**CS210 MINI PROJECT- PHASE II**

Name- **Ayush Kumar**

Roll No.- **2201CS95**

Game- **Catching Letters**

**Game Description-**

Players start with a blank black screen with a red line near the bottom of the screen. Letters start appearing near the top of the screen, randomly from A through Z and move downwards. The goal of the game is for the user to last as long as possible without letting any of the letters hit the red line. This game can be used as typing practice, as it can increase the users typing coordination and speed. It forces them to type letters on a keyboard as fast as possible.

**How To Play**-

The game is played with the use of the PS/2 keyboard device. The user is to type the appropriate keys on this device in order to ‘catch’ the letters while they are on their way to the red line. If a user types any key and the letter for that corresponding key is not on the screen at that instance, then nothing will occur and the letters will keep moving down. When the user ‘catches’ a letter, a new one will be displayed near the top half of the screen. The game over screen displays once any letter hits the red line.

The switches on the De1-SoC board have been also used for the games functionality. Putting SW1 high will result in the game to pause. Putting SW2 high will result in the game being reset. The game will continue to reset at random as long as SW2 is high. The remaining switches have no use. An effective game strategy would be to focus on keys in the lower half of the screen, as typing the letters in the top half will keep creating more near this region. Another useful strategy is to look for letters that appear more than once and try to eliminate those in a combination.

**Flow of Code:**

1. **Start**:
   * Begin program execution.
2. **Initialize Variables and Hardware**:
   * Initialize positions, velocities of letters, and hardware-specific settings (pixel buffer addresses).
3. **Enter Main Game Loop**:
   * Loop until the game is reset.
4. **Check Switches**:
   * If "Pause" is toggled, enter a pause loop.
   * If "Reset" is toggled, reinitialize all settings and restart the main loop.
5. **Game Update and Render**:
   * Clear the screen.
   * Draw a red line at a fixed position as a game boundary.
   * Iterate through each letter:
     + Get keyboard input.
     + Check if the corresponding key is pressed:
       - If yes, reset the letter's position and speed.
       - If no, update the position based on its velocity.
     + Draw the letter.
   * After updating all letters, use vertical synchronization to manage drawing.
6. **Check Game Over**:
   * If any letter crosses the boundary, set the game state to "over".
7. **Game Over Screen**:
   * Clear the screen and display "GAME OVER!" using characters on the screen.
   * Stay in this state until the game is reset.
8. **Reset Game**:
   * If reset switch is activated, reinitialize all game settings and restart the main loop.
9. **End**:
   * Program execution is halted or restarted based on user interaction with reset.