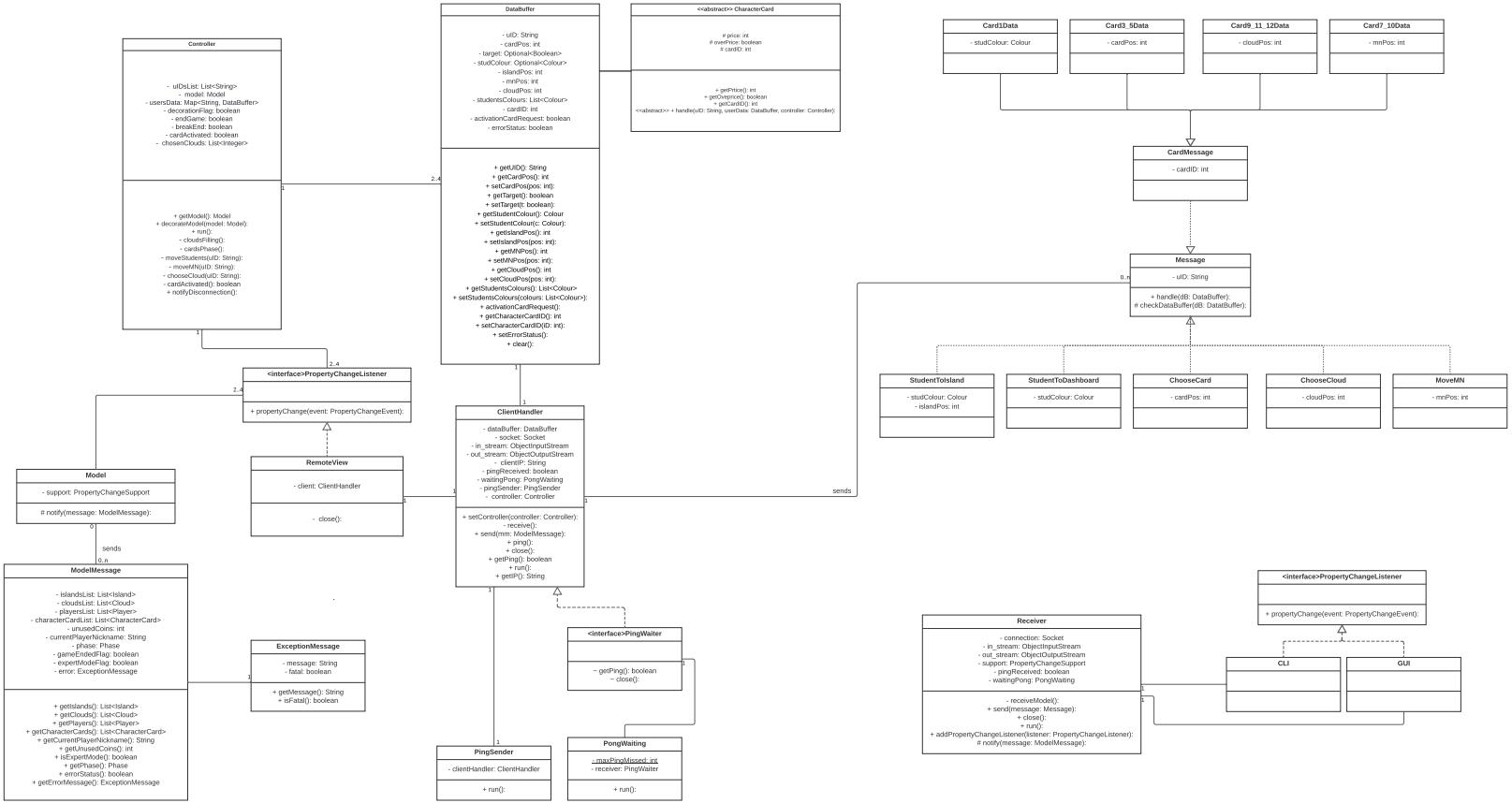
Controller
- uIDsList: List<String>
- model: Model
- usersData: Map<String, DataBuffer>
- decorationFlag: boolean
- endGame: boolean
- breakEnd: boolean
- cardActivated: boolean
- chosenClouds: List<Integer>
+ getModel(): Model
+ decorateModel(model: Model):
+ run():
- cloudsFilling():
- cardsPhase():
- moveStudents(uID: String):
- moveMN(uID: String):
- chooseCloud(uID: String):
- cardActivated(): boolean
+ notifyDisconnection():
DataBuffer
- uID: String
- cardPos: int
- target: Optional<Boolean>
- studColour: Optional<Colour>
- islandPos: int
- mnPos: int
- cloudPos: int
- studentsColours: List<Colour>
- cardID: int
- activationCardRequest: boolean
- errorStatus: boolean
+ getUID(): String
+ getCardPos(): int
+ setCardPos(pos: int):
+ getTarget(t): boolean
+ setTarget(t: boolean):
+ getStudentColour(): Colour
+ setStudentColour(c: Colour):
+ getIslandPos(): int
+ setIslandPos(pos: int):
+ getMNPos(): int
+ setMNPos(pos: int):
+ getCloudPos(): int
+ setCloudPos(pos: int):
+ getStudentsColours(): List<Colour>
+ setStudentsColours(colours: List<Colour>):
+ activationCardRequest():
+ getCharacterCardID(): int
+ setCharacterCardID(ID: int):
+ setErrorStatus():
+ clear():
<<abstract>> CharacterCard
price: int
overPrice: boolean
cardID: int
+ getPrice(): int
+ getOverprice(): boolean
+ getCardID(): int
<<abstract>> + handle(uID: String, userData: DataBuffer, controller: Controller):
Card1Data
- studColour: Colour
Card3_5Data
- cardPos: int
Card9_11_12Data
- cloudPos: int
Card7_10Data
- mnPos: int
CardMessage
- cardID: int
Message
- uID: String
+ handle(dB: DataBuffer):
# checkDataBuffer(dB: DatatBuffer):
StudentToIsland
- studColour: Colour
- islandPos: int
StudentToDashboard
- studColour: Colour
ChooseCard
- cardPos: int
ChooseCloud
- cloudPos: int
MoveMN
- mnPos: int
<interface>PropertyChangeListener
+ propertyChange(event: PropertyChangeEvent):
RemoteView
- client: ClientHandler
- close():
Model
- support: PropertyChangeSupport
notify(message: ModelMessage):
ClientHandler
- dataBuffer: DataBuffer
- socket: Socket
- in_stream: ObjectInputStream
- out_stream: ObjectOutputStream
- clientIP: String
- pingReceived: boolean
- waitingPong: PongWaiting
- pingSender: PingSender
- controller: Controller
+ setController(controller: Controller):
- receive():
+ send(mm: ModelMessage):
+ ping():
+ close():
+ getPing(): boolean
+ run():
+ getIP(): String
sends
ModelMessage
- islandsList: List<Island>
- cloudsList: List<Cloud>
- playersList: List<Player>
- characterCardsList: List<CharacterCard>
- unusedCoins: int
- currentPlayerNickname: String
- phase: Phase
- gameEndedFlag: boolean
- expertModeFlag: boolean
- error: ExceptionMessage
+ getIslands(): List<Island>
+ getClouds(): List<Cloud>
+ getPlayers(): List<Player>
+ getCharacterCards(): List<CharacterCard>
+ getCurrentPlayerNickname(): String
+ getUnusedCoins(): int
+ isExpertMode(): boolean
+ getPhase(): Phase
+ errorStatus(): boolean
+ getErrorMessage(): ExceptionMessage
ExceptionMessage
- message: String
- fatal: boolean
+ getMessage(): String
+ isFatal(): boolean
<interface>PingWaiter
~ getPing(): boolean
~ close():
PingSender
- clientHandler: ClientHandler
+ run():
PongWaiting
- maxPingMissed: int
- receiver: PingWaiter
+ run():
Receiver
- connection: Socket
- in_stream: ObjectInputStream
- out_stream: ObjectOutputStream
- support: PropertyChangeSupport
- pingReceived: boolean
- waitingPong: PongWaiting
- receiveModel():
+ send(message: Message):
+ close():
+ run():
+ addPropertyChangeListener(listener: PropertyChangeListener):
# notify(message: ModelMessage):
<interface>PropertyChangeListener
+ propertyChange(event: PropertyChangeEvent):
CLI
GUI