

Software Engineering 1

Universal Game System Initial Analysis

Krzysztof Kaczmarek, MiNI PW

20.12.2011

1 Abstract

Universal Game System organizes games between game applications. It is independent of any particular game rules, number of players required for a game or any other rule related requirements. It may organize simple games or contests between players.

Important: All games are automatic. There are no humans playing. All the system must work in fully automated fashion. Players are simulated by computers using artificial intelligence.

2 Glossary

Game activity performed together by a number of *players*. Each game must have same *rules* for all participants. Game consists of *moves* done subsequently by all *players* in given order. *Rules* are independent of the server. A game may begin when a number of required participants is collected. Strategy for collecting participants may be different for different games and is in responsibility of the *game master*. Also the game master decides when the game is finished and who is the winner.

Game finalization state of a *game* which closes single game. All *players* are informed about the game *result*.

Game initiation state of a game which collects *players*. This stage is started by game *server* and is controlled by *game master*.

Game master an application which connects to the *server*. It controls players registration (collecting), moves order and correctness of moves. Game master also defines the

game that is going to be played by collected group of *players*. Each move is sent first from a player to the server, then to the game master. If the move is correct the master sends confirmation (including the new game state) to the server which passes it to all players. Each game master may controll many games in the same time.

Game state information about game state that is sent between *players*, *game master* and *server*. Server passes this information to *game master* who is responsible for all rule related decisions. Game state is sent to all *players* after each move. Server is not interpreting this information in any way since it is different for different game types.

Game type checkers, chess, tic-tack-toe, ships, core-wars, etc.

Looser player who looses. There is no limitation on number of losers in single game. After a game is finished each player must be a looser or a winner.

Move single move performed by a single *player* in a *game* upon given *game state*.

Player (game participant) single game application that is capable of connecting to a *game server*, define its *game type*, receive game state information and perform moves using artificial intelligence.

Player pool collection of players who wait for being invited to a game. Each player who wants to join a game must first register in the pool for its particular *game type*. Game *server* uses this pool to select players for games.

Result a two state information about all players sent to each player when a game is finished: winner, looser. For example: player1: winner, player2: looser, player3: winner, player4: looser

Rules set of rules for particular *game*. Rules are defined implicitly by players who perform movements. Game *server* does not make up any rule related decisions. Correctness of moves is controlled by the game master.

Server organizes *games* between *players*. It generally passes messages between the game master and players. It does not make up any rule related decisions. They are all delegated to the *game master*. Game server may organize unlimited number of *games* in the same time. Game server chooses *players* from the *players pool*. Players never communicate directly but only via server.

Winner player who wins. There is no limitation on number of winners in single game. After a game is finished each player must be a looser or a winner.

3 Normal Activity Description

- A game master for given game type registers in the server. Server asks the master about particular game requirements (game name, number of participants required, etc.)
- Each player registers in the game server
- Game server places all players in the player pool.
- If there is enough players in the pool the server sends information about game initiation to the master (including players selected for game). The master is responsible for creating the first game state and points the first player to move.
- Game master:
 - sends game information to the server
 - receives list of players who participate this game
 - prepares game initial state
 - decides who makes the first move
 - sends move request to the selected player (via server)
 - receives move from the server
 - sends game state and move to the server (forwarded then to all participants)
 - decides if the game is finished or not
 - in such case sends special end game message to the server
 - decides who makes the next move
 - etc.
- Player:
 - receives a game state from the server
 - receives a move request from the server
 - makes move and sends it to the server
 - receives information about game final result
- Game server:
 - receives game state from game master
 - receives the number of next player to make a move
 - sends game state to players
 - sends move request to given player
 - receives move from player
 - sends this move to game master

- sends move result and game state to all players
- sends game final result to all players

4 Championship Mode Activity Description

In this mode server organizes championship between players registered in the player pool for particular game. Championship mode is possible only for one selected game type if and only if this game is for exactly two players.

- Championship is organized in such a way that each player plays with all other players.
- Wining player is the one with the greatest number of wins.
- Server selects pairs of players and organizes a game for them as described in Normal Activity.
- If a player is not responding for some time a server assumes that it is malfunctioning and other player wins a game.
- Server presents results of all games and presents the competitions winners.