

destination: int wizard: Wizards username: String

me: Player

cards: List<AssistantCard>

server: Message4Server characters: CharacterCard[]

Archipelago

+ Archipelago(ArrayList<Island>):

previousTowers: ArrayList<Tower>

+ uniteIslands(Land): Archipelago

+ hasChanged(): boolean

+ addStudent(Student): void

thereNoEntry: boolean

islands: ArrayList<Island>

students: ArrayList<Student>

allTowers: ArrayList<Tower>

previousTowers: ArrayList<Tower>

Message4Server

sendChooseCh10(ArrayList<Student>, ArrayList<Type_Student>): void

noEntry: boolean

+ changeTower(ArrayList<Tower>): void

+ getInfluence(ArrayList<Type_Student>): int

- head: Island

+ size(): int

head: Island

tower: Tower

towerColor: Colors

ID: int

Message4Server(ObjectOutputStream):

sendChosenCard(AssistantCard): void

sendChooseCh12(Type_Student): void

sendMovedStudent(Student, int): void

sendLogin(String): void

sendStepsMN(int): void

sendChooseCh8(): void

sendChooseCh5(Land): void

sendNumPlayers(int): void

sendChoice(Wizards): void

sendChooseCh2(): void

sendChooseCh4(): void

sendACK(): void

sendPONG(): void

sendNACK(): void

sendChoiceCloud(Cloud): void

sendRegistration(String): void

sendChooseCh11(Student): void

sendChoosingGame(String): void

sendChooseCh1(Student, Land): void

sendNoCh(): void

sendExpertMatch(boolean): void

MatchController

+ MatchController():

- in: ObjectInputStream

- server: Message4Server

+ Ok(ActionEvent): void

show_cloud(): void

+ thelfMethd(): void

+ ch question(): void

+ initialize(): void

show_entry(): void

+ refreshEntry(): void

+ run(): void

+ choose_cloud(MouseEvent): void

+ selectmn(MouseEvent): void

+ moveto(MouseEvent): void

+ usechno(ActionEvent): void

+ use_ch(MouseEvent): void

- show_wizard(ImageView, Wizards): void

+ show_characters(ActionEvent): void

+ columnselected(MouseEvent): void

+ use_assistant(MouseEvent): void

+ go_to_island(ActionEvent): void

+ selectfromentry(MouseEvent): void

+ movetoboard(ActionEvent): void

+ show_noentry(ImageView): void

+ show_boards(ActionEvent): void

visibleAssCards: List<AssistantCard>

+ usechyes(ActionEvent): void

+ setCardChosenbytheother(int, AssistantCard): void

+ setNoEffectColumns(): void

- print_island(int, Land): void

- show_towers(int): void

+ setDropShadow(): void

+ wakeUp(String): void

selected(int, int): void

+ setmatch(Match): void

- show_islands(int): void

+ useit(MouseEvent): void

+ useCh1(int): void

- show(int): void

+ useCh11(): void

+ useCh10(): void

+ useCh5(): void

+ useCh12(): void

disableMN: boolean

clientGui: ClientGui

me: Player

gui: Gui

stateLabel: String

action: Action

match: Match

in: ObjectInputStream

disableChCards: boolean

disableClouds: boolean

disableLands: boolean

server: Message4Server

disableBoards: boolean

disableEntrance: boolean

disableAssistants: boolean

characters: CharacterCard[]

disableColumns: boolean

- character(ImageView, CharacterCard): void

- show_student(ImageView, Student): void

~ characters: Pane

- action: Action

match: Match

- me: Player

- gui: Gui