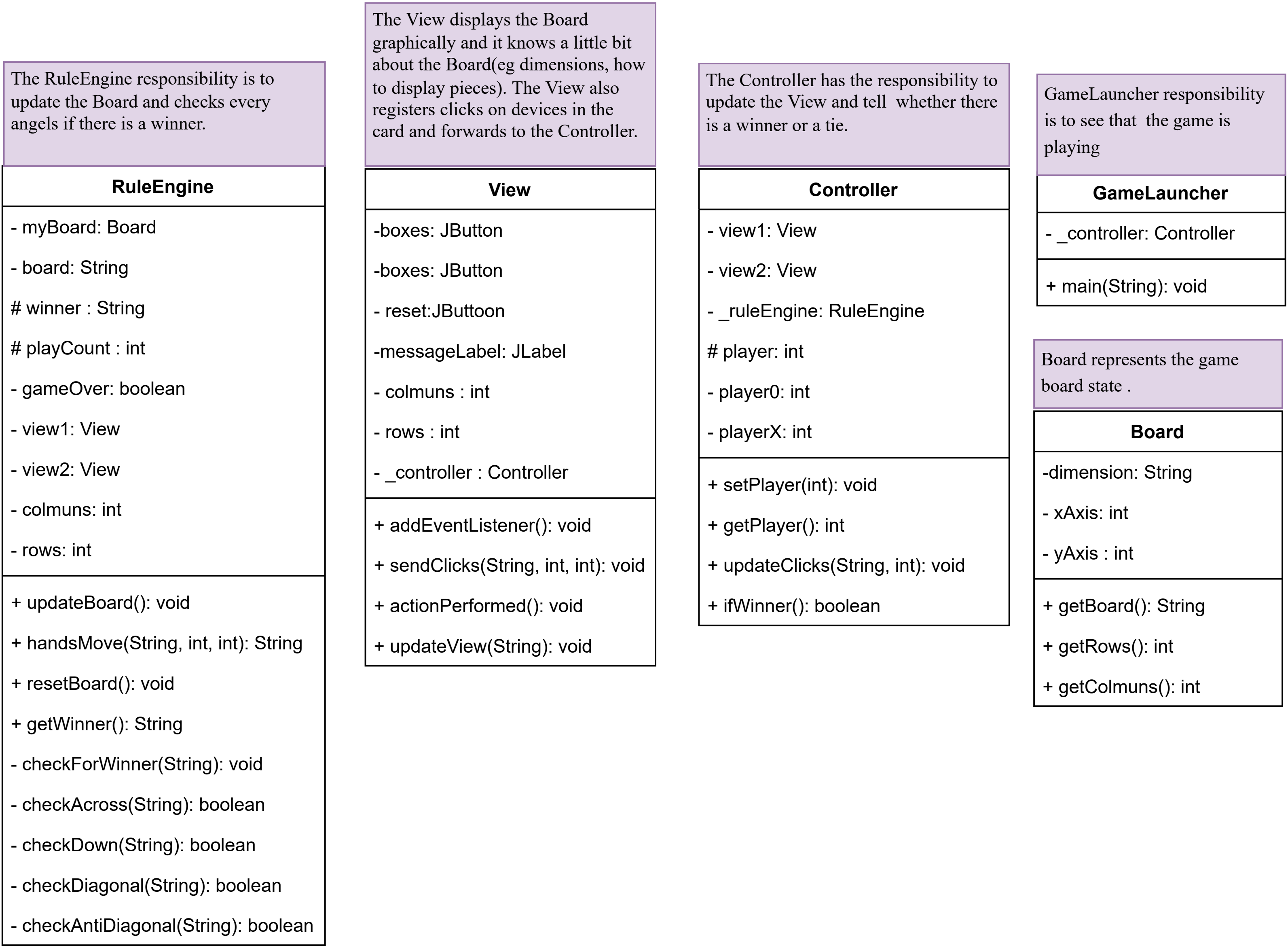

The RuleEngine responsibility is to update the Board and checks every angels if there is a winner.

RuleEngine

- myBoard: Board
- board: String
# winner : String
# playCount : int
- gameOver: boolean
- view1: View
- view2: View
- colmuns: int
- rows: int

+ updateBoard(): void
+ handsMove(String, int, int): String
+ resetBoard(): void
+ getWinner(): String
- checkForWinner(String): void
- checkAcross(String): boolean
- checkDown(String): boolean
- checkDiagonal(String): boolean
- checkAntiDiagonal(String): boolean

The View displays the Board graphically and it knows a little bit about the Board(eg dimensions, how to display pieces). The View also registers clicks on devices in the card and forwards to the Controller.

View

-boxes: JButton
-boxes: JButton
- reset:JButtoon
-messageLabel: JLabel
- colmuns : int
- rows : int
- _controller : Controller

+ addEventListener(): void
+ sendClicks(String, int, int): void
+ actionPerformed(): void
+ updateView(String): void

The Controller has the responsibility to update the View and tell whether there is a winner or a tie.

Controller

- view1: View
- view2: View
- _ruleEngine: RuleEngine
# player: int
- player0: int
- playerX: int

+ setPlayer(int): void
+ getPlayer(): int
+ updateClicks(String, int): void
+ ifWinner(): boolean

GameLauncher responsibility is to see that the game is playing

GameLauncher

- _controller: Controller

+ main(String): void

Board represents the game board state .

Board

-dimension: String
- xAxis: int
- yAxis : int

+ getBoard(): String
+ getRows(): int
+ getColmuns(): int