RobsonCreateMesh

- mesh: RobsonBlock[][]
- entries: List<RobsonBlock>

+ robsonGenerateRoads(String caminhoArquivo, RobsonFactoryBlock factoryBlock): RobsonBlock[][]
+ robsonCheckEntry(): boolean
+ robsonCheckExit(): boolean
+ robsonGetRightBlock(int line, int column): RobsonBlock
+ robsonGetLeftBlock(int line, int column): RobsonBlock
+ robsonGetUpBlock(int line, int column): RobsonBlock
+ robsonGetDownBlock(int line, int column): RobsonBlock
+ robsonGetMesh(): RobsonBlock[][]
+ robsonGetEntries(): List<RobsonBlock>

RobsonSpawner

- entries: List<RobsonBlock>
- limit: Integer
- cars: RobsonCar[]
- quantityOfCars: Integer
- mesh: RobsonBlock[][]
- mutex: Semaphore
- tearDown: boolean
- pause: boolean
- insertionInterval: int

+ RobsonSpawner(List<RobsonBlock> entries, int limit, RobsonBlock[][] mesh, int insertionInterval): RobsonSpawner
+ run(): void
+ robsonPickCarImage(): ImageIcon
+ robsonGetPosition():int
+ robsonTryLockQuantityOfCars(): void
+ releaseQuantityOfCars(): void
+ robsonPause(): void
+ robsonTearDown(): void

RobsonCar

- image
- speed
- line
- column
- position
- mesh
- cars
- spawner

+ RobsonCar(ImageIcon image, int speed, int line, int column, RobsonCa
+ run()
+robsonGo()
+robsonAutoKill()
+robsonAutoPurge()
+ getters e setters

RobsonMeshView

- matrix: RobsonBlock[][]

+ RobsonMeshView(RobsonBlock[][] matrix)
+ robsonUpdateCarIcon(RobsonBlock block): void
+ robsonCreateBlock(RobsonBlock block): JPanel
+ robsonGetImageBlock(RobsonBlock block): ImageIcon
+ robosnGetImage(String imagePath):ImageIcon

RobsonMainView

- limitCars: int
- interval: int
- factoryBlock: RobsonFactoryBlock
- mesh: RobsonBlock[][]
- spawner: RobsonSpawner

+ RobsonMainView()
+ robsonBuildSettings(): void
+ robsonLoadActions(): void
+ robsonLoadMeshFiles(): void
+ robsonLoadSemaMonitor(): void

Main

+ main()

RobsonCommand

- block: RobsonBlock
- nextBlock: RobsonBlock
- car: RobsonCar
- mesh: RobsonBlock[][]

+ RobsonCommand(RobsonBlock block, RobsonCar car)
+ robsonExecute(): void
+ robsonBlockUp(): void
+ robsonBlockRight(): void
+ robsonBlockDown(): void
+ robsonBlockLeft(): void
+ robsonCross(): void
+ robsonTakeBlock(): void
+ robsonNextCrossBlock(): void
+ robsonCrossUp(block): void
+ robsonCrossRight(block): void
+ robsonCrossDown(block): void
+ robsonCrossLeft(block): void
+ robsonCrossUpRight(block): void
+ robsonCrossUpLeft(block): void
+ robsonCrossRightDown(block): void
+ robsonCrossDownLeft(block): void

RobsonMoveCommand

+ RobsonMoveCommand(RobsonBlock block, RobsonCar car)

RobsonFactoryBlock

+ create(boolean cross, boolean entry, boolean exit, int direction, int line, int column): RobsonBlock

---

RobsonCreateMonitorBlock

+ create(boolean cross, boolean entry, boolean exit, int direction, int line, int column): RobsonCreateMonitorBlock

---

RobsonCreateSemaphoreBlock

+ create(boolean cross, boolean entry, boolean exit, int direction, int line, int column): RobsonCreateSemaphoreBlock

---

RobsonBlock

- entry: boolean
- exit: boolean
- cross: boolean
- direction: int
- line: int
- column: int
- icon: ImageIcon
- car: RobsonCar
- meshView: RobsonMeshView

+ RobsonBlock():RobsonBlock
+ robsonLockBlock():void
+ robsonTryLockBlock():void
+ robsonReleaseBlock():void
+ robsonReleaseCar():void
+ getters e setters
+ toString():String

RobsonMonitorBlock

+ RobsonMonitorBlock(): RobsonMonitorBlock
+ robsonLockBlock():void
+ robsonTryLockBlock():void
+ robsonReleaseBlock():void
+ robsonReleaseCar():void

RobsonSemaphoreBlock

+ RobsonSemaphoreBlock(): RobsonSemaphoreBlock
+ robsonLockBlock():void
+ robsonTryLockBlock():void
+ robsonReleaseBlock():void
+ robsonReleaseCar():void