Server   Client

Controller

<<Interface>> Connection
- num_clients: int
- ip_address: int[]
- move: Move
+setup_connectivity(): boolean
- wait_clients(): void
- start_connections(): void
+ send_data(String): void
+ send_ready(): boolean

<<Interface>> Connection
- num_clients: int
- ip_address: int[]
- move: Move
+setup_connectivity(): boolean
- wait_clients(): void
- start_connections(): void
+ send_data(String): void
+ send_ready(): boolean

Score
- player: Player
- board: GameBoard
+ calculate_score(): int
-public_objective_score(): in
-private_objective_score(): it

TotalScore
- player: ArrayList<Player>
- score: int[]
- connectivity: Connection
- send_total_score(): void

Player_Turn
- player: Player
- connectivity: Connection
- move: Move
- get_Move(): Move
- apply_effects():void

Turn
- players: ArrayList <Player>
-connectivity: Connection
- turn: int
- next_player_turn(): void
- end_turn(): void

Controller
-model: Model
-players: ArrayList <Player>
-dice: DiceBag
-board: GameBoard
-connectivity: Connection
- Setup_Board(): GameBoard
- Setup_Players(): ArrayList<Player>
- Setup_Connection(): Connection
- send_ready(): void
- end_game(): void

<<Interface>> Server_Stub_RMI

Server_Socket
- socket: Socket[]
-current_move: Move
+ Get_Data(): String
+ Get_Move(): Move
-

Server_RMI
- rmi: Server[]

Move
- die:Die
- tool: ToolCard
+ getDie(): Die
+ getToolCard: ToolCard

<<Interface>> Client_Stub_RMI

Client_Socket
- socket: Socket[]
-current_move: Move
+ Get_Data(): String
+ Get_Move(): Move
-

Client_RMI
- rmi: Server[]

Validity_Check
- board: WindowBoard
- move_check(): boolean

Move
- die: Die
- tool: ToolCard
- validity_check(): boolean
- send_data(): void

Event_Handler
- components: JComponent

View

GUI
- components: JComponents[]

CLI

Model

Die
- color: String
- value: int
+ draftDie(): int
+ reRollDie():int
+ increaseValue(): void
+ decreaseValue(): void

RoundTrack
- set: Arraylist<Die>
+ insertDie():

DraftPool
- set: Arraylist<Die>
+ reInsert()
+ reRollAll()
+ swapDieToRoundTrack()
+ removeDie()

DiceBag
- set: Arraylist<Die>
+ pullOutFromBag() Die
+ shuffleDiceBag(): void

GameBoard
- diceBag : DiceBag
- roundTrack: RoundTrack
- privateCardDeck: Deck
- publicCardDeck: Deck
- toolcardDeck: Deck
+

PlayerBoard
- color: String
- window: WindowPatternCard
+ insertWindow():
+ moveDieInWindow()
+ checkIntegrity()

WindowPatternDeck
- set: ArrayList<WindowPatternCard>
+ shuffleDeck(): void
+ takeCard(): void

WindowPatternCard
- name: String
- grid: ArrayList<Cell>
- difficulty: int
+ fillWindow():
+ getDifficulty():

Cell
- color: String
- shade: int
- die: Die
+ addCell():
+ checkCell():

Player
- playerNickname: String
- playerBoard: PlayerBoard
- privateObjective: PrivateObjectiveCard
- favorToken:int
+ moveToken()

Model
- players: Arraylist<Player>
- gameBoard: GameBoard
+

Toolcard
- number: int
- color: String
- favorToken: int
- effect: Effect
+ effectOn(): void
+ addToken():

(Abstract) Effect
+ applyEffect(): void

GrozingPliers
+ applyEffect(): void

TapWheel
+ applyEffect(): void

FluxRemover
+ applyEffect(): void

EglomiseBrush
+ applyEffect(): void

CopperFoilBrunisher
+ applyEffect(): void

Lathekin
+ applyEffect(): void

LensCutter
+ applyEffect(): void

FluxBrush
+ applyEffect(): void

GlazingHammer
+ applyEffect(): void

RunningPliers
+ applyEffect(): void

GrindingStone
+ applyEffect(): void

CorkBackedStraighedge
+ applyEffect(): void

Deck
- cardList: ArrayList<Card>
+ shuffleDeck():

(Abstract) Card
- name: String
- description: String
+

(Abstract) PublicObjectiveCard
- score: int
+ returnScore(PlayerBoard): int

PrivateObjectiveCard
- color: String
- score: int
+returnScore(): int

MediumShades
+returnScore():

ColorDiagonals
+returnScore():

RowColorVariety
+returnScore():

ColumnColorVariety
+returnScore():

DeepShades
+returnScore():

LightShades
+returnScore():

ShadeVariety
+returnScore():

RowShadeVariety
+returnScore():

ColumnShadeVariety
+returnScore():

ColorVariety
+returnScore():