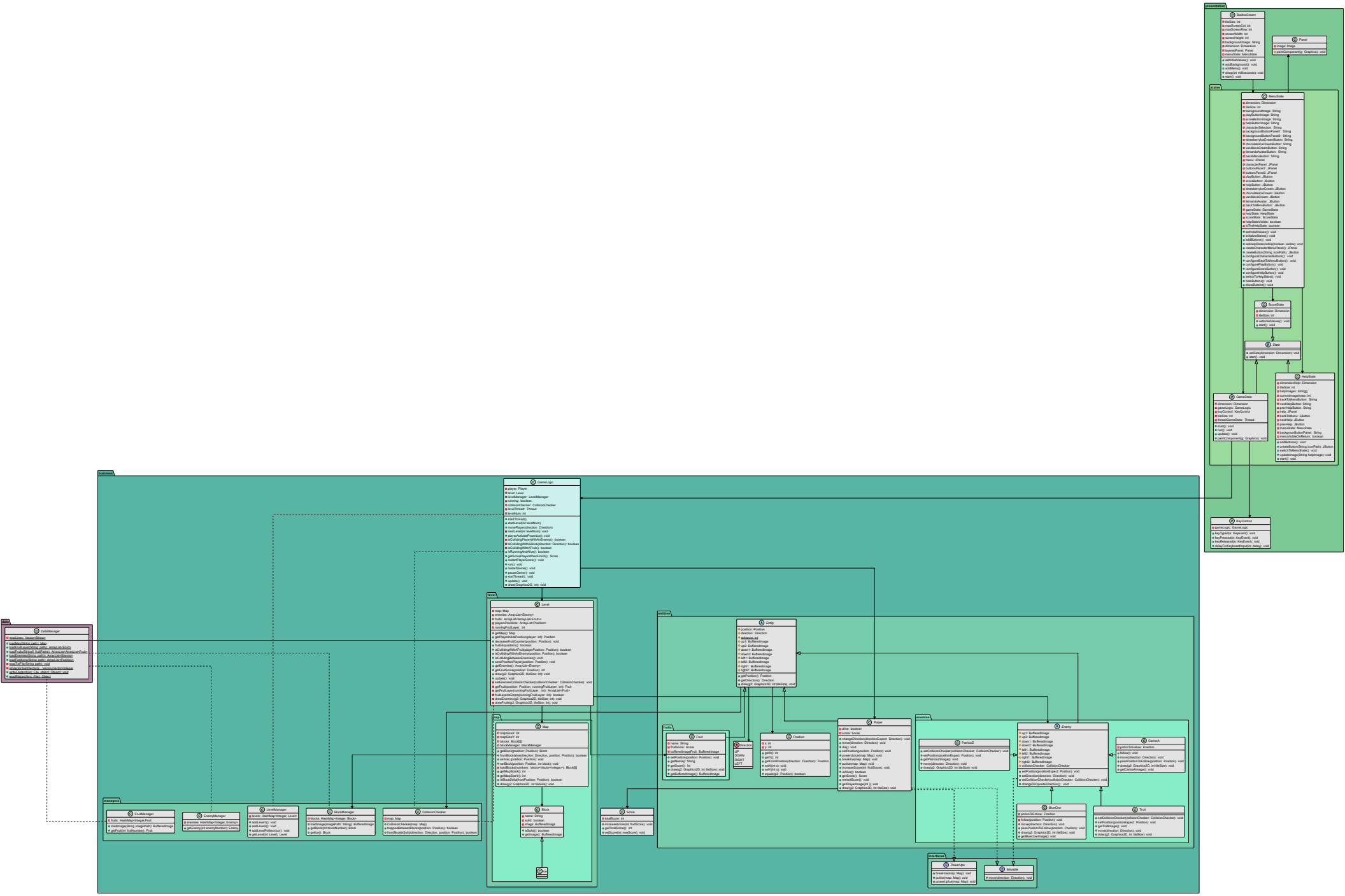
presentation
BadIceCream
- tileSize: int
- maxScreenCol: int
- maxScreenRow: int
- screenWidth: int
- screenHeight: int
- backgroundImage: String
- dimension: Dimension
- layeredPanel: JPanel
- menuState: MenuState
+ setInitialValues() : void
+ addBackground() : void
+ addMenu() : void
+ sleep(int milliseconds) : void
+ start() : void
Panel
- image: Image
+ paintComponent(g: Graphics) : void
states
MenuState
- dimension: Dimension
- tileSize: int
- backgroundImage: String
- playButtonImage: String
- scoreButtonImage: String
- helpButtonImage: String
- characterSelection: String
- backgroundButtonPanel1: String
- backgroundButtonPanel2: String
- strawberryIceCreamButton: String
- chocolateIceCreamButton: String
- vanillaIceCreamButton: String
- fernandoAvatarButton: String
- backToMenuButton: String
- menu: JPanel
- characterPanel: JPanel
- buttonsPanel1: JPanel
- buttonsPanel2: JPanel
- playButton: JButton
- scoreButton: JButton
- helpButton: JButton
- strawberryIceCream: JButton
- chocolateIceCream: JButton
- vanillaIceCream: JButton
- fernandoAvatar: JButton
- backToMenuButton: JButton
- currentState: GameState
- helpState: HelpState
- scoreState: ScoreState
- helpStateVisible: boolean
- isTheHelpState: boolean
+ setInitialValues() : void
+ addMenu() : void
+ addButtons() : void
+ setHelpStateVisible(boolean visible) : void
+ createCharacterMenuPanel() : JPanel
+ createButton(String iconPath) : JButton
+ configureCharacterButtons() : void
+ configureBackToMenuButton() : void
+ configurePlayButton() : void
+ configureScoreButton() : void
+ configureHelpButton() : void
+ switchToHelpState() : void
+ initialize() : void
+ showButtons() : void
ScoreState
- dimension: Dimension
- tileSize: int
+ setInitialValues() : void
+ start() : void
State
+ setSize(dimension: Dimension) : void
+ start() : void
GameState
- dimension: Dimension
- gameLogic: GameLogic
- keyControl: KeyControl
- tileSize: int
- threadGameState: Thread
+ start() : void
+ run() : void
+ update() : void
+ paintComponent(g: Graphics) : void
HelpState
- dimensionImage: Dimension
- tileSize: int
- helpImages: String[]
- currentImageIndex: int
- backToMenuButton: String
- nextHelpButton: String
- previousHelpButton: String
- help: JPanel
- backToMenu: JButton
- nextHelp: JButton
- previousHelp: JButton
- mainState: MenuState
- backToMenuButtonPath: String
- nextHelpButtonDefault: boolean
+ addButtons() : void
+ createButton(String iconPath) : JButton
+ switchToMenuState() : void
+ updateImage(String helpImage) : void
+ start() : void
KeyControl
- gameLogic: GameLogic
+ keyTyped(e: KeyEvent) : void
+ keyPressed(e: KeyEvent) : void
+ keyReleased(e: KeyEvent) : void
+ delayForKeyboardInput(int delay) : void
business
GameLogic
- player: Player
- level: Level
- levelManager: LevelManager
- running: boolean
- collisionChecker: CollisionChecker
- levelThread: Thread
- levelNum: int
+ startThread()
+ startLevel(int levelNum)
+ movePlayer(direction: Direction)
+ nextLevel(int levelNum) : void
+ playerActivatePowerUp() : void
+ isCollidingPlayerWithAnEnemy() : boolean
+ isCollidingWithABlock(direction: Direction): boolean
+ isCollidingWithAFruit() : boolean
+ isRunningInAKey() : boolean
+ getScorePlayerWhenFinish() : Score
+ restartPlayerScore() : void
+ run() : void
+ restartGame() : void
+ pauseGame() : void
+ startThread() : void
+ update() : void
+ draw(Graphics2D, int) : void
level
Level
- map: Map
- enemies: ArrayList<Enemy>
- fruits: ArrayList<ArrayList<Fruit>>
- playersPositions: ArrayList<Position>
- totalingFruitLayer: int
+ setMap() : Map
+ getPlayersInitialPositions(player: int): Position
+ decreaseFruitCounter(position: Position) : void
+ fruitEqualZero() : boolean
+ isCollidingWithFruitAplayerPosition: Position): boolean
+ isCollidingWithAnEnemy(position: Position): boolean
+ isCollidingWithEnemiesEnemies() : void
+ nextPositionPlayer(position: Position ) : void
+ getEnemies() : ArrayList<Enemy>
+ getFruitScore(position: Position): int
+ draw(g2: Graphics2D, tileSize: int) : void
+ update() : void
+ setEnemiesCollisionChecker(collisionChecker: CollisionChecker): void
+ getFruitposition: Position, runningFruitLayer: int): Fruit
+ getFruitsPositions(fruits: ArrayList<ArrayList<Fruit>>
+ fullLayerEmptyIncreasingFruitLayer: int): boolean
+ drawEnemies(g2: Graphics2D, tileSize: int): void
+ drawFruits(g2: Graphics2D, tileSize: int): void
map
Map
- mapSizeX: int
- mapSizeY: int
- blocks: Block[][]
- blockManager: BlockManager
+ getBlock(position: Position): Block
+ frontBlockIsSolid(direction: Direction, position: Position): boolean
+ setInit(position: Position): void
+ setBlock(position: Position, int block): void
+ loadBlocks(numbers: Vector<Vector<Integer>> Block[][]
+ getMapSizeX() : int
+ getMapSizeY() : int
+ isBlockSolid(getPosition: Position): boolean
+ draw(g2: Graphics2D, int tileSize): void
Block
- name: String
- solid: boolean
- image: BufferedImage
+ isSolid() : boolean
+ getImage() : BufferedImage
Void
entities
Entity
- position: Position
- direction: Direction
- speed: int
- up1: BufferedImage
- up2: BufferedImage
- down1: BufferedImage
- down2: BufferedImage
- left1: BufferedImage
- left2: BufferedImage
- right1: BufferedImage
- right2: BufferedImage
+ getPosition() : Position
+ getDirection() : Direction
+ draw(g2: Graphics2D, int tileSize): void
fruits
Fruit
- name: String
- fruitScore: Score
- bufferedImageFruit: BufferedImage
+ setPosition(position: Position): void
+ getName() : String
+ getScore() : int
+ draw(g2: Graphics2D, int tileSize): void
+ getBufferedImage() : BufferedImage
Direction
UP
DOWN
RIGHT
LEFT
Position
- x: int
- y: int
+ getX() : int
+ getY() : int
+ getFromPosition(direction: Direction): Position
+ setX(int x): void
+ setY(int y): void
+ equals(p2: Position): boolean
Player
- alive: boolean
- score: Score
+ changeCheckerDirectionExpect: Direction): void
+ move(direction: Direction): void
+ die() : void
+ setPosition(position: Position): void
+ powerUp(map: Map): void
+ break(map: Map): void
+ put(map: Map): void
+ increaseScore(int fruitScore): void
+ alive() : boolean
+ getScore() : Score
+ setScore(Score)
+ getPlayerImage(int i): void
+ draw(g2: Graphics2D, int tileSize): void
enemies
PatrolZ
+ setCollisionChecker(collisionChecker: CollisionChecker): void
+ setPositionpositionExpect: Position): void
+ getPatrolzImage() : void
+ move(direction: Direction): void
+ draw(g2: Graphics2D, int tileSize): void
Enemy
- up1: BufferedImage
- up2: BufferedImage
- down1: BufferedImage
- down2: BufferedImage
- left1: BufferedImage
- left2: BufferedImage
- right1: BufferedImage
- right2: BufferedImage
- collisionChecker: CollisionChecker
+ setPositionpositionExpect: Position): void
+ setDirection(direction: Direction): void
+ setCollisionChecker(collisionChecker: CollisionChecker): void
+ changeToOppositeDirection() : void
CartoonA
- portionToFollow: Position
+ follow() : void
+ move(direction: Direction): void
+ pressPositionToFollow(position: Position): void
+ draw(g2: Graphics2D, int tileSize): void
+ getCartoonImage() : void
BlueCow
- portionToFollow: Position
+ follow(position: Position): void
+ move(direction: Direction): void
+ pressPositionToFollow(position: Position): void
+ draw(g2: Graphics2D, int tileSize): void
+ getBlueCowImage() : void
Troll
+ setCollisionChecker(collisionChecker: CollisionChecker): void
+ setPositionpositionExpect: Position): void
+ getTrollImage() : void
+ move(direction: Direction): void
+ draw(g2: Graphics2D, int tileSize): void
managers
FruitManager
- fruits: HashMap<Integer,Fruit>
+ loadImage(String imagePath): BufferedImage
+ getFruit(int fruitNumber): Fruit
EnemyManager
- enemies: HashMap<Integer, Enemy>
+ getEnemy(int enemyNumber): Enemy
LevelManager
- levels: HashMap<Integer, Level>
+ addLevel() : void
+ addLevel() : void
+ addLevelPolitecico() : void
+ getLevel(int Level): Level
BlockManager
- blocks: HashMap<Integer, Block>
+ loadImage(imagePath: String): BufferedImage
+ getBlock(int blockNumber): Block
+ getIce() : Block
CollisionChecker
- map: Map
+ CollisionChecker(map: Map)
+ hippoBehaveBlockisSolid(position: Position): boolean
+ frontBlockIsSolid(direction: Direction, position: Position): boolean
Score
- totalScore: int
+ increaseScore(int fruitScore): void
+ getTotalScore() : int
+ setScore(int newScore): void
interfaces
PowerUps
+ break(map: Map): void
+ put(map: Map): void
+ power(pr(map: Map): void
Movable
+ move(direction: Direction): void
data
DataManager
+ mapLines: Vector<String>
+ loadMapString(path): Map
+ loadFruitLayer(String path): ArrayList<Fruit>
+ loadFruitsStringA: fruitPaths): ArrayList<ArrayList<Fruit>>
+ loadPositionsString(path): ArrayList<String>
+ loadTotalFruitsString(path): ArrayList<Position>
+ getFruitsPositions(): ArrayList<Position>
+ getVectorTotalVector(): Vector<Vector<String>>
+ writeTo(section: File, object: Object): void
+ readSection(section: File): Object