

Homework 6 Design
Computer Science 182
James Reinke, Will Tachau
April 21, 2014

1. Tournament Manager

- This process is responsible for creating the tournament brackets upon request from the environment and keeping track of win/loss records for each member in the tournament
- Accepts “join” messages from **Players** to add them to the bracket, and begins a tournament when the conditions are right.
- All of the round statistics are managed by the **Match Manager** process
 - To that effect, **Match Manager** controls player cheating, the **Tournament Manager** is simply responsible for keeping track of what the Match managers reports from the Match
- The **Tournament Manager** keeps track of the wins/losses for each player in a single list throughout the tournament while simultaneously keeping track of the progression of the bracket
- At the end of the tournament, the **Tournament Manager** sends the bracket results to the console, along with a list of wins/losses for each player
 - It is necessary to send both because players may receive byes without winning any Matches, so both statistics must be recorded

2. Match Manager

- Responsible for starting a Match between two **Players**, as dictated by the **Tournament Manager**, and keeping track of scores.
- The **Match Manager** randomly selects 15 dice to be chosen for that round (assuming the round is not being played under *standard* Yahtzee rules) following the convention described in the homework assignment.
- Starts a turn by asking one **Player** for their move given the first five numbers in the dice roll. (Includes a time-out in this message to check if the player has logged out/ lost connection)

- If *both* **Players** lose connection, and do not join within a given time, the **Match Manager** reports to the **Tournament Manager** that the atom **bye** advances.
- **Player** processes report which dice they want to keep and if they want to use their dice to score points
 - This allows the Match process to ensure that no cheating occurs; it knows the possible legal moves for each player and allows the game to progress according to those conventions
 - If the player does not want to use their dice for points yet, they have the option of “re-rolling” twice, and the **Match Manager** responds with the next subset of dice.
- The Match manager keeps a list of dice that the players currently have, so when it receives a message from a player to use their dice for a specific slot on the score card, the Match manager will know whether or not this is a legal move and will respond accordingly
 - If a **Player** makes an illegal move (i.e. “cheats”), an invalid atom will be sent back to the player to make another move
 - If the **Player** makes too many consecutive invalid moves (say 3) then the Match manager drops that player from the Match and follows the same protocol as if that player had crashed.
 - After every turn, the **Match Manager** sends an “update” message to each **Player** containing the revisions to the overall scorecard.
- If a game results in a tie, then the Match manager discards it and starts a new game.
 - If there are $(k/2)$ consecutive ties, the **Match Manager** restarts the match under *standard* rules, where each of the players is given different dice configurations.

3. **Player**

- Players are independent processes that interact with the **Match Manager**’s API to play the match.
- A **Player** process is unable to cheat because it does not share its own score card with the **Match Manager**, but rather the intermediate moves it makes during the Match

- **Player** processes will send several atoms to the **Match Manager** to update game play
 - **choose**
 - * this atom is followed by a list of numbers that specify which dice it wants to keep from a given roll
 - **play**
 - * this atom is followed by a number item on our score list that specifies it wants its current hand to count for points in that spot
 - * The **Match Manager** knows what the player has chosen to keep and what it was given after keeping dice, so the **Match Manager** knows if this move is valid and how many points it should be worth for the player.