Game

- id: long {readOnly}
- key: UUID {readOnly}
- created: Instant {readOnly}
- userGame: List<UserGame> {readOnly}
- fleet: List<Ship> {readOnly}
- shots: List<Shot> {readOnly}
- boardPool: String
- playerCount: int {readOnly}

Fleet

- id: long {readOnly}
- userGame: UserGame
- game: Game
- ship: List<Ship> {readOnly}

Fleet reports new info to Game

Shot

- id: long {readOnly}
- fromUser: UserGame
- toUser: UserGame
- game: Game
- xCoord: int
- yCoord: int
- timestamp: Instant {readOnly}

Shot reports new info to Game

Shot is what the user hucks at their opponents. The User sends their shots to the UserGame and the UserGame will then query Game to process them.

User begins the game by creating their own instance of UserGame, and submitting their fleet. UserGame keeps tabs on the state of the Fleet.

Game is the main location of information. UserGames query Game for information, processes it, and returns it to the relevant Users.

Ship

- id: long {readOnly}
- fleet: Fleet {readOnly}
- xCoord: int
- yCoord: int

The Ship is the object that makes up the Fleet. It is also the target for opposing users

UserGame

- id: long {readOnly}
- user: User
- game: Game
- fleet: Fleet

UserGame is the liason between the User and the rest of the game. Each User has their own UserGame.

User

- id: long {readOnly}
- key: UUID {readOnly}
- created: Instant {readOnly}
- displayName: String
- oauthKey: String
- userGame: UserGame {readOnly}

- generateKey(): UUID

The User is, of course, the class representing the Player.