

→ methods

+ startDisplay(): void

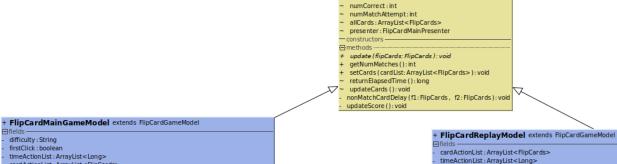
replay (delay: long): void

restoreStartState():void

update (cardCalled: FlipCards): void

+ FlipCardGameModel

fields ----
numMatches:int



- firstClick: boolean - timeActionList: ArrayList<Long> - cardActionList: ArrayList<RippCards> - cardActionList: ArrayList<RippCards> - constructors - FilpCardMainGameModel(difficulty: String, presenter: RipCardMainPresenter) - Emethods - filpCardGameManagerBuilder(difficulty: String, presenter: RipCardMainPresenter): void - update (cardCalled: RipCards): void - setMumMatches (difficulty: String): void - setMumMatches (difficulty: String): void - getInstructions(): String - getCardActionList(): ArrayList<RipCards> - getTimeActionList(): ArrayList<Long> - getLastStateList(): ArrayList<RipCards>

+ FlipCardGameView

startTime(): void

stopTime (): void

getColor():int

timeElapsed():long

getContext():Context

getDifficulty (): String

getSymbolChoice (): String

getTableLayout(): TableLayout

getResults (): FlipCardResult

gameEnded (results: FlipCardResult): void

displayInstructions (instructions: String): void

getCurrGame (): FlipCardMainGame Model

updateScore (toShow: String): void

-methods

FlipCardMainView extends AppCompatActivity implements FlipCardGameView currUser : UserInfoFacade flipCardScore: TextView timer: Chronometer tableLayout: TableLayout btnInstantreplay: Button btnFlipCardResult: Button result : FlipCardResult presenter : FlipCardGamePresenter i methods -# onCreate (savedInstanceState: Bundle): void + displayInstructions (instructions: String): void onBackPressed():void getSymbolChoice(): String getColor():int getDifficulty (): String getTableLayout(): TableLayout gameEnded (newResult: FlipCardResult): void startTime():void stopTime():void timeElapsed():long updateScore (toShow: String): void getContext(): Context getResults (): FlipCardResult getCurrGame(): FlipCardMainGameModel htnFlinCardResult (view: View): void

btnInstantReplay (view: View): void

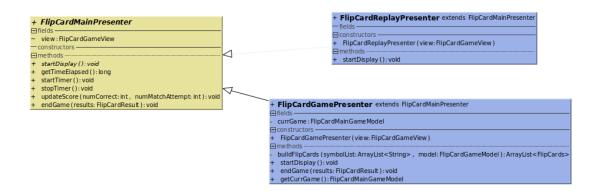
FlipCardReplayModel (cardActionList: ArrayList<FlipCardS>, timeActionList: ArrayList<Long>, lastStateList: ArrayList<FlipCardS>, presenter: FlipCardMainPresenter)

+ FlipCardCustomizationView extends AppCompatActivity implements RipCardCustomizationPresenter.View

- FlipCardCustomizationPresenter.View

- CustomizationSpn:Spinner
- currUser:UserInfoFacade
- constructors
- methods

onCreate (savedinstanceState: Bundle):void
- initializeScreenViaPresenter():void
+ addToSpinner(supportedSymbols:ArrayUst<String>):void
- spinnerValueGetter():String
- btnEndCustomizations (view:View):void
+ onBackPressed():void





⊟fields - currContext: Context - stk: TableLayout - symbolLst: ArrayList<String> - desiredManager: FlipCardGameModel - cardBackColor: int ⊟constructors + FlipCardSbuilder (symbolList: ArrayList<String> , packageContext: Context , stk: TableLayout , desiredManager: FlipCardGameModel , cardBackColor: int) ⊟methods - generateNewRow(): TableRow + createCards(): ArrayList<FlipCards>

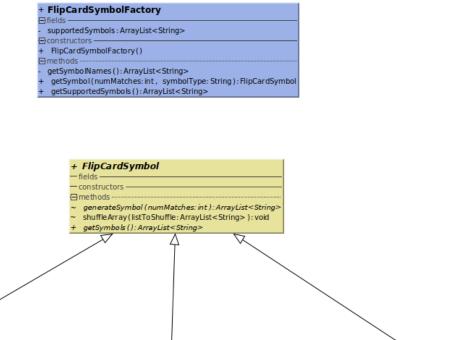
+ FlipCardsBuilder

FlipCards btnInstance: Button symbol : String flipped: boolear fullColor: Drawable enabled: boolean manager: FlipCardGameModel disableCards: boolean FlipCards (packageContext; Context, cardBackColor; int, symbol; String, row; TableRow, btnHeight; int, btnWidth; int, manager; FlipCardGameModel) build (packageContext: Context, cardBackColor: int, symbol: String, row: TableRow, btnHeight: int, btnWidth: int, manager: FlipCardGameModel): void initBoarderColor (btnInstance: Button, cardBackColor: int, btnWidth: int): Drawable isFlipped():boolean flipCard(): void getSymbol(): String lockCard(): void resetState():void setManager (manager: FlipCardGameModel): void disableBtnCall (): void turnCardToBack(): void

FlipCardResult implements Serializable ⊟fields numCorrect : int time To Completion: intnumTotalMatches:int difficulty: String → constructors + FlipCardResult (difficulty: String, numMatches: int, numTotalMatches: int, time: long) -methods roundTime (time: long): int getTimeToCompletion():int $getStrTimeToCompletion \ (): String$ getStrNumCorrect():String getStrNumIncorrect(): String getStrDifficulty (): String

callManagerUpdate():void

handleOnClick (button: Button): OnClickListener



~ CharacterSymbol extends FlipCardSymbol

generateSymbol (numMatches: int): ArrayList<String>

charList: ArrayList < String>

CharacterSymbol(numMatches: int)

+ getSymbols():ArrayList<String>

☐ constructors -

⊕methods ·

~ NumberSym... extends FlipCardSymbol

generateSymbol (numMatches: int): ArrayList<String>

numberList : ArrayList < String>

NumberSymbol (numMatches: int)

+ getSymbols(): ArrayList<String>

☐constructors -

~ NumberS ⊕ methods ···

~ AsciiSymbol extends FlipCardSymbol

generateSymbol (numMatches: int): ArrayList < String>

asciiList: ArrayList < String>

AsciiSymbol (numMatches: int)

+ getSymbols(): ArrayList<String>

⊕fields ·

☐ constructors –

☐ methods ·