25 Key Code Sections in Snake Game

1. GameManager.updateGame()

File: GameManager.java
Context: Core game logic
Function: Updates the game state for each frame, handling snake movement, collisions, and food eating. This is the heart of the gameplay mechanics. It checks for border/self collisions, updates the snake position based on direction, and manages score when food is eaten.

2. GameManager.run()

File: GameManager.java
Context: Game loop
Function: The main game loop that runs on a separate thread. It controls the timing of updates and rendering, ensuring the game runs at a consistent speed. It calls updateGame() and drawSurface() at fixed intervals and manages the game-over flash effect.

3. GameManager.drawSurface()

File: GameManager.java
Context: Game rendering
Function: Renders the game state to the screen, drawing the background, snake segments, food, and game over messages. This method is responsible for all visual aspects of the gameplay.

4. MainActivity.checkUser()

File: MainActivity.java
Context: User authentication
Function: Authenticates users against both local storage and Firebase. It implements a fallback mechanism that first checks local storage and then Firebase if needed. It shows progress indicators and handles login errors.

5. GameActivity.onBackPressed()

File: GameActivity.java
Context: Navigation handling
Function: Prevents accidental exits from the game by showing a confirmation dialog when the back button is pressed during active gameplay. This helps maintain game state and improve user experience.

6. GameManager.handleGameOver()

File: GameManager.java
Context: Game over handling
Function: Manages what happens when the game ends. It sets the game state, saves high scores to Firebase, fetches a motivational quote, and shows the restart button through GameActivity.

7. settings.handleBackNavigation()

File: settings.java
Context: Navigation between screens
Function: Manages proper navigation from the settings screen back to the calling activity (either MainActivity or GameActivity). It preserves state by using the "CALLING_ACTIVITY" parameter to determine the correct return path.

8. GameActivity.settingsLauncher

File: GameActivity.java
Context: Settings navigation
Function: An ActivityResultLauncher that properly handles navigation to and from the settings screen during gameplay. It preserves the game state and refreshes the appearance when returning from settings.

9. MyFBDB.updateUserHighScore()

File: MyFBDB.java
Context: Firebase integration
Function: Updates a user's high score in Firebase, but only if the new score is higher than the existing one. It implements an asynchronous callback mechanism to handle database operations without blocking the UI.

10. GameActivity.updateScore()

File: GameActivity.java
Context: UI updates during gameplay
Function: Updates the score display in the UI and checks if the current score exceeds the personal high score. It's called by GameManager whenever the snake eats food and the score increases.

11. GameActivity.loadHighScores()

File: GameActivity.java
Context: High score retrieval
Function: Retrieves personal and global high scores from Firebase and updates the UI. It handles errors gracefully and uses asynchronous callbacks to prevent blocking the main thread.

12. PrefsManager.saveSnakeColor()

File: PrefsManager.java
Context: Game customization
Function: Saves the user's preferred snake color to SharedPreferences. This is used by the settings screen to customize the game appearance.

13. GameManager.refreshAppearance()

File: GameManager.java
Context: Game customization
Function: Updates the snake's visual appearance based on the user's preferences. It's called when returning from settings to immediately apply the new color without restarting the game.

14. UserFileStorage.saveUser()

File: UserFileStorage.java
Context: Local data persistence
Function: Saves a user's credentials to local storage. This enables offline login and serves as a cache for Firebase users to reduce network requests.

15. MyFBDB.userExistsAsync()

File: MyFBDB.java
Context: Firebase authentication
Function: Asynchronously checks if a username exists in the Firebase database. It's used during login and registration to verify user credentials and prevent duplicate registrations.

16. GameManager.setDirection()

File: GameManager.java
Context: Game controls
Function: Changes the snake's direction based on user input, with validation to prevent 180-degree turns (which would cause immediate game over). It's called by the direction buttons in GameActivity.

17. GameManager.initGame()

File: GameManager.java
Context: Game initialization
Function: Initializes or resets the game state, including snake position, score, direction, and game over status. It's called when starting a new game or restarting after game over.

18. MainActivity.launchGame()

File: MainActivity.java
Context: Navigation
Function: Starts the game after successful login by launching GameActivity with the username. This handles the transition from the login screen to the game screen.

19. settings.saveSnakeColorAndNotify()

File: settings.java
Context: Game customization
Function: Saves the selected snake color and provides user feedback through a toast message. It's called when the user selects a color in the settings screen.

20. GameActivity.onCreate()

File: GameActivity.java
Context: Activity initialization
Function: Sets up the game screen, including finding UI elements, getting the username from the intent, loading high scores, and initializing the GameManager. It also sets up button listeners for game controls.

21. GameActivity.showGameOverUI()

File: GameActivity.java
Context: Game over handling

Function: Shows the restart and high scores buttons when the game ends. It's called by GameManager when a collision is detected and the game ends.

22. MyFBDB.getUserHighScore()

File: MyFBDB.java
Context: High score tracking
Function: Retrieves a user's high score from Firebase. It's used to display personal best scores and compare with current scores.

23. MyFBDB.getGlobalHighScore()

File: MyFBDB.java
Context: High score tracking
Function: Retrieves the highest score across all users from Firebase. It's used to display the global top score in the stats bar.

24. MainActivity.saveLoginToPreferences()

File: MainActivity.java
Context: User convenience
Function: Saves the last login credentials to SharedPreferences for easier subsequent logins. This improves the user experience by remembering credentials.

25. settings.updateMusicService()

File: settings.java
Context: Game audio
Function: Toggles the background music service based on user preferences. It starts or stops the BackgroundMusicService when the music switch is toggled.