

```

+ FlipCardInit
- fields
- constructors
+ FlipCardInit ()
- methods
+ startGame (currContext: Context): void

```

```

+ FlipCardGameModel
- fields
~ numMatches: int
~ numCorrect: int
~ numMatchAttempt: int
~ allCards: ArrayList<FlipCards>
~ presenter: FlipCardMainPresenter
- constructors
- methods
+ update (flipCards: FlipCards): void
+ getNumMatches (): int
+ setCards (cardList: ArrayList<FlipCards>): void
~ returnElapsedTime (): long
~ updateCards (): void
~ nonMatchCardDelay (f1: FlipCards, f2: FlipCards): void
~ updateScore (): void

```

```

+ FlipCardMainGameModel extends FlipCardGameModel
- fields
- difficulty: String
- firstClick: boolean
- timeActionList: ArrayList<Long>
- cardActionList: ArrayList<FlipCards>
- constructors
+ FlipCardMainGameModel (difficulty: String, presenter: FlipCardMainPresenter)
- methods
+ flipCardGameManagerBuilder (difficulty: String, presenter: FlipCardMainPresenter): void
+ update (cardCalled: FlipCards): void
+ setNumMatches (difficulty: String): void
+ getInstructions (): String
+ getCardActionList (): ArrayList<FlipCards>
+ getTimeActionList (): ArrayList<Long>
+ getLastStateList (): ArrayList<FlipCards>

```

```

+ FlipCardReplayModel extends FlipCardGameModel
- fields
- cardActionList: ArrayList<FlipCards>
- timeActionList: ArrayList<Long>
- constructors
+ FlipCardReplayModel (cardActionList: ArrayList<FlipCards>, timeActionList: ArrayList<Long>, lastStateList: ArrayList<FlipCards>, presenter: FlipCardMainPresenter)
- methods
+ startDisplay (): void
+ replay (delay: long): void
+ restoreStartState (): void
+ update (cardCalled: FlipCards): void

```

```

+ FlipCardMainView extends AppCompatActivity
implements FlipCardGameView
- fields
- currUser: UserInfoFacade
- flipCardScore: TextView
- timer: Chronometer
- tableLayout: TableLayout
- btnInstantReplay: Button
- btnFlipCardResult: Button
- result: FlipCardResult
- presenter: FlipCardGamePresenter
- constructors
- 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
+ btnFlipCardResult (view: View): void
+ btnInstantReplay (view: View): void

```

```

+ FlipCardResultView extends AppCompatActivity
- fields
~ user: UserInfoFacade
- result: FlipCardResult
- constructors
- methods
+ endGame (view: View): void
+ displayScores (newResult: FlipCardResult, difficulty: TextView, timeToCompletion: TextView, numCorrect: TextView, numIncorrect: TextView): void
# onCreate (savedInstanceState: Bundle): void
+ setResult (result: FlipCardResult): void
+ onBackPressed (): void

```

```

+ FlipCardGameView
- fields
- methods
+ gameEnded (results: FlipCardResult): void
+ updateScore (toShow: String): void
+ startTime (): void
+ stopTime (): void
+ timeElapsed (): long
+ getContext (): Context
+ getDifficulty (): String
+ getColor (): int
+ getSymbolChoice (): String
+ getTableLayout (): TableLayout
+ displayInstructions (instructions: String): void
+ getResults (): FlipCardResult
+ getCurrGame (): FlipCardMainGameModel

```

```

+ FlipCardCustomizationView extends AppCompatActivity
implements FlipCardCustomizationPresenter.View
- fields
- customizationSpn: Spinner
- currUser: UserInfoFacade
- constructors
- methods
# onCreate (savedInstanceState: Bundle): void
+ initializeScreenViaPresenter (): void
+ addToSpinner (supportedSymbols: ArrayList<String>): void
+ spinnerValueGetter (): String
+ btnEndCustomizations (view: View): void
+ onBackPressed (): void

```

```

+ FlipCardMainPresenter
├── fields
├── ~ view : FlipCardGameView
├── constructors
├── methods
+ startDisplay () : void
+ getTimeElapsed () : long
+ startTimer () : void
+ stopTimer () : void
+ updateScore (numCorrect: int, numMatchAttempt: int) : void
+ endGame (results: FlipCardResult) : void

```

```

+ FlipCardReplayPresenter extends FlipCardMainPresenter
├── fields
├── constructors
+ FlipCardReplayPresenter (view: FlipCardGameView)
├── methods
+ startDisplay () : void

```

```

+ FlipCardCustomizationPresen...
├── fields
├── ~ view : View
├── ~ symbolFactory : FlipCardSymbolFactory
├── constructors
+ FlipCardCustomizationPresenter (view: View)
├── methods
+ initializeScreen () : void

```

```

+ FlipCardGamePresenter extends FlipCardMainPresenter
├── fields
├── ~ currGame : FlipCardMainGameModel
├── constructors
+ FlipCardGamePresenter (view: FlipCardGameView)
├── methods
+ buildFlipCards (symbolList: ArrayList<String>, model: FlipCardGameModel) : ArrayList<FlipCards>
+ startDisplay () : void
+ endGame (results: FlipCardResult) : void
+ getCurrGame () : FlipCardMainGameModel

```

```

+ FlipCardsBuilder
├── fields
├── ~ currContext : Context
├── ~ stk : TableLayout
├── ~ symbolList : 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>

```

```

+ FlipCards
├── fields
├── ~ btnInstance : Button
├── ~ symbol : String
├── ~ flipped : boolean
├── ~ fullColor : Drawable
├── ~ enabled : boolean
├── ~ manager : FlipCardGameModel
├── ~ disableCards : boolean
├── constructors
+ FlipCards (packageContext: Context, cardBackColor: int, symbol: String, row: TableRow, btnHeight: int, btnWidth: int, manager: FlipCardGameModel)
├── methods
+ 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
+ callManagerUpdate () : void
+ handleClick (button: Button) : OnClickListener

```

```

+ FlipCardResult
├── Implements Serializable
├── fields
├── ~ numCorrect : int
├── ~ timeToCompletion : int
├── ~ numTotalMatches : 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

```

