, the Controller shall send an error message to the Simulator.
7. After sending or receiving an error message, the Controller shall:
   (a) Close its connection to the Simulator
   (b) Display the error message to the user

4.2 Non-Functional Requirements

1. Each Controller must make a reasonable effort, as judged by the Course Instructor, to play the game well. For instance, a controller that makes its team do nothing but run in circles is unacceptable.