Menu
- matcher: Matcher
- input: String
- menuName : MenuNames

+ run();

<<Enumeration>> MenuNames

LoginMenu

MainMenu

ProfileMenu

Extends
Extends
Extends
Extends

GameMenu

UserController
- ArrayList<User> users

+createUser(username: String , password: String , nickname: String): void
+getUser(username: String , password: String): User
+login(username , password) : void

Main
+main(args:String[]): void

LoginMenu
- input: String
- matcher: Matcher

+ run()

<<Enumeration>> LoginMenuCommands
- regex: String

LoginMenuController
+ menuNavigation()

MainMenu
- input: String
- matcher: Matcher

+ run()

<<Enumeration>> MainMenuCommands
- regex: String

MainMenuController
+ menuNavigation();

ProfileValidation

+usernameIsValid(username : String) : Boolean
+passwordIsValid(password : String) : Boolean
+nicknameIsValid(nickname : String) : Boolean
+usernameIsUsed(username: String) : Boolean
+nicknameIsUsed(nickname) : Boolean

<<Enumeration>> ProfileMenuCommands
- regex : String

ProfileMenu
- input: String
- matcher: Matcher

+ run()

GameController
- ArrayList<Terrain> map
- ArrayList<Civilization> civilizations
- time : int
- turn : int;

+ run()
+ printMap(): String;
+ move(direction: String): void
+ move(location Location): void
+showCity(cityName : String): String.

User
- username: String
- password : String
- nickname: String
- Score: int
- isLoggedIn: boolean

+ setter And getter
+User <<constructor>>

ProfileMenuController

+ changeNickname(): void
+ changePassword(): void

<<Enumeration>> GameMenuCommands
- regex: String

GameMenu
- input: String
- matcher: Matcher

+ run()

Cheat
+ increaseTurn(int);
+ increaseGold(int);

City
- Arraylist<Terrain> terrains;
- Arraylist<Citizen> citizens;
- name: String
- happiness : int
- food : int
- gold : int
- production: int
- hp : int
- ownership: Civilization

CityController
- ArrayList<UnitCreator> unitsToBeCreated;

+ handleCreatingUnit(TypeOfUnit , City) : void;
+ createUnit() : String;
+ buyTiles(Terrain) : String
+ update();
+ addCitizen();

Civilization
- Arraylist<City> cities;
- Arraylist<Unit> units;
- ArrayList<Terrain> knownTerrains;
- Arraylist<Technology> gainedTechnologies;
- name : String
- capital: City
- score : int
- food : int
- gold : int
- happiness: int
- science: int
- currentResearch: Technology

Citizen
- terrain: Terrain

+ assignWork(Citizen , Terrain);
+ removeWork(Citizen , Terrain);

civilizationController
+ civilization: Civilization

+ research(Technology) : String
+ nextTurn() : void
+ updateHappiness();
+ updateFood();
+ updateGold();
+ updateResearch();
+ updateFogOfWar();
+ updateScience();

SelectController
+ selectedUnit: Unit
+ selectedCity: City

- selectCombatUnit(location Location): String
- selectNonCombatUnit(location Location): String
- positionIsValid(location Location): boolean
- selectCity(String name): String
- selectCity(location Location): String

UnitController
+ moveUnit(Unit , Location): String;
+ hasOwnership(Unit) : boolean;
+ sleep(Unit) : String;
+healUnit(unit): void
+ cancelMission(Unit): String;
+wake(Unit): String;
+deleteUnit(Unit): String;
+upgrade(Unit): String

<<Enumeration>> TypeOfImprovement
- name : String
- gold: int
- food: int
- production: int
- technology: TypeOfTechnology;
- turnsNeeded: int
- canBeFoundOn: String[]

Improvement
- typeOfImprovement : TypeOfImprovement
- turn : int

Terrain
- typeOfTerrain: TypeOfTerrain;
- terrainFeatures: TerrainFeatures;
- hasRiver: boolean
- Arraylist<Resource> resources;
- location Location
- improvement: Imrpvement

Technology
- typeOfTechnology :
- turn : int

<<Enumeration>> TypeOfTechnology
- name : String;
- cost : int;
- prerequisiteTech : String;
- leadsToTech : String
- turnsNeeded: int;
- scienceNeeded: int;
- Unlocks: TypeOfTechnology[]

Unit
- typeOfUnit : TypeOfUnits
- location : Location;
- hp: int
- civilization: Civilization
- status : UnitStatus
- turn : int

CombatUnitController
- isMilitary(Unit): boolean;
- alert(Unit): String;
- fortify(Unit): String;
- getCurrentCity(Unit , Arraylist<City> cities): City
- garrison(unit Unit , city City);
- isSiegeUnit(Unit) : boolean;
- setUpUnit(Unit): String;
- canAttack(Unit , location): boolean;
- getCityBylocation(location): City;
- getUnitByLocation(location): Unit;
- attack(Unit , City) : String;
- attack(Unit, enemyUnit); String

Extends

NonCombatUnitController
- canFoundCityHere(location , map);
- isSetterUnit(Unit): boolean;
- foundCity(location): String;
- isWorker(Unit): boolean;
- hasImprovement(improvement: String , location);
- canBuildImprovement(improvement: String) : boolean
- buildImprovement(improvement: String , terrain: Terrain): String
- getTerrainByLocation(location): Terrain;
- needsRepairing(location): boolean;
- repair(location): String;
- removeJungle(): String;
- removeRoute(): String;
- isJungleHere(location): boolean;
- isRouteHere(location): boolean;

Extends

<<Enumeration>> UnitStatus

<<Enumeration>> TypeOfUnits
- name : String
- cost : int
- combatType : String
- combatStrenght : int
- rangedCombatStronght : int;
- range : int;
- mp: int;
- resource : Resource
- technologyRequired : Technology;
- hp: int
- turnsNeededToCreate: int

Location
- X: int
- Y: int

<<Enumeration>> TypeOfTerrain
- name : String
- gold: int
- food: int
- production: int
- mpNeeded: int;
- changeOfCombat: int;
- possibleFeatures: TerrainFeatures[]
- possibleResources: Resources[]

<<Enumeration>> Resources
- Name: String
- resourceType: String
- food : int
- production: int
- gold: int
- canBeFoundOn: String
- technologyNeeded: TypeOfTechnology
- improvementNeeded: TypeOfImprovement

Info
- Arraylist<Civilization> civilizations

- showResearches(): String;
- showUnits(): String;
- showCities(): String;
- showDemographics(): String;
- showNotifications(): String;
- showMilitaryUnits(): String;
- showEconomics(): String;

<<Enumeration>> TerrainFeatures
- name: String
- food: int
- production: int
- gold: int
- mp: int
- changeOfCombat: int
- possibleResources: String[]