

<<Enumeration>> Era	
ANCIENT	
CLASSICAL	
MEDIEVAL	
RENAISSANCE	
INDUSTRIAL	
MODERN	
FUTURE	

<<Enumeration>> TerrainType	
...	
-id	
-foodYield: int	
-productionYield: int	
-goldYield: int	
-combatModifier: int	
-movementCost: int	
-featuresPossible: ArrayList<TerrainFeature>	
-resourcesPossible: ArrayList<Resource>	
+TerrainFeature: <<Enumeration>>	

<<Enumeration>> UnitType	
...	
-id	
-cost: int	
-combatType: CombatType	
-combatStrength: int	
-rangedCombatStrength: int	
-range: int	
-movement: int	
-resourceRequired: Resource	
-technologyRequired: Technology	
+CombatType: <<Enumeration>>	
+UnitAction: <<Enumeration>>	

<<Enumeration>> BuildingType	
...	
-memberName	
-memberName	

<<Enumeration>> Specialist	
ARTIST	
ENGINEER	
MERCHANT	
SCIENTIST	

<<Enumeration>> Direction	
UP_RIGHT	
RIGHT	
DOWN_RIGHT	
DOWN_LEFT	
LEFT	
UP_LEFT	
-id: int	

<<Enumeration>> Technology	
-id: int	
-cost: int	
-technologiesRequired: ArrayList<Technology>	
-technologiesLeadsTo: ArrayList<Technology>	
-era: Era	
-objectsUnlocks: ArrayList<Object>	

<<Enumeration>> Improvement	
-id: int	
-foodYield: int	
-productionYield: int	
-goldYield: int	
-technologyRequired: Technology	
-terrainsRequired: ArrayList<Terrain>	
+getImprovementById(id: int): Improvement	

<<Enumeration>> Resource	
-id: int	
-foodYield: int	
-productionYield: int	
-goldYield: int	
-terrainsPossible: ArrayList<Terrain>	
-improvementRequired: Improvement	
-technologyRequired: Technology	
-type: Type	
+ResourceType: <<Enumeration>>	

Civilization

- name: String
- capital: City
- cities: ArrayList<City>
- destroyedCities: ArrayList<City>
- annexedCities: ArrayList<City>
- puppetCities: ArrayList<City>
- units: ArrayList<Unit>
- selectedCity: City
- selectedUnit: Unit
- currentResearch: Technology
- population: int
- happiness: int
- unhappiness: int
- gold: int
- science: int
- resourceCount: int[]
- technologyReached: boolean[]

- +getCivilizationByName(civName: String): Civilization
- +getTotalHappinessLevel(): int

Tile

- neighbors: ArrayList<Tile>
- hasRiverOnSide: ArrayList<Boolean>
- index: int
- coordinate: Pair<double, double>
- terrain: TerrainType
- terrainFeature: TerrainType.TerrainFeature
- tileKnowledge: TileKnowledge
- resources: ArrayList<Resource>
- improvement: Improvement
- building: Building
- unit: Unit
- hasRoad: boolean
- hasRailRoad: boolean

- +isBlock(): boolean

- +Pair: <<Class>>
- +TileKnowledge: <<Enumeration>>

Unit

- unitType: UnitType
- unitAction: UnitAction
- mp: int
- hp: int
- civilization: Civilization
- tile: Tile

Building

- memberName

- memberName

City

- name: String
- civilization: Civilization
- tile: Tile
- garrisonedUnit: Unit
- territory: ArrayList<Tile>
- workingTiles: ArrayList<Tile>
- currentProduction: Enum<?>
- turnsLeftForProductionConstruction: Enum<?>[]
- combatStrength: int
- hp: int
- population: int
- happiness: int
- unhappiness: int
- food: int
- production: int
- foodYield: double
- goldYield: double
- productionYield: double
- scienceYield: double

- +getCityByName(cityName: String): City
- +getBuildingTypesInCity(): ArrayList<Enum<BuildingType>>

Map

- tiles: ArrayList<Tile>

- generateRandomMap(seed: int): void

User

- users: ArrayList<User>
- username: String
- password: String
- nickname: String
- score: int
- isLogin: boolean

- +hasUser(user: User): boolean
- +addUser(user: User): boolean
- +getUser(username: String): User
- +checkPassword(password: String): boolean

GameArea

- map: Map
- user2civ: HashMap<User, Civilization>
- turn: int
- year: int
- era: Era

- +addUser(user: User, civ: Civilization): boolean
- +getUserByCivilization(civ: Civilization): User
- +getCivilizationByUser(user: User): Civilization

UserController

```

+login(username: String, password: String): boolean
+register(username: String, password: String, nickname: String): boolean
+changeNickname(username: String, newNickname: String): boolean
+changePassword(username: String, password: String, newPassword: String): boolean

```

GameController...

```

-gameArea: GameArea
+getCivilizationByUsername(username: String): JsonObject
+getAllCivilizations(): JsonObject
+getAllPlayers(): JsonObject
+newGame(players: User[]): JsonObject
+nextTurn(username: String): JsonObject
+infoResearch(username: String): JsonObject
+infoUnits(username: String): JsonObject
+infoCities(username: String): JsonObject
+infoDiplomacy(username: String): JsonObject
+infoDemographics(username: String): JsonObject
+infoNotifications(username: String): JsonObject
+infoMilitary(username: String): JsonObject
+infoEconomic(username: String): JsonObject
+infoDiplomatic(username: String): JsonObject
+infoDeals(username: String): JsonObject
+selectCity(username: String, cityName: String): JsonObject
+selectCity(username: String, tileId: int): JsonObject
+selectUnit(username: String, tileId: int, isCombatUnit: boolean): JsonObject

```

...GameController...

```

+unitMoveTo(username: String, tileId: int): JsonObject
+unitSleep(username: String): JsonObject
+unitAlert(username: String): JsonObject
+unitFortify(username: String, untilHeal: boolean): JsonObject
+unitGarrison(username: String): JsonObject
+unitSetupForRangedAttack(username: String): JsonObject
+unitAttack(username: String, tileId: int): JsonObject
+unitFoundCity(username: String): JsonObject
+unitCancelMission(username: String): JsonObject
+unitWake(username: String): JsonObject
+unitDelete(username: String): JsonObject
+unitBuildRoad(username: String): JsonObject
+unitBuildRailRoad(username: String): JsonObject
+unitBuildImprovement(username: String, improvementId: int): JsonObject
+unitRemoveJungle(username: String): JsonObject
+unitRemoveRoute(username: String): JsonObject
+unitRepair(username: String): JsonObject
+mapShow(username: String, tileId: int): JsonObject
+mapShow(username: String, cityName: String): JsonObject
+mapMove(username: String, directionId: int, count: int): JsonObject

```

...GameController

```

+cityLockCitizenToTile(username: String, tileId: int): JsonObject
+cityRemoveCitizenFromWork(username: String, tileId: int): JsonObject
+cityGetOutput(username: String): JsonObject
+cityGetUnemployedCitizens(username: String): JsonObject
+cityGetBuildings(username: String): JsonObject
+cityPurchaseTile(username: String, tileId: int): JsonObject
+cityGetCurrentProduction(username: String): JsonObject
+citySetCurrentProduction(username: String, productionId: int): JsonObject
+cityBuyProduction(username: String, productionId: int): JsonObject
+cityGetAllAvailableProductions(username: String): JsonObject
+cityGetWorkingTiles(username: String): JsonObject
+cityWorkOnTile(username: String, tileId: int): JsonObject
+cityAssignSpecialist(username: String, tileId: int): JsonObject
+cityDestroy(username: String, tileId: int): JsonObject
+cityAnnex(username: String, tileId: int): JsonObject
+cityPuppet(username: String, tileId: int): JsonObject

```

AbstractMenuView

#scanner: Scanner
#user: User
#menu: Menu
#userController: UserController
#gameController: GameController

+run(): void
+constantCaseToCamelCase(constCaseStr : String): String
+getCommandMatcher(input: String, regex: Enum<?>): Matcher
+runCommand(): boolean
+getCommands(): Enum<?>[]
+enterMenu(matcher: Matcher): Menu
+exitMenu(matcher: Matcher): Menu
+showMenu(matcher: Matcher): Menu

LoginMenuView

+getCommands(): Enum<?>[]
+enterMenu(matcher: Matcher): Menu
+exitMenu(matcher: Matcher): Menu
+showMenu(matcher: Matcher): Menu
+register(matcher: Matcher): Menu

+login(matcher: Matcher): Menu

+Command: <<Enumeration>>
+Validator: <<Enumeration>>
+Message: <<Enumeration>>

MainMenuView

+getCommands(): Enum<?>[]
+enterMenu(matcher: Matcher): Menu
+exitMenu(matcher: Matcher): Menu
+showMenu(matcher: Matcher): Menu
+logout(matcher: Matcher): Menu
+newGame(matcher: Matcher): Menu

+Command: <<Enumeration>>
+Validator: <<Enumeration>>
+Message: <<Enumeration>>

GameMenuView

+getCommands(): Enum<?>[]
+enterMenu(matcher: Matcher): Menu
+exitMenu(matcher: Matcher): Menu
+showMenu(matcher: Matcher): Menu
+info(matcher: Matcher): Menu
+select(matcher: Matcher): Menu
+unitAction(matcher: Matcher): Menu
+showMap(matcher: Matcher): Menu
+moveMap(matcher: Matcher): Menu
+cityAction(matcher: Matcher): Menu

+Command: <<Enumeration>>
+Validator: <<Enumeration>>
+Message: <<Enumeration>>

ProfileMenuView

+getCommands(): Enum<?>[]
+enterMenu(matcher: Matcher): Menu
+showMenu(matcher: Matcher): Menu
+exitMenu(matcher: Matcher): Menu
+change(matcher: Matcher): Menu

+Command: <<Enumeration>>
+Validator: <<Enumeration>>
+Message: <<Enumeration>>

<<Enumeration>>

Menu

LOGIN
MAIN
GAME
PROFILE

-menuView: AbstractMenuView

+runCommand(): boolean
+getMenuByName(menuName: String): Menu