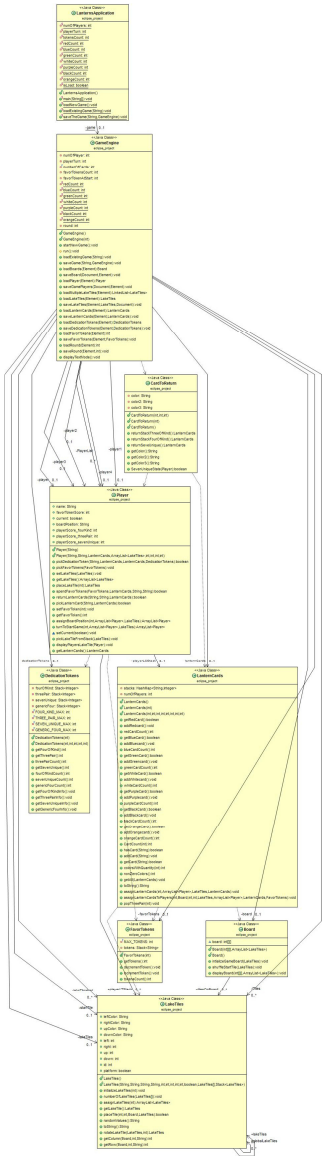
«Java Class»
GameApplication
mtree_main
- numOfPlayers: int
- playerTurn: int
- numberOfTurns: int
- favorTokensCount: int
- redCount: int
- blueCount: int
- greenCount: int
- whiteCount: int
- purpleCount: int
- blackCount: int
- goldCount: int
- fileLoad: boolean
+ GameApplication()
+ newString(): void
+ newNewGame(): void
+ loadExistingGame(String): void
+ saveTheGame(String, GameEngine): void
game 1..1
«Java Class»
GameEngine
mtree_main
- numOfPlayers: int
- playerTurn: int
- /numberOfTurns: int
- favorTokensCount: int
- favorTokensStart: int
- redCount: int
- blueCount: int
- greenCount: int
- whiteCount: int
- purpleCount: int
- blackCount: int
- goldCount: int
- round: int
+ GameEngine()
+ GameEngine(int)
+ startNewGame(): void
+ run(): void
+ loadExistingGame(String): void
+ saveGame(String,GameEngine): void
+ loadBoardElement(Element): void
+ loadCardsDeckElement(Element): void
+ loadPlayer(Element):Player
+ saveGamePlayers(Document,Element): void
+ loadStatusLabelTileElement(LinkedLabelTiles)
+ loadLabelTileElement():LabelTiles
+ saveLabelTileElement(LabelTiles,Document): void
+ loadLanternCardsElement():LanternCards
+ saveLanternCardsElement(LanternCards): void
+ loadDedicatedTokensElement(DedicatedTokens
+ saveDedicatedTokensElement(DedicatedTokens): void
+ loadFavorTokensElement()
+ saveFavorTokensElement(FavorTokens): void
+ loadBoardElement()
+ saveBoard(Board): void
+ displayBoard(): void
«Java Class»
CardTableau
mtree_main
+ color: String
+ cardID: String
+ cost: int
+ CardTableau(int,int,int)
+ CardTableau(int)
+ CardTableau()
+ returnStackTrueOfKind(): LanternCards
+ returnStackFalseOfKind(): LanternCards
+ returnSelectedInput(): LanternCards
+ getColor(): String
+ getCardID(): String
+ getCost(): int
+ getSelectedCardStatus(Player): boolean
player2 0..1
player1 0..1
player3 0..1
player4 0..1
«Java Class»
Player
mtree_main
+ name: String
+ favorTokensScore: int
+ current: boolean
+ boardPosition: String
+ placeScore_fourthest: int
+ placeScore_threeRear: int
+ placeScore_secondnext: int
+ Player(String)
+ setCardsDeck(String,LanternCards,Array,LanternCards,Array,int,int,int)
+ setDedicatedToken(String,LanternCards,LanternCards,DedicatedTokens):boolean
+ setFavorTokensFavorTokens): void
+ setLabelTiles(LabelTiles): void
+ getLabelTiles():Array,LabelTiles>
+ placeLabelTile(LabelTile)
+ spendFavorTokens(FavorTokens,LanternCards,String,String):boolean
+ setCardsDeck(String,LanternCards):boolean
+ setLanternCard(String,LanternCards):boolean
+ setFavorTokens(int)
+ getFavorTokens():int
+ assignBoardPosition(int,Array),int>Player>,LabelTiles,Array,int>Player>
+ turnToNextSpecial(Array,int>Player>,LabelTiles,Array,int>Player>
+ setCurrentStatus(): void
+ displayLabelTiles(LabelTiles): void
+ displayPlayerLabelTile(Player): void
+ getLanternCards():LanternCards
dedicatedTokens 0..1
«Java Class»
DedicatedTokens
mtree_main
+ favorWhite: Stack<Integer>
+ threeRar: Stack<Integer>
+ secondnext: Stack<Integer>
+ generalFavor: Stack<Integer>
/ TOUR_AND_DUCK: int
/ THREE_Rab_DUCK: int
/ SEVEN_LABEL_DUCK: int
/ GENERAL_TOUR_DUCK: int
+ DedicatedTokens(int,int,int,int)
+ getFavorWhite():int
+ getThreeRar():int
+ threeRarCount():int
+ favorWhiteCount():int
+ secondnextCount():int
+ generalFavorCount():int
+ getFavorWhite(int): void
+ getThreeRar(int): void
+ getSecondnext(int): void
+ getGeneralFavor(int): void
LanternCards 0..1
«Java Class»
LanternCards
mtree_main
+ stackp: Stack<Map.String,Integer>
+ numOfPlayers: int
+ LanternCards()
+ LanternCards(int)
+ LanternCards(int,int,int,int,int,int,int)
+ getRedCard():boolean
+ addRedcard(): void
+ redCardCount():int
+ getBlueCard():boolean
+ addBluecard(): void
+ blueCardCount():int
+ getGreenCard():boolean
+ addGreencard(): void
+ greenCardCount():int
+ getWhiteCard():boolean
+ addWhitecard(): void
+ whiteCardCount():int
+ getPurpleCard():boolean
+ addPurplecard(): void
+ purpleCardCount():int
+ getBlackCard():boolean
+ getBlackcard():int
+ addBlackcard(): void
+ getCreamCard():boolean
+ getGreyCard(): void
+ greyCardCount():int
+ Cg():Count(int):int
+ isNullCard():boolean
+ toString():void
+ getCard(String):void
+ getCard(String):boolean
+ cardsInHandSize(int):int
+ cardDecrease(int): void
+ getAllLanternCards(): void
+ toString():String
+ assignLanternCards(int,Array,int>Player>,LabelTiles,LanternCards):void
+ assignLanternCards(Player,int,int,int,LabelTiles,Array,int>Player>,LanternCards,FavorTokens):void
+ addTileBack(int):void
favorTokens 0..1
«Java Class»
FavorTokens
mtree_main
/ MAX_TOKENS: int
+ tokens: Stack<String>
+ FavorTokens()
+ FavorTokens(int)
+ displayFavorTokens(): void
+ displayFavorTokens(): void
+ displayCount():int
board 0..1
«Java Class»
Board
mtree_main
+ board: int[][]
+ Board(int[][],Array,int>,LabelTiles)
+ Board()
+ setBoardLabelsAndSetLabelTiles(): void
+ shuffleLabelTiles(LabelTiles): void
+ displayBoard(int[][],Array,int>,LabelTiles): void
«Java Class»
LabelTiles
mtree_main
+ leftColor: String
+ rightColor: String
+ upColor: String
+ downColor: String
+ left: int
+ right: int
+ up: int
+ down: int
+ tl: int
+ tr: int
+ platform: boolean
+ LabelTiles()
+ LabelTiles(String,String,String,String,int,int,int,int,boolean,LabelTiles[],Stack<LabelTiles>)
+ shiftLabelTiles(int): void
+ numberOfLabelTiles(LabelTiles[]):int
+ assignLabelTiles(int,Array,int>,LabelTiles>
+ getLabelTiles():LabelTiles
+ isLabelTiles(LabelTiles):boolean
+ rotateLabel(int):String
+ toString():String
+ rotateLabelTiles(int,int):LabelTiles
+ getColumnBoard(int,String):int
+ getRowBoard(int,String):int
nextTile 0..1
linkedLabelTiles 0..*