**Dino Run Game Documentation**

**Introduction**

This documentation provides an overview of the C++ code for a simple console-based Dino Run game. The game includes a dinosaur character that can jump to avoid obstacles (hurdles) and accumulate points. The user interacts with the game through keyboard inputs.

**Table of Contents:**

1. [Prerequisites]

2. [Code Overview]

- [Header Files]

- [Global Variables]

- [Class: DinoGame]

- [Public Methods]

- [gotoxy]

- [setcursor]

- [displayScore]

- [init]

- [moveDino]

- [drawHurdle]

- [play]

- [instructions]

- [LoadingBar]

3. [Main Function]

4. [User Interface]

5. [Game Flow]

6. [Conclusion]

**Prerequisites:**

Before running the Dino Run game, ensure that a C++ compiler is installed on your system. The code uses standard C++ libraries and does not have external dependencies.

**Code Overview:**

**Header Files:**

The code includes several standard C++ header files for input/output, console manipulation, time handling, and file operations. These include `<iostream>`, `<conio.h>`, `<time.h>`, `<windows.h>`, `<stack>`, `<queue>`, `<cstdlib>`, and `<fstream>`.

**Global Variables:**

- `scoreStack`: A stack to store and manage the player's scores.

- `score`: The current score of the player.

- `console`: Handle to the console for cursor manipulation.

- `CursorPosition`: Represents the cursor position on the console.

- `dinoY`: Represents the vertical position of the dinosaur.

- `speed`: Controls the speed of the game.

- `gameover`: Indicates whether the game is over.

**Class: DinoGame**

**Public Methods:**

1**. `gotoxy`**

- Description: Positions the cursor at a specified console coordinate.

- Parameters: `int x, int y` - X and Y coordinates.

- Usage: Used for drawing and positioning elements on the console.

2. **`setcursor`**

- Description: Sets the cursor visibility and size.

- Parameters: `bool visible, DWORD size` - Visibility and size of the cursor.

- Usage: Configures the console cursor.

3. **`displayScore`**

- Description: Displays the current score on the console and writes it to a file.

- Parameters: `stack<int> scoreStack, int score` - The stack of scores and the current score.

- Usage: Updates and shows the score during gameplay.

4. **`init**`

- Description: Initializes the game by drawing the boundaries and title on the console.

- Usage: Called at the beginning of the game.

5. **`moveDino`**

- Description: Moves the dinosaur character on the console.

- Parameters: `int jump` - Controls the dinosaur's jump.

- Usage: Animates the dinosaur's movement.

6. **`drawHurdle`**

- Description: Draws hurdles on the console and handles collisions.

- Usage: Animates and manages the hurdles in the game.

7. **`play`**

- Description: Main game loop that processes user input and updates the game state.

- Usage: Controls the flow of the game.

8. **`instructions`**

- Description: Displays game instructions on the console.

- Usage: Called when the user selects the instruction option.

9. **`LoadingBar`**

- Description: Displays a loading bar while simulating a loading process.

- Usage: Provides visual feedback during initialization.

**Main Function:**

The `main` function serves as the entry point for the program. It initializes the game, displays a menu, and processes user input to start the game, show instructions, or quit.

**User Interface:**

The user interface is text-based and displayed in the console. It includes a menu with options to start the game, view instructions, or quit. During gameplay, the dinosaur and hurdles are represented using ASCII characters.

**Game Flow:**

1. The user is presented with a menu to start the game, view instructions, or quit.

2. If the user chooses to start the game, a loading bar is displayed.

3. The main game loop initiates, allowing the player to control the dinosaur's movement and jump to avoid hurdles.

4. The game updates the score, displays it on the console, and writes it to a file.

5. If the player collides with a hurdle, the game ends, and a game-over message is displayed.

6. The user can choose to view instructions, which provide details on controls and gameplay.

7. The user can exit the game at any time by selecting the quit option.

**Conclusion:**

The Dino Run game provides a simple yet entertaining console-based experience. Players can control a dinosaur character, jump to avoid hurdles, and accumulate scores. The code is structured using a class-based approach, making it modular and easy to understand.