Project Title: Number Guessing game

Group Number: 2

Team Members : OM, Samira , Anish

Submission Date: 23/10/25

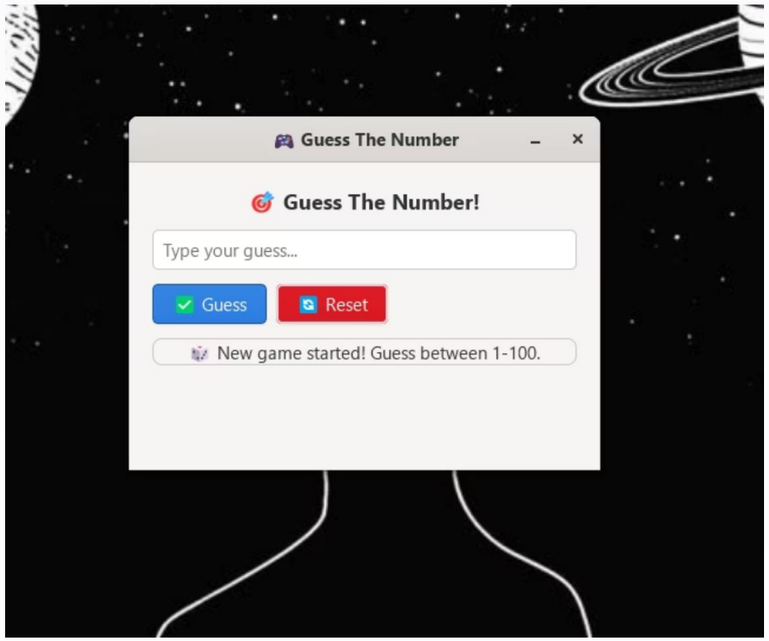
Submitted To: Anish

Project Type: GAME

Numbers Guessing Game

Mini Project Presentation

Hello everyone! First of all I'll introduce my self. My self samira vadke. Our group is group no. 2. I'll be presenting out mini project which is numbers guessing game.



Project Overview

Main Idea & Goal

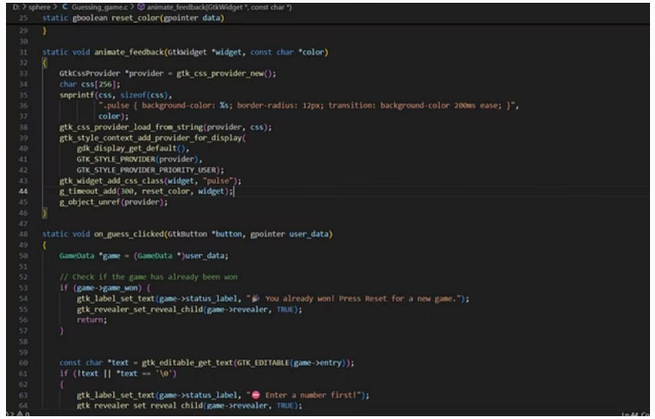
The Concept

The main idea of this project is to create an interactive game where the computer randomly selects a number between 1 and 100, and the player has to guess it. After each guess, the game provides feedback like "Too high", "Too low", or "Correct!", until the player finds the right number.

The Goal

The goal of this project was to make a fun and simple logic-based game while also learning how to design a Graphical User Interface (GUI) using the GTK library in C.

Technologies & Tools Used



```c
static gboolean reset_color(gpointer data)
}

static void animate_feedback(GtkWidget *widget, const char *color)
{
    GtkCssProvider *provider = gtk_css_provider_new();
    char css[256];
    snprintf(css, sizeof(css),
             ".pulse { background-color: %s; border-radius: 12px; transition: background-color 200ms ease; }",
             color);
    gtk_css_provider_load_from_string(provider, css);
    gtk_style_context_add_provider_for_display(
        gdk_display_get_default(),
        GTK_STYLE_PROVIDER(provider),
        GTK_STYLE_PROVIDER_PRIORITY_USER);
    gtk_widget_add_css_class(widget, "pulse");
    g_timeout_add(300, reset_color, widget);
    g_object_unref(provider);
}

static void on_guess_clicked(GtkButton *button, gpointer user_data)
{
    GameData *game = (GameData *)user_data;

    // Check if the game has already been won
    if (game->game_won) {
        gtk_label_set_text(game->status_label, "🎉 You already won! Press Reset for a new game.");
        gtk_revealer_set_reveal_child(game->revealer, TRUE);
        return;
    }


    const char *text = gtk_editable_get_text(GTK_EDITABLE(game->entry));
    if (!text || *text == '\0')
    {
        gtk_label_set_text(game->status_label, "🔴 Enter a number first!");
        gtk_revealer_set_reveal_child(game->revealer, TRUE);
```

Programming Language

We have build this project using c programming language.

GUI Framework

GTK (GIMP Toolkit) — used for building the graphical user interface such as buttons, labels, and entry boxes.

Development Environment

vs code, MSYS2 (UCRT64) — used for compiling and running the GTK-based C program.

Libraries Used:

- *gtk/gtk.h → for GUI components*
- *stdlib.h and time.h → for generating random numbers.*

Team Collaboration

This project was completed as a team effort.

Om Shinde

Our team lead have done Base code and core logic

Anish

Has done Designing user interface (UI)

Samira Vadke

I was responsible for making the report and adding comments to the code

Together, we ensured that the application was both functional and user-friendly.

daily updates

17th October 2025 (Friday)

- *Project assigned to the team.*

18th October 2025 (Saturday)

- *Tasks distributed:*
  - *Om: Base code and core logic*
  - *Anish: Designing user interface (UI)*
  - *Samira: Comments and report documentation*

19th October 2025 (Sunday)

- *Om completed the base code and core logic implementation.*

20th October 2025 (Monday)

- *Anish completed the UI design.*
- *Samira complete comments and the report work.*
- *Om tested and finalized the full project integration.*

22nd October 2025 (Wednesday)

- *Samira updated the final report.*
- *Final version of the project prepared for submission.*

Demonstration of the Project

Now, I'll demonstrate how the game works. LIVE

01

Game Start

When the program starts, a window appears titled "Guess the Number".

02

Player Input

The player enters a number between 1 and 100 in the input box.

03

Guess Button

On clicking the "Guess" button, the game checks if the entered number is correct.

04

Feedback

If it's too high or too low, the program displays feedback accordingly.

05

Success Message

When the player guesses correctly, a congratulatory message appears along with the total number of attempts.

06

Reset Option

The "Reset" button can be used to start a new game with a fresh random number.

This creates an engaging and interactive experience for the user.

Key Learnings

From this project, we learned:

GTK Library Mastery

How to use the GTK library to design a desktop GUI application in C.

Event Management

How to manage events and callbacks, such as button clicks.

Frontend-Backend Integration

How to connect frontend (UI) with backend logic in C programming.

Code Quality

The importance of modular coding, code readability, and user experience design.

> *And most importantly, how small projects like this can strengthen understanding of logic building, debugging, and event-driven programming.*

Conclusion

To conclude, this project gave us hands-on experience with GUI development in C, and helped me build a complete, interactive, and enjoyable application from scratch. we would like to thank our team for the cooperation and amazing support.

Thank you!