The Game entity contains the information pertaining to gameplay.

Game

- id: long {readOnly}
- key: UUID {readOnly}
- created: Instant {readOnly}

currentUserGame: UserGame

- userGames: List<UserGame> {readOnly}
- boardSize: int
- turnCount: Long
- playerCount: int {readOnly}

- generateKey(): UUID

ShipLocation is the entity responsible for storing each player's ship locations.

ShipLocation

+ MIN_SHIP_NUMBER: int {readOnly}
+ MIN_X_COORD: int {readOnly}
+ MIN_Y_COORD: int {readOnly}
- id: long {readOnly}
- userGame: UserGame {readOnly}
- shipNumber: int

location:Location

1

Game is where the business of playing the game happens. When the user opens the app, a game object is created and then the game initializes the user's UserGame instance.

UserGame is the central information hub. Any information it does not have, it gets by querying the other entities.

*

The user will place all of their ships before they can join a game. UserGame owns the ShipLocation instances and updates them as the game progresses.

UserGame

- id: long {readOnly}
- key: UUID
- user: User {readOnly}
- game: Game {readOnly}
- locations: List<ShipLocation>
- fromShots: List<Shot>
- toShots: List<Shot>

turnCount: int

+ hashCode(): int
+ equals(Object obj): boolean
- generateKey(): UUID

UserGame is the liason between the user and the rest of the game. All information coming to User goes through UserGame first.

*

2

The user fires/creates a volley of shots. The shots are reported to UserGame to be processed.

1

*

User

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

+ hashCode(): int
+ equals(Object obj): boolean
- generateKey(): UUID

The User entity represents the player.

Shot

- id: long {readOnly}
- fromUser: UserGame {readOnly}
- toUser: UserGame

location: Location

- timestamp: Instant {readOnly}

+ isHit(): boolean

The User fires shots at other users' ships.