«Controller»
Package::Controller
- +curretnUser: User
- +government: Government
- +map
- +getTradeList
- +getTradeHitory

«LoginMenuCommands»
Package:: View

«LoginMenu»
Package::Viwe
- -controller: loginMenucontroller
- +run(Scanner): void
- -createUser(Matcher): void
- -loginUser(Matcher): void
- -forgrtPassword(): void

«LoginMenuController»
Package::Controller
- +createUser(userName: String, password: String, nickname: String, email: String, slogan: String, recoveryQuestion: RecoveryQuestion, recoveryAnswer): Messages
- +loginUser(username: String, password: String): Messages
- +forgetPassword(answer: String): Messages

«User»
Package::Model
- +Users: ArrayList<User>
- -username: String
- -password: String
- -nickname: String
- -email: String
- -slogan: String
- -recoveryQuestion: RecoveryQuestion
- -recoveryAnswer: String
- -lastAttempt: long
- -numberOfAttempts: int
- +checkPassword(String): boolean
- +getters()
- +setLastAttempt(): void
- +getLastAttempt(): Long
- +resetAttempt(): void
- +setNewPassword(): void
- +checkPasswordFormat(String): boolean
- +checkUsernameFormat(String): boolean

«ProfileMenuCommands»
Package:: View

«ProfileMenu»
Package::View
- -controller: ProfileMenuController
- +run(): void
- -profileChange(Matcher): void
- -profileRemoveSlogan(): void
- -profileDisplayHighscore(): void
- -profileDisplayRank(): void
- -profileDisplaySlang(): void
- -profileDisplay(): void

«ProfileMenuController»
Package::Controller
- +profileChangeUsername(String): Messages
- +profileChangeNickname(String): Messages
- +profileChangePassword(String): Messages
- +profileChangeEmail(String): Messages
- +profileChangeSlogan(String): Messages
- +profileRemoveSlogan(): Messages
- +profileDisplayHighscore(): Messages
- +profileDisplayRank(): Messages
- +profileDisplaySlang(): Messages
- +profileDisplay(): Messages

«Messages»
Package:: Controller

«Bulding»
Package::Model
- -hp: int

«MapPixel»
Package::Model
- -bulding: Bulding
- -texture: Texture
- -doesHaveOil: boolean
- +getters()

«Map»
Package::Model
- -height: int
- -width: int
- -map: ArrayList<MapPixel>
- +getters()

«Government»
Package::Model
- -popularity: int
- -foodList: ArrayList<Food>
- -taxRate: int
- -fearRate: int
- -foodRate: int
- -stoneAmount: int
- -population: int
- -money: int
- -wheat: int
- -flour: int
- -hops: int
- -ale: int
- -iron: int
- -wood: int
- -pitch: int
- +getters()

«GameMenu»
Package::View
- -controller: GameMenuController
- -selectedBuldingPos: int[2]
- +run(): void
- -showMap(Matcher): viod
- -moveMap(Matcher): void
- -showDetails(Matcher): void
- -showPopularityFactors(): void
- -showPopularity(): void
- -showFoodList(): void
- -setFoodRate(Matcher): void
- -showFoodRate(): void
- -setTaxRate(Matcher): void
- -showTaxRate(): void
- -setFearRate(Mathcer): void
- -dropBuilding(Matcher): void
- -selectBulding(Matcher): void
- -createUnit(Matcher): void
- -repirdBulding(Matcher): void
- -selectUnit(Matcher): void
- -moveUnit(Matcher): void
- -patrolUnit(Matcher): void
- -setState(Matcher): void
- -attackWithUnit(Matcher): void
- -airAttack(Matcher): void
- -pourOil(Matcher): void
- -digTunnel(Matcher): void
- -buildEquipment(Matcher): void
- -disband unit(): void
- -setTextureOfPixel(Matcher): void
- -setTextureOfArea(Matcher): void
- -clearPixel(Matcher): void
- - dropRock(Matcher): void
- - dropTree(Matcher): void
- -dropBuilding(Matcher): void
- -dropUnit(Matcher): void
- -addTrade(Matcher): void
- -showTradeList(): void
- -acceptTrade(Matcher): void
- -tradeHistory(): void
- -showPriceList(): void
- -buyFromShop(Matcher): void
- -sellToShop(Mathcer): void
- -endTurn(): void

«GameMenuController»
Package::Controller
- +selectedUnit: unit
- +showMap(x, y): Messages
- +moveMap(direction1, amount1, direction2, amount2): Messages
- +showDetails(x, y): Messages
- +showPopularityFactors(): Messages
- +showPopularity(): Messages
- +showFoodList(): Messages
- +setFoodRate(int): Messages
- +showFoodRate(): Messages
- +setTaxRate(int): Messages
- +showTaxRate(): Messages
- +setFearRate(int): Messages
- +dropBuilding(x, y, type): Messages
- +selectBulding(x,y): Messages
- +createUnit(x:int, y:int, type, count: int): Messages
- +repirdBulding(Bulding): Messages
- +selectUnit(x,y): Messages
- +moveUnit(x,y): Messages
- +patrolUnit(x1,x2,y1,y2): Messages
- +setState(x,y,defenfingState): Messages
- +attackWithUnit(enemy: User):Messages
- +airAttack(x,y): Messages
- +pourOil(direction): Messages
- +digTunnel(x,y): Messages
- +buildEquipment(equipment: Equipment): Messages
- +disband unit(): Messages
- +setTextureOfPixel(x ,y ,textuer: Texure): Messages
- +setTextureOfArea(x1,x2,y1,y2,texure): Messages
- +clearPixel(x,y): Messages
- +dropRock(x,y,direction): Messages
- +dropTree(x,y,treeType: Tree): Messages
- +dropBuilding(x,y,building : Building): Messages
- +dropUnit(x,y,unitType: Unit,count): Messages
- +addTrade(resourceType: ResourceType, amount, price , message: String): Messages
- +showTradeList(): Messages
- +acceptTrade(id, message: Sting): Messages
- +tradeHistory(): Messages
- +showPriceList(): Messages
- +buyFromShop(name, amount): Messages
- +sellToShop(name, amount): Messages
- +endTurn(): void

«Direction»
Package::Controller
- x:int
- y:int

«Tree»
Package::Controller

«Building»
Package::Controller

«Unit»
Package::Controller

«Texture»
Package::Controller

«Equipment»
Package::Controller

«ResourceType»
Package::Controller

«DefendingState»
Package::Model

people

«Unit»
Package::Model.People
- #position: int[2]
- #defendingState: DefendingState
- #speed: int
- #enduranceL int
- #damage: int

«People»
Package::Model.People