Unpackaged Classes

GameManager

- + totalScore: int
- userManager: UserManager
- currLevelType: LevelType
- activeUser: User
- levels: int
- runningAds: boolean
- difficulty: int
- + GameManager()
- GameManager(in: Parcel)
- + writeToParcel(dest: Parcel, flags: int): void
- + describeContents(): int
- + getActiveUser(): User
- + setActiveUser(activeUser: User): void
- + setRunningAds(runningAds: boolean): void
- + setLevels(levels: int): void
- + getDifficulty(): int
- + setDifficulty(difficulty: int): void
- + start(context: AppCompatActivity): void
- + continueFromSave(context: AppCompatActivity): void
- loadSettingsFromUser(): void
- + play(context: AppCompatActivity): void
- gameOver(context: AppCompatActivity): void
- endGame(context: AppCompatActivity): void
- updateHiscores(): void
- continuePlaying(context: AppCompatActivity): void
- createRandomLevel(context: AppCompatActivity): Intent
- createGivenLevel(context: AppCompatActivity): Intent
- createLevelIntent(context: Context): Intent
- addExtras(intent: Intent): void
- updateActiveUser(): void
- tryRunAds(context: AppCompatActivity, intent: Intent): void
- showAd(context: AppCompatActivity, intent: Intent): void
- adNoShow(context: AppCompatActivity, intent: Intent): void
- donate(context: Contet): void
- -create End Popup (context: App Compat Activity, in tent: Intent): Alert Dialog. Builder
- transitionLevel(context: AppCompatActivity, intent: Intent): void



MainActivity

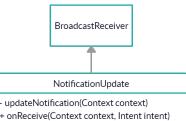
- + mContext: Context
- ~ notificationTime: int = 60
- + getMContext(): Context
- # onCreate(SavedInstanceState bundle): void
- startGame(): void
- initializeGame(): void
- · initializeNotifications(): void

Question

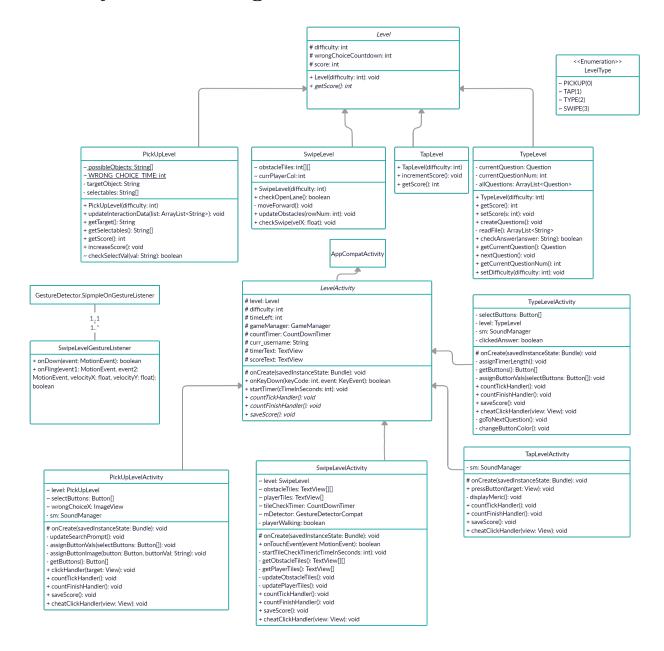
- question: String
- answers: ArrayList<String>
- correctAnswer: String
- randomizeAnswerOrder(): void
- + getQuestions(): String
- + getAnswers(): ArrayList<String>
- + getCorrectAnswer(): String
- + setCorrectAnswer(String correctAnswer): void

SoundManager

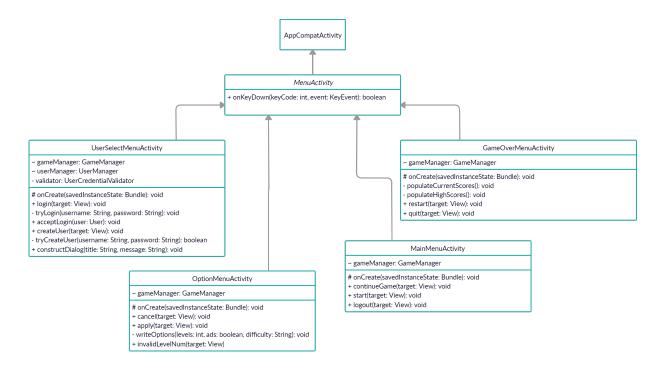
- mContext: Context
- main: MediaPlayer
- wowEffect: MediaPlayer
- booEffect: MediaPlayer
- + SoundManager(Context c)
- + startMainMusic(): void
- + playWowEffect(): void
- + playBooEffect(): void



MenuSystems Package



LevelSystems Package



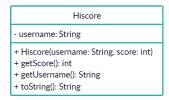
FileInterfacing Package

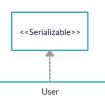
- file: File

~ FileInterface(filepath: String) - openFile(filepath: String): void ~ readFile(): List<String> - readDataFromFile(scan: Scanner): List<String> ~ writeToFile(data: String): void FileController - fileInterface: FileInterface ~ FileController(appDataDir: String, fileName: String) + load(): List - formatReadData (fileData: List < String >): List < String[] >~ updateList(loadedData: List<String[]>): List + save(objects: List): void ~ formatOutputData(objects: List): String HiscoreFileController UserFileController ~ updateList(loadedData: List<String[]>): List ~ updateList(loadedData: List<String[]>): List ~ formatOutputData(objects: List): String ~ formatOutputData(objects: List): String - createDataFromHiscore(hiscore: Hiscore): StringBuilder - createDataFromHiscore(hiscore: Hiscore): StringBuilder

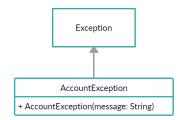
FileInterface

DataTypes Package





- username: String
- password: String
- pickupScore: int
- tapScore: int
- typeScore: int
- swipeScore: int
- hiScore: int
- currentRunLevels: int
- currentRunLevelType: LevelType
- currentRunDifficulty: int
- runningAds: boolean
- + User(username: String, password: String)
- + isRunningAds(): boolean
- $+\ set Running Ads (running Ads:\ boolean)$
- + getCurrentRunDifficulty(): int
- + setCurrentRunDifficulty(currentRunDifficulty: int): void
- $+ \ getCurrentRunLevels (): int$
- $+ \ set Current Run Levels (current Run Levels: int): void \\$
- + getCurrentRunLevelType(): LevelType
- + set Current Run Level Type (current Run Level Type: Level Type): void
- + getPickupScore(): int
- + getTapScore(): int
- + getTypeScore(): int
- + getSwipeScore(): int
- + getScore(): int
- + setScore(score: int, game: LevelType)
- + getHiscore(): int
- + setHiscore(hiscore: int): void
- + getUsername(): String
- + getPassword(): String



HiscoreManager

manager: HiscoreManager

- hiscores: List<Hiscore>
- HiscoreManager(appDataDir: String)
- + initialize(directory: String): void
- + getManager(): HiscoreManager
- refresh(): void
- + saveState(): void
- + getHiScores(): List<Hiscore>
- + addHiscore(newHiscore: Hiscore): void

UserCredentialValidator

- manager: UserManager

- + UserCredentialValidator()
- + login(username: String, password: String): User
- + createAccount(username: String, password: String): void
- -validateInputAndCreateAccount (username: String, password: String): void
- validInput(username: String, password: String): boolean
- createAccountInManager(): void

UserManager

- <u>manager: UserManager</u>
- fileController: UserFileController
- users: List<User>
- UserManager(appDataDir: String)
- + initialize(directory: String): void
- + getManager(): UserManager
- refresh(): void
- + saveState(): void
- ~ attemptLogin(username: String, password: String): Optional<User>
- loginCorrect(user: User, username: String, password: String): boolean
- ~ createUser(username: String, password: String): void
- ~ checkUserExists(username: String)
- + getUser(username: String): User