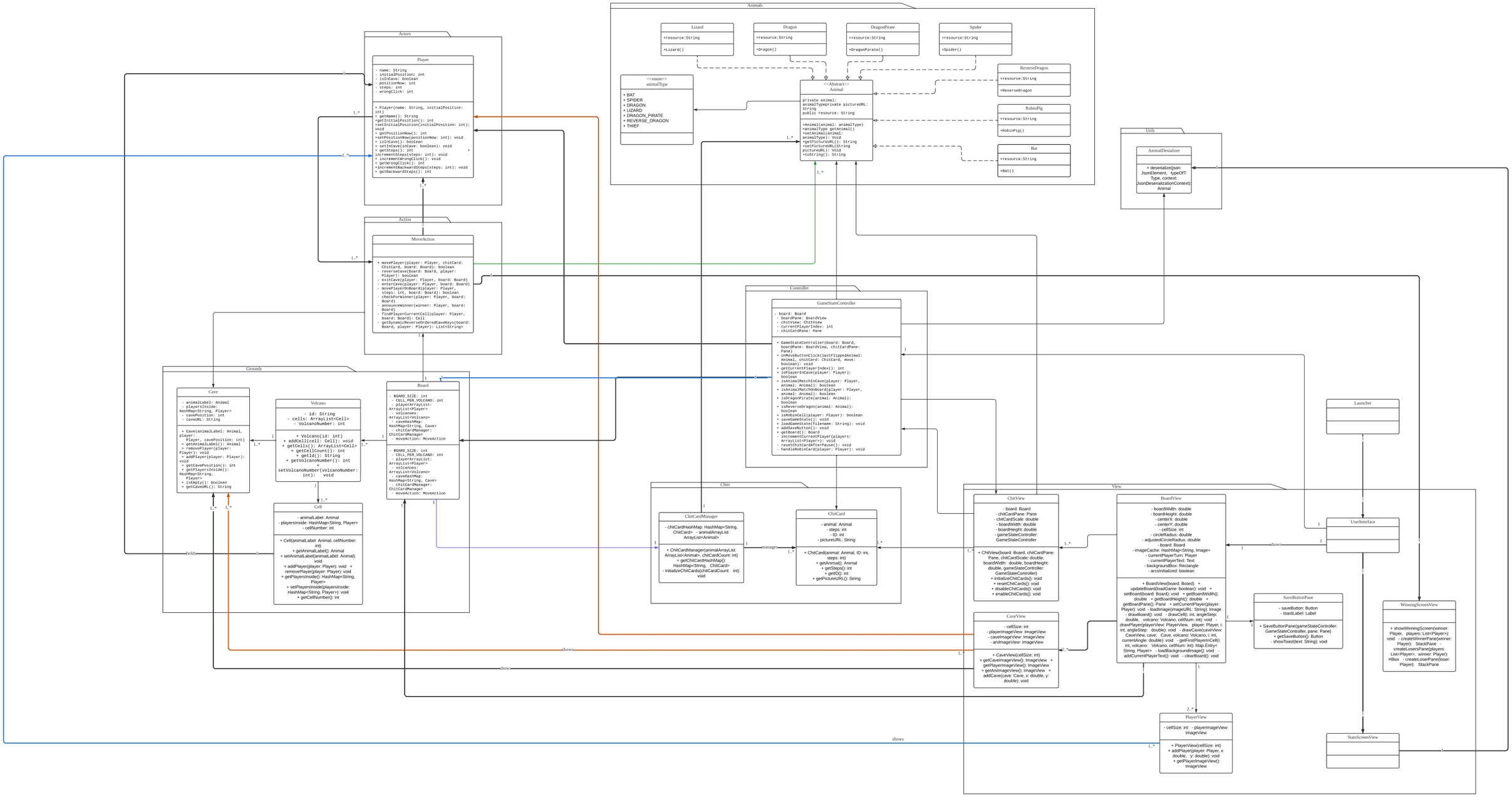
Actors
Player
- name: String
- initialPosition: int
- isInCave: boolean
- positionNow: int
- steps: int
- wrongClick: int
+ Player(name: String, initialPosition: int)
+ getName(): String
+getInitialPosition(): int
+setInitialPosition(initialPosition: int): void
+ getPositionNow(): int
+setPositionNow(positionNow: int): void
+ isInCave(): boolean
+ setInCave(inCave: boolean): void
+ getSteps(): int
incrementSteps(steps: int): void
+ incrementWrongClick(): void
+ getWrongClick(): int
+incrementBackwardSteps(steps: int): void
+ getBackwardSteps(): int
Action
MoveAction
+ movePlayer(player: Player, chitCard: ChitCard, board: Board): boolean
- reverseCave(board: Board, player: Player): boolean
- exitCave(player: Player, board: Board)
- enterCave(player: Player, board: Board)
- movePlayerOnBoard(player: Player, steps: int, board: Board): boolean
- checkForWinner(player: Player, board: Board)
- announceWinner(winner: Player, board: Board)
- findPlayerCurrentCell(player: Player, board: Board): Cell
- getDynamicReverseOrderedCaveKeys(board: Board, player: Player): List<String>
Animals
Lizard
+resource:String
+Lizard()
Dragon
+resource:String
+Dragon()
DragonPirate
+resource:String
+DragonPirate()
Spider
+resource:String
+Spider()
<<enum>>
animalType
+ BAT
+ SPIDER
+ DRAGON
+ LIZARD
+ DRAGON_PIRATE
+ REVERSE_DRAGON
+ THIEF
<<Abstract>>
Animal
private animal: animalTypeprivate pictureURL: String
public resource: String
+Animal(animal: animalType)
+animalType getAnimal()
+setAnimal(animal: animalType): Void
+getPictureURL(): String
+setPictureURL(String pictureURL): Void
+toString(): String
ReverseDragon
+resource:String
+ReverseDragon
RobinPig
+resource:String
+RobinPig()
Bat
+resource:String
+Bat()
Utils
AnimalDesializer
+ deserialize(json, JsonElement, typeOfT: Type, context: JsonDeserializationContext): Animal
Controller
GameStateController
- board: Board
- boardPane: BoardView
- chitView: ChitView
- currentPlayerIndex: int
- chitCardPane: Pane
+ GameStateController(board: Board, boardPane: BoardView, chitCardPane: Pane)
+ onMoveButtonClick(lastFlippedAnimal: Animal, chitCard: ChitCard, move: boolean): void
+ getCurrentPlayerIndex(): int
+ isPlayerInCave(player: Player): boolean
+ isAnimalMatchInCave(player: Player, animal: Animal): boolean
+ isAnimalMatchOnBoard(player: Player, animal: Animal): boolean
+ isDragonPirate(animal: Animal): boolean
+ isReverseDragon(animal: Animal): boolean
+ isRobinCell(player: Player): boolean
+ saveGameState(): void
+ loadGameState(filename: String): void
+ addSaveButton(): void
+ getBoard(): Board
increementCurrentPlayer(players: ArrayList<Player>): void
- resetChitCardAfterPause(): void
- handleRobinCard(player: Player): void
Grounds
Cave
- animalLabel: Animal
- playersInside: HashMap<String, Player>
- cavePosition: int
- caveURL: String
+ Cave(animalLabel: Animal, player: Player, cavePosition: int)
+ getAnimalLabel(): Animal
+ removePlayer(player: Player): void
+ addPlayer(player: Player): void
+ getCavePosition(): int
+ getPlayersInside(): HashMap<String, Player>
+ isEmpty(): boolean
+ getCaveURL(): String
Volcano
- id: String
- cells: ArrayList<Cell>
- VolcanoNumber: int
+ Volcano(id: int)
+ addCell(cell: Cell): void
+ getCells(): ArrayList<Cell>
+ getCellCount(): int
+ getId(): String
+ getVolcanoNumber(): int
+ setVolcanoNumber(volcanoNumber: int): void
Board
- BOARD_SIZE: int
- CELL_PER_VOLCANO: int
- playerArrayList: ArrayList<Player>
- volcanoes: ArrayList<Volcano>
- caveHashMap: HashMap<String, Cave>
- chitCardManager: ChitCardManager
- moveAction: MoveAction
- BOARD_SIZE: int
- CELL_PER_VOLCANO: int
- playerArrayList: ArrayList<Player>
- volcanoes: ArrayList<Volcano>
- caveHashMap: HashMap<String, Cave>
- chitCardManager: ChitCardManager
- moveAction: MoveAction
Cell
- animalLabel: Animal
- playersInside: HashMap<String, Player>
- cellNumber: int
+ Cell(animalLabel: Animal, cellNumber: int)
+ getAnimalLabel(): Animal
+ setAnimalLabel(animalLabel: Animal): void
+ addPlayer(player: Player): void
+ removePlayer(player: Player): void
+ getPlayersInside(): HashMap<String, Player>
+ setPlayersInside(playersInside: HashMap<String, Player>): void
+ getCellNumber(): int
Chits
ChitCardManager
- chitCardHashMap: HashMap<String, ChitCard>
- animalArrayList: ArrayList<Animal>
+ ChitCardManager(animalArrayList: ArrayList<Animal>, chitCardCount: int)
+ getChitCardHashMap(): HashMap<String, ChitCard>
- initializeChitCards(chitCardCount: int): void
ChitCard
- animal: Animal
- steps: int
- ID: int
- pictureURL: String
+ ChitCard(animal: Animal, ID: int, steps: int)
+ getAnimal(): Animal
+ getSteps(): int
+ getID(): int
+ getPictureURL(): String
View
ChitView
- board: Board
- chitCardPane: Pane
- chitCardScale: double
- boardWidth: double
- boardHeight: double
- gameStateController: GameStateController
+ ChitView(board: Board, chitCardPane: Pane, chitCardScale: double, boardWidth: double, boardHeight: double, gameStateController: GameStateController)
+ initializeChitCards(): void
+ resetChitCards(): void
+ disableChitCards(): void
+ enableChitCards(): void
BoardView
- boardWidth: double
- boardHeight: double
- centerX: double
- centerY: double
- cellSize: int
- circleRadius: double
- adjustedCircleRadius: double
- board: Board
- imageCache: HashMap<String, Image>
- currentPlayerTurn: Player
- currentPlayerText: Text
- backgroundBox: Rectangle
- arcsInitialized: boolean
+ BoardView(board: Board) + updateBoard(loadGame: boolean): void + setBoard(board: Board): void + getBoardWidth(): double + getBoardHeight(): double + getBoardPane(): Pane + setCurrentPlayer(player: Player): void - loadImage(imageURL: String): Image - drawBoard(): void - drawCell(i: int, angleStep: double, volcano: Volcano, cellNum: int): void - drawPlayer(playerView: PlayerView, player: Player, i: int, angleStep: double): void - drawCave(caveView: CaveView, cave: Cave, volcano: Volcano, i: int, currentAngle: double): void - getFirstPlayerInCell(): int, volcano: Volcano, cellNum: int): Map.Entry<String, Player> - loadBackgroundImage(): void - addCurrentPlayerText(): void - clearBoard(): void
CaveView
- cellSize: int
- playerImageView: ImageView
- caveImageView: ImageView
- aniImageView: ImageView
+ CaveView(cellSize: int)
+ getCaveImageView(): ImageView + getPlayerImageView(): ImageView
+ getAniImageView(): ImageView + addCave(cave: Cave, x: double, y: double): void
PlayerView
- cellSize: int - playerImageView: ImageView
+ PlayerView(cellSize: int)
+ addPlayer(player: Player, x: double, y: double): void
+ getPlayerImageView(): ImageView
Launcher
UserInterface
SaveButtonPane
- saveButton: Button
- toastLabel: Label
+ SaveButtonPane(gameStateController: GameStateController, pane: Pane)
+ getSaveButton(): Button
- showToast(text: String): void
WinningScreenView
+ showWinningScreen(winner: Player, players: List<Player>): void - createWinnerPane(winner: Player): StackPane - createLosersPane(players: List<Player>, winner: Player): HBox - createLoserPane(loser: Player): StackPane
StatsScreenView
hold
manages
shows
shows
shows
shows
1
1..*
2..*