Gridlock – Interactive Puzzle Game   UML Class Diagram

BoundedQueue

<<Constructor>>BoundedQueue()

+remove():E
+add(E newValue):void
+Forceadd(E newValue):void
+isFull():boolean
+isEpty:boolean
+setDebug(boolean newValue:void

Main

gameScreen:Screen
menuScreen:Screen
networkScreen:Screen
engine:GameEngine

gameScreen:Screen
menuScreen:Screen
networkScreen:Screen
engine:GameEngine

+tart(Stage primaryStage) :void
+main(String[] args):void
+ ShowGameScreen:void
+ ShowMenuScreen:void
+ ShowNetworkScreen():void

Screen

-screenTile:String
-stage:Stage
-fxmlLoader:FXMLLoader
-controller:Controller
-scene:Scene

<<Constructor>>Screen()
+show():void
+loadShowSettings():void
+StageShow():void
+setController(Controller controller) :void
+getScene():Scene

Controll

+initialize():void

NetworkScreen

- controller:Controller
- stage:Stage

NetworkScreen(Stage s, GameEngine engine, Main main)

BoardScreen

- controller:Controller
- Stage:stage

+StageShow():void

MenuScreen

-controller:Controller

+MenuScreen(Stage s, GameEngine engine, Main main)

GameEngine

NumDifficultie: int
NumThreads: int
- queueList :ArrayList<BoundedQueue<Puzzle>>
- queue:BoundedQueue<Puzzle>
- executor:ExecutorService
- GameWin:boolean
- currDifficulty :int
- size:int
- gameMode :Mode
- StoryLevel :int
- gradeList: ArrayList<GradeLvl>
- NetworkMode:boolean
- NetworkWaiting:boolean
- NetworkPuzzle:Puzzle
- NetworkOpponentWon:boolean

+GameEngine(): void
+getStoryLevel():int
+getAnyPuzzle(): Puzzle
+getNewPuzzle:Puzzle
+GetCarList(): ArrayList<Car>
+MakeMove(int r, int c, int newR,int newC):boolean
+GameWon(int time):GradeLvl
+GetFinalGrade(): GradeLvl
+CalculateGrade(int timeLeft):GradeLvl
+GameLoss():void
+getNextMove():int
+getBoardSize:int
+UIGetPuzzle:ArrayList<Car>
+RestartPuzzle:void
+getMoves():int
+getMinMoves():int
+FindMoves(int r, int c):int[]
+SetDifficulty():int
+IncrementDifficulty():void
+DecrementDifficulty():void
+setMode(Mode mode)
+getTime():int
+getMode():Mode
+NetworkGetMove():int[]
+NetworkMoveWaiting():boolean
+NetworkSetMode(boolean net):void
+NetworkOpponentWon():boolean

GameThread

queueList:ArrayList<BoundedQueue<Puzzle>>
-queue:BoundedQueue<Puzzle>
-count:int
-size:int
-minMoves:int

+ <<Constructor>>GenThread()
+queueListFull:boolean
+findSmallestQueue:Difficulty
+run():void

Puzzle

+Puzzle(int n, int minMoves, boolean gen)
+getBestMove():int[]
+RestartPuzzle():void
+getSize():void
+getinitMoves():int
+GetBoard():ArrayList<Integer>
+ isGoalState():boolean
+MakeMove(int r,int c, int newR,int newC):boolean
+getMoves():int
+GenCarList():ArrayList<Car>
+GenSolution(int maxCars, int carProb, int verProb):GameBoard
+addCar(GameBoard board, int r, int c, int type):boolean
+RCtol(int r, int c):int
+printBoard():void
+FindMoves(int r, int c):int[]
+equals(Object obj):boolean
+getInitial():GameBoard

-GameBoard board: GameBoard
- minInitMoves: int
- numMoves:int
- Search search; //Add parameters here for size etc
-Size:int
- GoalC:int
-GoalR =2 ; //Shifted randomly
# EMPTY:int
# GOALCAR :int
# HCAR=4:int
# HTRUCK:int
#VCAR =1:int
VTRUCK =2: int

Mode

+ Mode: enum

GradeLvl

+ GradeLvl: enum
# mark:int
# s:String

<<Constructor>>GradeLvl()
+getMark():int
+getString():String

Difficulty

+Difficulty: enum
+moves:int

- <<Constructor>>Difficulty()
+getmoves():int

GameController

iGOALCAR :int
animTime :int
boardColor :Color
lineColor:Color
primaryTextColor :Color
Color secondaryTextColor :Color
boolean isGameWon:boolean
GameEngine engine:GameEngine
Main main:Main
squareWidth:double
int nSquares:int
workload:ArrayList<Car>
- goalCar:Car
running:boolean:boolean
boolean animating :boolean
Rectangle curtain:Rectangle
Pane boardPane:Pane
movesMade :Label
buttonNewGame :Button

+GameController(Stage s, GameEngine engine, Main main)
#curtainShow():void
curtainHide():void
MainMenuAction():void
NewGameAction():void
#drawBoard(Pane p, ArrayList<Car> wkload, boolean move): ArrayList
+MakeMove(int oldR, int oldC, int r, int c)():void
#GameWon():void
+GetAnimatingFin():boolean
+DisplayWinScreen():boolean
+findCar(int r,int c):Car
+AddCartoPane(Node c, Pane p):void
+GetNewBoard():void
#rawBorder(Pane p)():void
#drawCars(Pane p, ArrayList<Car> wkload, boolean move):ArrayList<Car>
#setCenterY(Pane pane, Label label):void

MenuController

<<Constructor>>MenuController()
StartButtonAction(ActionEvent event)():void
MultiplayerButtonAction(ActionEvent event):void
ExitButton(ActionEvent event):void
+ initialize():void
+helpButton():void
+initHelp():void
+sliderChanged():void
+toggleChanged(Toggle newToggle):void
+getDiffString(double difficulty):String

Car

-IMAGE:Image
-carNode:ImageView
#GOALCAR:int
#VERCAR:int
#VERTRUCK:int
-r:int
-c:int
-x:double
-y:double
-length:int
-squareLength:double
-type:int
-dragging:boolean
-mousex:double
-mousey:doubl
-boardReference:WeakReference<BoardController>
-initR:int
-initC:int
-moveSpace:int[]
-moveBounds:Bounds
initx:double
inity:double
moveable :boolean

<<constructor>> car
-frontEndCarConstructor(double squareLength, Bounds b,
BoardController bcontroller):void
+getCar:Node
+CoordtoN(double coord):int
+ NtoCoord(int n):int
+addMouseEvent():void
+getCaryType():int
+ CarMakeAnimatingMove(int newR, int newC, int animTime)
+makeSetAnimatingMove(int newR, int newC, int animTime) +handle(MouseEv
+ CarMakeAnimatingMove(int newR, int newC, int animTime) :void
+setCoord(double newCoord, double minCoord, double maxCoord):void
+setMoveable(boolean move):void

NetworkController

#networkUpdateTime :int
#oppAnimTime:int
-boardPane:Pane
-NboardPane:Pane
-NmovesMade:Label
-nworkload :ArrayList<>();
- theirgoalCar: Car
- timeline Timeline
- Ncurtain Rectangle
- scounter: Timeline
- pregame : boolean
- Timecounter : int
- message: Label
- GameOver: boolean

+NetworkController(Stage s, GameEngine engine, Main
+initialize():void
+GameWon():void
+MakeOpponentMove(int [] arr):int
+MakeMove(int oldR, int oldC, int r, int c):void
+GetNewBoard():void
+NcurtainHide():void
+NcurtainShow():void
+RequestNewGame():void

BoardController

-GAME_OVER_MSGS: String[]
-GAME_WON_MSGS:String[]
-totalTime:Label
-buttonPause:Button
-buttonNewGame:Button
-buttonHint:Button
-buttonMenu:Button
-countDonw:TimeLine
- totalSeconds: int
-currSeconds: int
- running: boolean
- mode: Mode

+BoardController(Stage s, GameEngine engine, Main main)
+initialize():void
HintAction():void
PauseAction(ActionEvent event):void
RestartAction(ActionEvent event):void
initialize():void
DisplayWinScreen():void
+void GetNewBoard():void
+drawBoardAdit():void
+stopGame(String msg)():void
+ GameWon():void
# StoryModeEndScreen():void
#convertTime(long secondDelta):String

Search

EMPTY=0:int
GOALCAR :int
HORCAR:int
HORTRUCK:int
VERCAR:int
VERTRUCK :int
closedMap:Map<ArrayList<Integer>,GameBoard>
queue:Queue<GameBoard>
Cmax:int
Rmax :int
Size:int
GoalC :in
GoalR :in
GoalCar:in
GenC:in
GenR:in
gen:boolean

<<constructor>> search
+GenBoard(GameBoard board):GameBoard
+SearchBoard(GameBoard board):LinkedList<GameBoard>
-FindMoves(GameBoard state, int moves):GameBoard
-FindPrev(GameBoard state, LinkedList<GameBoard> list):int
-addQueue(GameBoard next, GameBoard prev):void
-isGoal(GameBoard state):boolean
-FindNeighbour(GameBoard curr)
- UpSpaces(GameBoard state, int r, int c):int
- DownSpaces(GameBoard state, int r, int c):int
-LeftSpaces(GameBoard state, int r, int c):int
-RightSpaces(GameBoard state, int r, int c):int
- RCtol(int r, int c):int
-getSize(int id) :int

GameBoard

- arr : ArrayList<Integer>
- moves:int
- Size:int

+ <<Constructor>>GameBoard()
+set(int i, int j):void
+setMoves(int i)+setRC(int r, int c, int value): void
+getRC(int r, int c):int
+ RCtol(int r, int c): int
+size():int
+incMoves:void
+getMoves:int
+resetMoves:void
+getArr:Aarray<Integer>
+printBoard():void
+compareBoard(GameBoard oldB): int []
+FindMoves(int r, int c):int[]
+hashCode():int
+equals(Oject obj):boolean
+getSize(int id):int
+copyGameBoard():GameBoard
+findGoalCarR():int

SearchBoard

moves:int
size: int

StoredBoard(int moves, int size, Integer[] a
getBoard():GameBoard

Extends