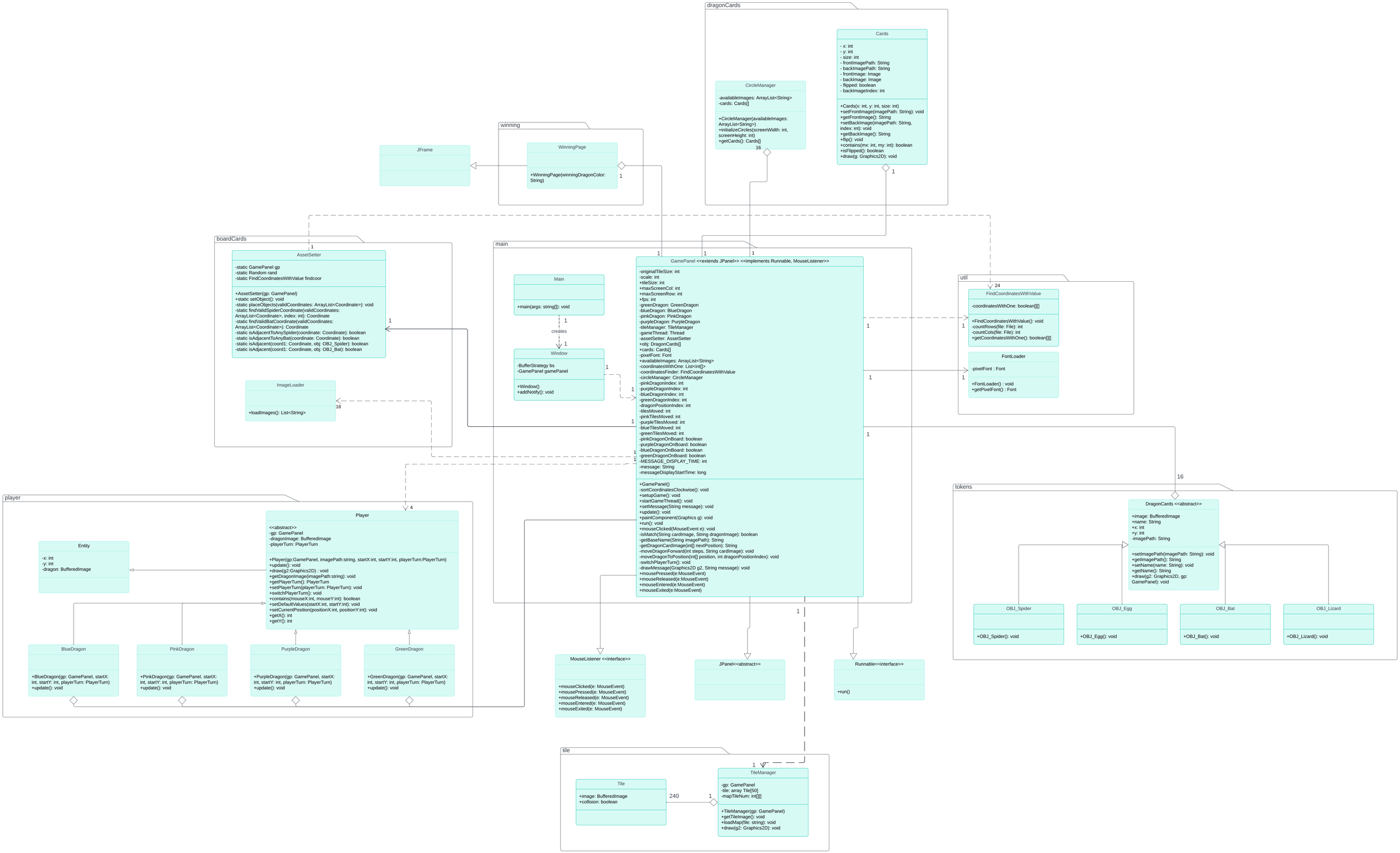
dragonCards
Cards
- x: int
- y: int
- size: int
- frontImagePath: String
- backImagePath: String
- frontImage: Image
- backImage: Image
- flipped: boolean
- backImageIndex: int
+Cards(x: int, y: int, size: int)
+setFrontImage(imagePath: String): void
+getFrontImage(): String
+setBackImage(imagePath: String, index: int): void
+getBackImage(): String
+flip(): void
+contains(mx: int, my: int): boolean
+isFlipped(): boolean
+draw(g: Graphics2D): void
CircleManager
-availableImages: ArrayList<String>
-cards: Cards[]
+CircleManager(availableImages: ArrayList<String>)
+initializeCircles(screenWidth: int, screenHeight: int)
+getCards(): Cards[]
16
1
winning
JFrame
WinningPage
+WinningPage(winningDragonColor: String)
1
boardCards
AssetSetter
-static GamePanel gp
-static Random rand
-static FindCoordinatesWithValue findcoor
+AssetSetter(gp: GamePanel)
+static setObject(): void
-static placeObjects(validCoordinates: ArrayList<Coordinate>): void
-static findValidSpiderCoordinate(validCoordinates: ArrayList<Coordinate>, index: int): Coordinate
-static findValidBatCoordinate(validCoordinates: ArrayList<Coordinate>): Coordinate
-static isAdjacentToAnySpider(coordinate: Coordinate): boolean
-static isAdjacentToAnyBat(coordinate: Coordinate): boolean
-static isAdjacent(coord1: Coordinate, obj: OBJ_Spider): boolean
-static isAdjacent(coord1: Coordinate, obj: OBJ_Bat): boolean
ImageLoader
+loadImages(): List<String>
16
1
main
Main
+main(args: string[]): void
1
creates
1
Window
-BufferStrategy bs
-GamePanel gamePanel
+Window()
+addNotify(): void
GamePanel <<extends JPanel>> <<implements Runnable, MouseListener>>
-originalTileSize: int
-scale: int
+tileSize: int
+maxScreenCol: int
+maxScreenRow: int
+fps: int
-greenDragon: GreenDragon
-blueDragon: BlueDragon
-pinkDragon: PinkDragon
-purpleDragon: PurpleDragon
-tileManager: TileManager
-gameThread: Thread
-assetSetter: AssetSetter
+obj: DragonCards[]
+cards: Cards[]
-pixelFont: Font
+availableImages: ArrayList<String>
-coordinatesWithOne: List<int[]>
-coordinatesFinder: FindCoordinatesWithValue
-circleManager: CircleManager
-pinkDragonIndex: int
-purpleDragonIndex: int
-blueDragonIndex: int
-greenDragonIndex: int
-dragonPositionIndex: int
-tilesMoved: int
-pinkTilesMoved: int
-purpleTilesMoved: int
-blueTilesMoved: int
-greenTilesMoved: int
-pinkDragonOnBoard: boolean
-purpleDragonOnBoard: boolean
-blueDragonOnBoard: boolean
-greenDragonOnBoard: boolean
-MESSAGE_DISPLAY_TIME: int
-message: String
-messageDisplayStartTime: long
+GamePanel()
-sortCoordinatesClockwise(): void
+setupGame(): void
+startGameThread(): void
+setMessage(String message): void
+update(): void
+paintComponent(Graphics g): void
+run(): void
+mouseClicked(MouseEvent e): void
-isMatch(String cardImage, String dragonImage): boolean
-getBaseName(String imagePath): String
-getDragonCardImage(int[] nextPosition): String
-moveDragonForward(int steps, String cardImage): void
-moveDragonToPosition(int[] position, int dragonPositionIndex): void
-switchPlayerTurn(): void
-drawMessage(Graphics2D g2, String message): void
+mousePressed(e:MouseEvent)
+mouseReleased(e:MouseEvent)
+mouseEntered(e:MouseEvent)
+mouseExited(e:MouseEvent)
util
24
FindCoordinatesWithValue
-coordinatesWithOne: boolean[][]
+FindCoordinatesWithValue(): void
-countRows(file: File): int
-countCols(file: File): int
+getCoordinatesWithOne(): boolean[][]
FontLoader
-pixelFont : Font
+FontLoader() : void
+getPixelFont() : Font
1
player
4
Entity
-x: int
-y: int
-dragon: BufferedImage
Player
<<abstract>>
-gp: GamePanel
-dragonImage: BufferedImage
-playerTurn: PlayerTurn
+Player(gp:GamePanel, imagePath:string, startX:int, startY:int, playerTurn:PlayerTurn)
+update(): void
+draw(g2:Graphics2D) : void
+getDragonImage(imagePath:string): void
+getPlayerTurn(): PlayerTurn
+setPlayerTurn(playerTurn: PlayerTurn): void
+switchPlayerTurn(): void
+contains(mouseX:int, mouseY:int): boolean
+setDefaultValues(startX:int, startY:int): void
+setCurrentPosition(positionX:int, positionY:int): void
+getX(): int
+getY(): int
BlueDragon
+BlueDragon(gp: GamePanel, startX: int, startY: int, playerTurn: PlayerTurn)
+update(): void
PinkDragon
+PinkDragon(gp: GamePanel, startX: int, startY: int, playerTurn: PlayerTurn)
+update(): void
PurpleDragon
+PurpleDragon(gp: GamePanel, startX: int, startY: int, playerTurn: PlayerTurn)
+update(): void
GreenDragon
+GreenDragon(gp: GamePanel, startX: int, startY: int, playerTurn: PlayerTurn)
+update(): void
MouseListener <<interface>>
+mouseClicked(e: MouseEvent)
+mousePressed(e: MouseEvent)
+mouseReleased(e: MouseEvent)
+mouseEntered(e: MouseEvent)
+mouseExited(e: MouseEvent)
JPanel<<abstract>>
Runnable<<interface>>
+run()
tokens
16
DragonCards <<abstract>>
+image: BufferedImage
+name: String
+x: int
+y: int
-imagePath: String
+setImagePath(imagePath: String): void
+getImagePath(): String
+setName(name: String): void
+getName(): String
+draw(g2: Graphics2D, gp: GamePanel): void
OBJ_Spider
+OBJ_Spider(): void
OBJ_Egg
+OBJ_Egg(): void
OBJ_Bat
+OBJ_Bat(): void
OBJ_Lizard
+OBJ_Lizard(): void
tile
1
Tile
+image: BufferedImage
+collision: boolean
240
1
TileManager
-gp: GamePanel
-tile: array Tile[50]
-mapTileNum: int[][]
+TileManager(gp: GamePanel)
+getTileImage(): void
+loadMap(file: string): void
+draw(g2: Graphics2D): void