# Game

## MapTools

### MapChecker
- map: Map

- MapChecker(Map)
- checkConditions(Player)
- outOfBounds(Robot): boolean
- getLocation(Robot, TiledMapTileLayer): TiledMapTileLayer.Cell
- setLocation(Robot, TiledMapTileLayer.Cell)
- removeRobot()
- putRobot()

### Map
- layers: HashMap<String, TiledMapTileLayer>
- mapSizeX: int
- mapSizeY: int
- board: TiledMap

- Map(MapSelector)
- getMapSizeX(): int
- getMapSizeY(): int
- getBoard(): TiledMap
- getMapLayer(String): TiledMapLayer

### <<Enum>> MapSelector
- MAP1
- MAP2

- MapSelector(String, name)

## Cards

### <<Enum>> ProgramCard
- MOVE_1
- MOVE_2
- MOVE_3
- BACK_UP
- TURN_RIGHT
- TURN_LEFT
- U_TURN

- getNewDeck(): Stack<ProgramCard>
- ProgramCard(int, int, String)
- getSteps(): int
- getRotation(): int
- getName(): String
- getFace(): Sprite
- draw(Batch)

### ProgrammingDeck
- NUM_CARDS_HAND: int
- NUM_CARDS_SELECT: int
- programCardDeck: Stack<ProgramCard>
- ThrownCards: List<ProgramCard>

- ProgrammingDeck()
- Size(): int
- drawCard(): ProgramCard
- thrownCards(Collection<ProgramCard>)
- shuffle()
- getDeck(): Stack<ProgramCard>

### RoboRallyGame
- app: RoboRallyApp
- myConnection: Multiplayer
- players: List<Player>
- map: Map
- eventhandler: Eventhandler
- cheatmode: boolean
- mapChecker: MapChecker
- deck: ProgrammingDeck

- addPlayer()
- getPlayers()
- getMap()
- ExecuteProgramCard(int): boolean
- ExecuteProgramCard(Player, ProgramCard)
- cheatMove(Direction): boolean
- toggleCheatMode()
- isCheatModeOn()

## Objects

### Robot
- heading: Direction
- Vector2: Location

- heading(): Direction
- setHeading(Direction)
- setLoc(Vector2)
- setLoc(float, float)
- getLoc(): Vector2

### Player
- piece: Piece
- robot: Robot
- hand: List<ProgramCard>

- Player(id)
- getHand(): List<ProgramCard>
- getRobot(): Robot
- getPiece(): Piece

### <<Enum>> Piece
- PIECE1
- PIECE2
- PIECES: List<Piece>

-------------Cells---------------
- cell: TiledMapTileLayer.Cell
- diedCell: TiledMapTileLayer.Cell
- wonCell: TiledMapTileLayer.Cell

-----------Textures------------
- textures: TextureRegion[][]

- Piece(int)
- get(int): Piece
- getCell(): TiledMapTileLayer.Cell
- getDiedCell(): TiledMapTileLayer.Cell
- getWonCell(): TiledMapTileLayer.Cell

### EventHandler
- game: RoboRallyGame

- EventHandler(RoboRallyGame)
- HandleKeys(int): boolen

### <<Enum>> Direction
- NORTH
- SOUTH
- WEST
- EAST
- DIRECTIONS: List<Directions>

- getX(): int
- getY(): int
- rotate(Direction, int): Direction
- rotate(int):

### GameLib
- playProgramCard(Robot, ProgramCard)
- updateLocation(Robot, ProgramCard)
- updateHeadint(Robot, ProgramCard)
- move(Robot, Direction)

## <<absract class>> Game (Libdgdx)

### RoboRallyApp
- game: RoboRallyGame
- self: Player

- TITLE: String
- TITLE_SIZE: int
- LOGO_PATH: String
- ROBOT_SKINS_PATH: String
- GUI_SKIN_PATH: String
- CARD_SKINS_PATH: String

- TITLE: String
- GUI_SKIN: Skin
- stage: Stage
- titleScreen: Screen

- create()
- render()
- dispose()
- getGUI_SKIN()
- getTitleScreen()
- startNewGame()
- joinNameGame()
- getGame()
- getSelf()

# GUI.Screens

### <<interface>> IMenuScreen
- setTitle(String)
- setHeading(String)
- setLogo(String)
- addButton(String, float, InputListener): TextButton
- addTextField(String, float): TextField
- addLabel(String, float): Label
- topListener(): InputListener
- middleListener(): InputListener
- bottomListener(): InputListener

### <<interface>> Screen (Libdgdx)

### InputAdapter (Libdgdx)

### PlayerView
- renderer: OrthogonalTiledMapRenderer
- app: RoboRallyApp
- self: Player

- spriteBatch: SpriteBatch

- PlayerView(RoboRallyApp)
- show()
- resize()
- pause()
- resume()
- keyDown(int): boolean
- hide()
- dispose()

### <> MenuScreen
- game: RoboRallyApp
- stage: Stage
- heading: Label

- TOP1, TOP2: float
- BOTTOM1, BOTTOM2, BOTTOM3: float

- MenuScreen(RoboRallyApp)
- setTitle(string)
- setLogo(String)
- setHeading(String)
- addButton(String, float, InputListener): Textbutton
- addTextField(String, float): TextField
- addLabel(String, float): Label
- topListener(): InputListener
- middleListener(): InputListener
- bottomListener(): InputListener
- render(float)
- show()
- resize()
- pause()
- resume()
- hide()
- dispose()
- getCenterWidth()
- getCenterHight()
- getWidgetHight(float, float)
- getWidgetWidth(float)

## Multiplayer

### MultiplayerHostScreen
- ip: String
- localIP: String
- port: int
- ipLabel: Label
- portField: TextField

- MultiplayerHostScreen(RoboRallyApp)
- topListener(): InputListener
- middleListener(): InputListener
- bottomListener(): InputListener
- getLocalhost(): String
- getIp(): String
- hostPressed()

### MultiplayerJoinScreen
- ipField: TextField
- label: Label
- hostIP: String
- hostPort: int

- MultiplayerHostScreen(RoboRallyApp)
- topListener(): InputListener
- middleListener(): InputListener
- bottomListener(): InputListener
- joinPressed()

### MultiplayerOptionScreen
- MultiplayerHostScreen(RoboRallyApp)
- topListener(): InputListener
- middleListener(): InputListener
- bottomListener(): InputListener

# Multiplayer

### <<interface>> IMultiplayer
- register(EndPoint)
- sendPacket(Robot)
- recieved(Connection, Object)

### Multiplayer
- tcpPort: int
- connections: List<Connections>

- register(EndPoint)
- sendPacket(Robot)
- recieved(Connection, Object)

### Listener (Kryonet)

### MultiplayerHost
- server: Server

- MultiplayerHost()
- connected(Connection)
- disconnected(Connection)
- recieved(Connection, Object)

### MultiplayerClient
- client: Client
- hostIP: String
- hostPort: int
- TIMEOUT: int

- recievedPackets: Queue<Packet>

- MultiplayerClient(Client, int, int)