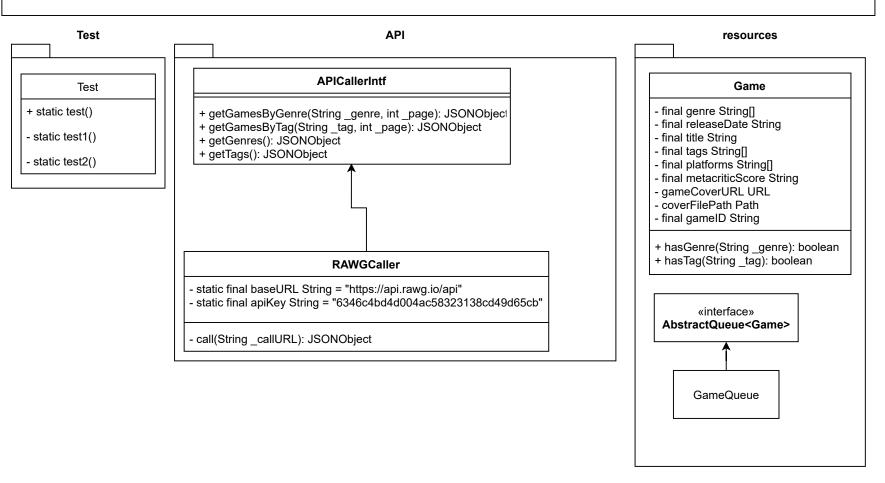
DataStorage DataStorageIntf + saveFile(JSONObject toBeStored): void + readFile(): JSONObject MockDataStorage desiredLoadedData JSON(DataStorage fileOut FileWriter static fileIn BufferedReader static final fileName String = "C:\\TestingProjectSaveFiles\\gameGenieSaveData.txt" static final canNotFindFileJSONString String = "{\"viewedGames\":[],\"userTags\":[],\"userGenre\":[]}" SaveDataTranslator static dataStorage DataStorageIntf + static saveUserData(UserHistoryIntf dataToSave): void + static loadUserData(): UserHistoryInts - static unwrap(JSONObject saveData): UserHistoryIntf - static jsonArrayToStrIntMap(JSONArray inputArray): HashMap<String, Integer> static jsonArrayToStrStrMap(JSONArray inputArray): HashMap<String, String> static toJSON(HashMap<String, Integer>, HashMap<String, Integer>, HashMap<String, String>): JSONObject

static fromStringIntMap(HashMap<String, Integer>): JSONArray

static fromStringStringMap(HashMap<String,String>): JSONArray

GameFactory GameTranslatorIntf gameGenres String[] + gerGamesByGenre(String _genre, int _page): Game[] - gameTags String[] + getGamesByTag(String tag, int page): Game[] - gameTranslator GameTranslator + getGenres(): String[] - static gameFactory GameFactory + getTags(): String[] - pageKeeperGenres HashMap<String, Integer> - pageKeeperTags HashMap<String, Interger> - isArryaRated(Game[] toRate): boolean + getGameQueue(): GameQueue<Game> GameTranslator + getRecommendation(): Game - compareGameBasedÖnTag(Game _Game1, Game _game2, String _favoriteTag): Game - myCaller APICallerInft getAnArrayThatIsNotRated(Game[] toCheck, String searchValue): Game[] - parseGameJSON(JSONObject_toParse): Game[]

game



user

UserHistoryIntf

- + getTopGenre(): String
- + getTopTag(): String
- + addGenre(String genre, int preferenceMod): void
- + addTag(String tag, int preferenceMod): void
- + addGame(String gameTitle, String gameID): void
- + hasViewedGame(String gameID): boolean
- + isEmpty(): bollean
- + isEqual(UserHistoryIntf inputObi): boolean

UserHistory

- userGenres DoubledLinkList
- userTags DoubledLinkList
- viewedGames HashMap<String, String>
- fromMapToLinkList(HashMap<String, Integer> _inputMap): DoubledLinkList
- fromLinkListToMap(DoubledLinkList inputList): HashMap<String, Inreger>

DoubledLinkList

- + addElement(String nodeTitle, int preferenceValue): void
- searchAndPlace(Node node): boolean
- · orderAscending(): void
- orderDescending(): void
- swap(Node lesserValue, Node greaterValue): void
- printList(): void
- hasNode(Node toCompare): boolean
- + isEqual(DoubledLinkList toCompare): boolean
- + greatestValue(): String
- + lowestValue(): String
- + empty(): boolean

User

- static user User
- userHistory UserHistoryIntf
- + static addStartScreenSelections(CheckBox[] userSelection, int incrementValue): void
- + static parseGame(Game inputGame, int incrementValue): void
- addViewedGame(String gameID, String gameTitle): void
- + wasViewed)String _gameID): boolean
- addGenre(String genreName, int inputValue): void
- addTag(String tagName, int inputValue): void

- # head Node = null
- # tail Node = null

viewLogic

GamePickerLogic

- currentGame Game
- # gameTitle Lable
- # getRecommendationButton Button
- # dislikeButton Button
- # likeButton Button
- # doNotKnowButton Button
- # gameCoverArtImageView ImageView
- + initialize(): void
- + DislikeClicked(ActionEvent actionEvent
- + LikeClicked(ActionEvent actionEvent)
- + DoNotKnowClicked(ActionEvent actionEvent)
- + setCurrentGame(Game game)
- + getNextGame()
- + getRecommendationClicked()

GameRecommendationLogic

- currentGame Game
- gamePlatforms Label
- gameTitleLabel Label
- gameReleaseDate Label
- gameGenre Label
- gameMetacritic Lable
- recommendationGameTitle
- recommendationGameCoverArt ImageView
- doneWithRecommendationButton Button
- + initialize()
- doneClicked()

StartScreenLogic

- actionBox CheckBox
- shooterBox CheckBox
- indieBox CheckBox
- casualBox CheckBox
- adventureBox CheckBox - rpgBox CheckBox
- strategyBox CheckBox
- simulationBox CheckBox
- puzzleBox CheckBox
- arcadeBox CheckBox
- platformerBox CheckBox
- racingBox CheckBox
- sportsBox CheckBox
- massivelyMultiplayerBox CheckBox
- fightingBox CheckBox
- boardGamesBox CheckBox
- educationalBox CheckBox
- cardBox CheckBox
- + doneButton Button
- + initialize
- userClickedDoneButton()

gameGenie

GameGenieController

- primaryStage Stage
- -static controller GameGenieController
- -static gameQueue GameQueue<Game>
- -static final startScreenFile String = "/startScreen.fxml"
- -static final gamePickerFile String = "/gamePickerScreen.fxml"
- -static final gameRecommendationFile String = "/gameRecommendationScreen.fxml"

setPrimaryStage(Stage _stage): void -updateStage(String _fxmlFile): void

- + changeSceneIntoGamePicker(): void
- + changeSceneIntoGameRecommendation(): void
- + static getInstance(): GameGenieController
- + static handleStartCheckBoxes(CheckBox[] _startCheckBoxes): void
- + static getGamePickerGame(): Game
- + static userLikedGame(): void
- + static userDislikedGame(): void
- + static userDoesNotKnow(): void
- + static getRecommendation(): Game
- + static applicationClosing(): void

Main

- + static main(String[] args): void
- + start(Stage orimaryStage): void
- + stop(): void

UserController

- static final int defaultLikeValue = 1
- static final int defaultDislikeValue = -1

static handleCheckBoxes(CheckBox[] _checkBoxArray): void

static liked(Game _game): void

static disliked(Game game): void

static userDataLoaded(): boolean

static programClose(): void

GameController

+ field: type

static getRecommendation(): Game

static getGameQueue(): GameQueue<Game>

- + static wasGameViewed(String _gameID): boolean
- + static mostLikedGenre(): String
- + static mostLikedTag(): String