

Requirements

1. Game Module Requirements

- a. The system will hold descriptive metadata about the game rules, bot input and output, as well as the game itself. Essentially all information needed to represent the game High
- b. The system shall be able to generate game instances from the game module. High
- c. The system will allow admin to update, create, and delete Game Modules. High

2. Tournament Requirements

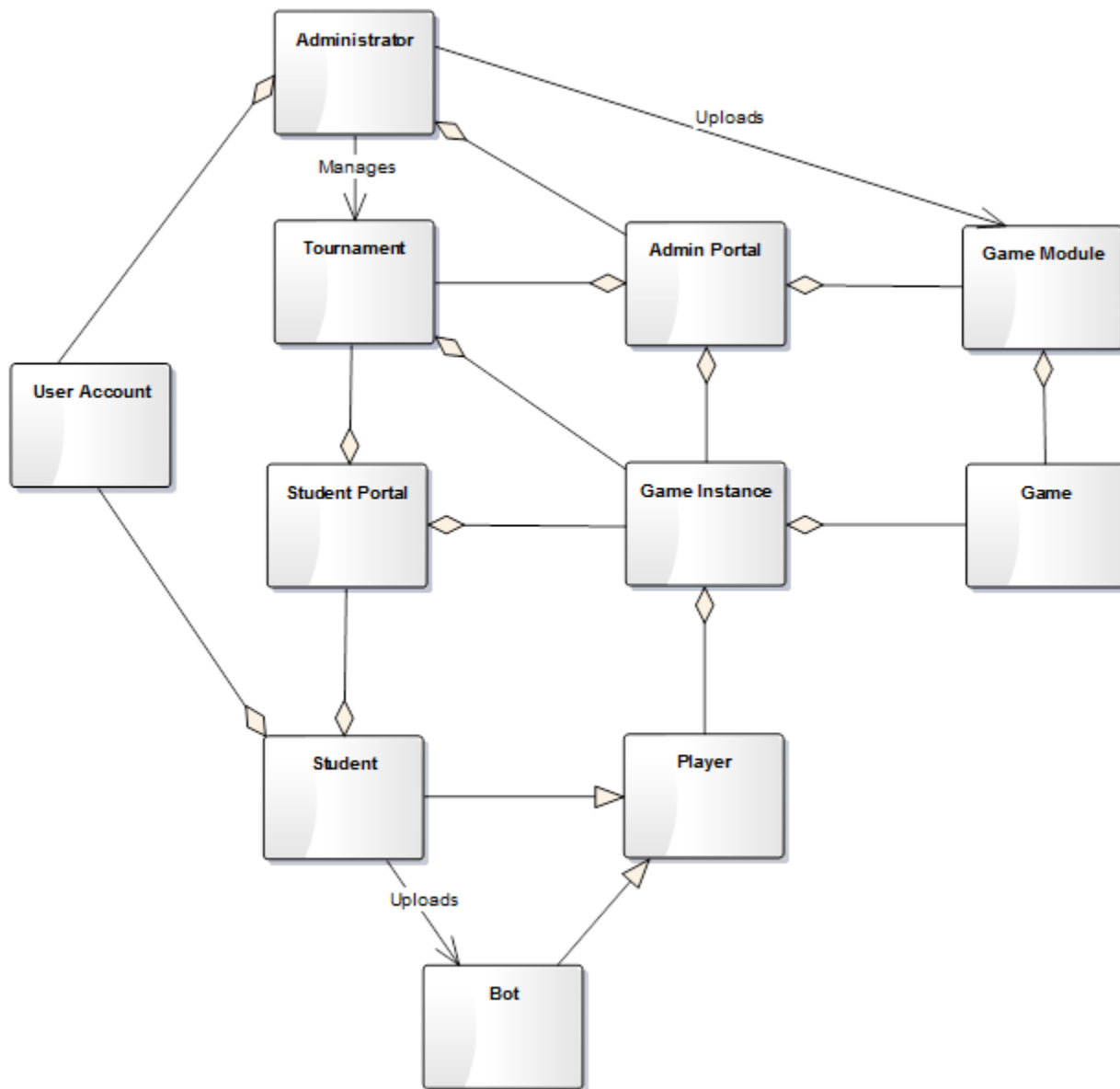
- a. Tournaments allow many bots to play against each other within individual game instances, until one ultimately wins. High
- b. Tournaments allow many bots and humans to play against each other, until one ultimately wins. Low
- c. All game instances within a tournament will play the same game. High
- d. Tournaments will have start and end dates for bot uploading. High
- e. The system will ensure the results of the games and tournaments are saved for later viewing. High
- f. The system will display the results via a web page. High
- g. The system will be responsible for ensuring all game rules are met. High

3. Privilege Requirements

- a. The system will have a Student Portal High
 - i. The system shall allow students to log in via a web page. High
 - ii. The system should allow students to submit their bots via a web page High
 - 1. Students shall be able to upload multiple bots for each game Med
 - 2. The system will associate the student's information with the submitted bots. High
 - 3. The system shall screen the bots for system compatibility and inform students of any potential problems. (e.g. does not compile) High
 - 4. The system shall save the bots. High
 - iii. Students will be able to run their own bots against each other or play the bot themselves High
 - 1. The bots view of the game should be visible to the student for debugging purposes. Med
 - 2. The system shall allow a student to upload two bots to play against each other. Med

| | | |
|------|--|------|
| b. | The system will have an Admin Portal | High |
| i. | The system will allow admins to log in via a web page | High |
| ii. | The system shall allow an admin to add, update, and remove game modules. | High |
| iii. | The system shall allow an admin to add, update, and delete student and admin accounts | High |
| iv. | The system shall allow an admin to delete game results. | High |
| v. | The system shall allow an admin to delete bots. | High |
| vi. | The system shall allow an admin to create, update, delete, start, pause, and end tournaments. | High |
| 4. | Game Requirements | |
| a. | The system shall allow game instances to communicate with bots via standard input and output. | High |
| b. | The system shall generate HTML representations of the current game board and state. | High |
| c. | The system shall limit game instances to exactly two players, whether human or bots. | High |
| d. | The system shall impose a time limit on a game instance's moves. | High |
| 5. | Bot Requirements | |
| a. | The system shall limit bots to reasonable memory and CPU usage. | Med |
| b. | The system shall run bots in an isolated execution environment to protect system stability. | High |
| c. | The system shall ensure Bots are generated in the format specified by the Game, otherwise they will be disqualified. | High |

Domain Model



Glossary

Student Portal: Allows students to create game instances, upload bots to their game instances, upload bots to running tournament, view active and past game instances and tournaments.

Admin Portal: Allows administrators to create, update, or delete tournaments; add, update and delete user accounts; create, update, and delete game modules, and access all features available to students under the Student Portal.

Game Module: All of the information needed to represent a game in the system. Including descriptive metadata about the game rules, bot input and output, and the game itself.

Game: The game handles verification of game moves fed to it by players (students or bots), tracks the state of the game, determines when win or loss conditions are met, and generates the HTML to display the game board.

Game Instance: This represents a currently running or already run game. This may include Bot vs Bot or Bot vs. Human. Consists of the two Players, and a Game.

Player: A Student or a bot

Bot: A computer program meant to play a game.

Student: A PSU computer science student

Administrator: An authorized individual responsible for maintenance of user accounts, tournaments, game modules, and all other configurable aspects of the system.

Tournament: A tournament will be a collection of games, segmented in such a way to find the “best” bot.

User Account: Either an administrator’s or a student’s user account