## Jungle: Iteration 3

Team And Yet It Compiles

Laura South, Isaac Mauro, Brian Larson, Yan Wang, Duck Keun Yang

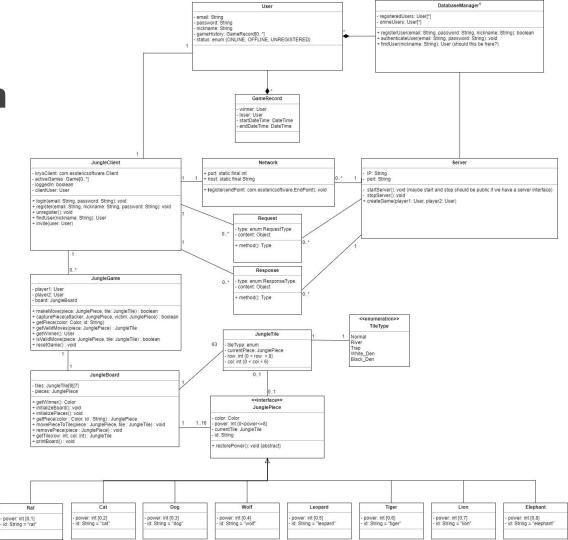
## Changes from the previous iteration

- Invitation / Profile will not be implemented as separate classes
- JungleServer/JungleClient/DatabaseManager classes added.
  JungleGame and User are now managed and created by JungleClient and JungleServer
- Created several classes used to store information that is sent between JungleServer and JungleClient
- Glossary of the second iteration is now in alphabetical order

### **Implementation Decisions**

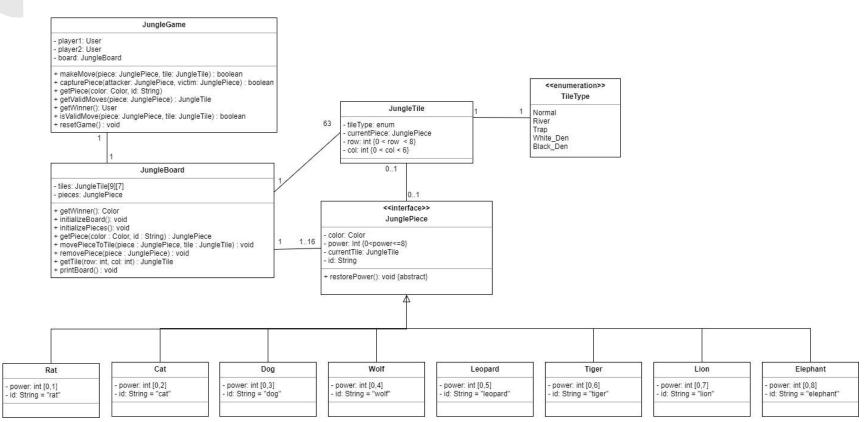
- Decided to use KryoNet library for network communications.
  This allows us to focus on the problem domain without worrying about network code and protocols
- JUnit is being used for unit testing
- JavaFX is being used for the UI
- We setup a build configuration using Maven so it is easier to manage dependencies and lays the foundations for implementing continuous integration.

# Class Diagram (Overview)



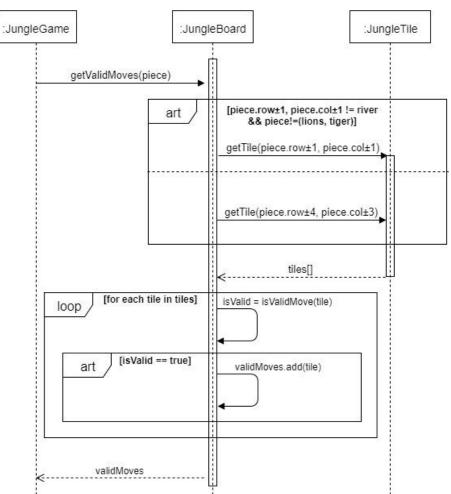
DatabaseManager<sup>1</sup> User Class Diagram email: String registeredUsers: User[\*] onineUsers: User[\*] - password: String nickname: String + registerUser(email: String, password: String, nickname: String): boolean - gameHistory: GameRecord[0..\*] + authenticateUser(email: String, password: String); void status: enum (ONLINE, OFFLINE, UNREGISTERED) (part 1) + findUser(nickname: String): User (should this be here?) GameRecord winner: User loser: User startDateTime: DateTime endDateTime: DateTime JungleClient Network Server kryoClient: com.esotericsoftware.Client + port: static final int IP: String activeGames: Game[0..\*] + host: static final String port: String loggedin: boolean clientUser: User +1register(endPoint: com.esotericsoftware.EndPoint): void startServer(): void (maybe start and stop should be public if we have a server interface) stopServer(): void + login(email: String, password: String): void createGame(player1: User, player2: User) + register(email: String, nickname: String, password: String): void + unregister(): void + findUser(nickname: String): User Request + invite(user: User) type: enum RequestType content: Object + method(): Type Response - type: enum ResponseType content: Object + method(): Type

### Class Diagram (part 2)



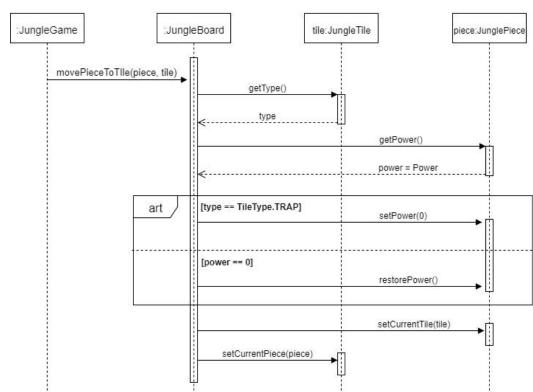




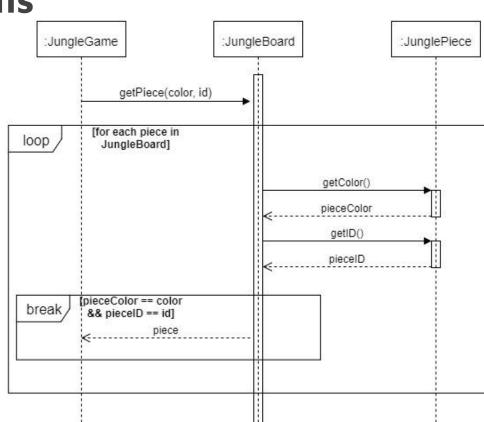




#### movePieceToTile







getPiece

# Demo