Hangman-Sockets Game

Project Proposal

Session: 2022-2026

Submitted by:

Ahsan Amin	2021-CS-634
Mubassir Ibrar	2022-CS-717
Rana Ahsan	2022-CS-730

Supervised by:

Miss. Maira

Department of Computer Science, New Campus

University of Engineering and Technology Lahore, Pakistan

Contents

CHAPTER 1: PROJECT SYNOPSIS

1.1 Abstract

This proposal outlines the development of a Hangman game using TCP sockets in Python. The project aims to create a network-based, multiplayer word-guessing game that leverages socket programming to enable remote gameplay. The proposal provides an overview of the project's scope, objectives, methodology, and tools used for implementation.

1.2 Introduction

The Hangman-Sockets project involves creating a digital version of the classic word-guessing game with a networked twist. By utilizing TCP sockets and Python's networking capabilities, the project introduces a novel approach to the traditional Hangman game, enabling players to interact over a network. The introduction highlights the innovative networking approach, the educational value of the project, and its contributions to understanding network programming and game development.

1.3 Problem Statement

The problem statement addresses the need for an engaging, network-based multiplayer game that demonstrates practical application of socket programming. It identifies the challenges of creating a real-time, interactive game environment using network communication, aiming to develop a robust and user-friendly Hangman game that showcases effective client-server architecture.

1.4 Objectives

- Develop a network-based Hangman game using Python and TCP sockets
- Implement a robust client-server communication system
- Create an interactive user interface for gameplay
- Develop game logic for word guessing, network communication, and game state management
- Ensure efficient handling of multiple concurrent client connections
- Implement secure and reliable data transmission using Pickle encoding/decoding

1.5 Features/Scope

- Network-based multiplayer Hangman gameplay
- Dynamic word input mechanism on the server-side
- Real-time game state synchronization
- Concurrent connection handling
- Visible progress tracking for word guessing
- Error handling and network communication robustness
- Cross-platform compatibility

1.6 Related Work

Existing network-based games and socket programming implementations will be reviewed. A comparative analysis will be conducted to identify strengths and weaknesses, guiding the development process of the proposed Hangman-Sockets game.

1.7 Proposed Methodology/System

The project will follow an iterative development approach, starting with requirements gathering, followed by design, implementation, testing, and deployment phases. The Hangman-Sockets game will be built using Python programming language with a focus on socket programming and concurrent network communication. Continuous testing and iterative refinement will drive the development process.

1.8 Tools and Technologies

- Python programming language
- TCP Sockets for network communication
- Pickle for data serialization
- Threading for concurrent connection handling
- Optional GUI library (e.g., Tkinter or PyQt) for client interface
- Git version control for collaborative development

1.9 Team Members Individual Tasks/Work Division

NAME	TASK
Ahsan Amin	GUI Design and Implementation
Rana Ahsan	Game Logic and Deployment
Mubassir Ibrar	Testing and Documentation

1.10 References

[1] *Kurose, J. F., & Ross, K. W. (2021). Computer Networking: A Top-Down Approach (8th ed.). Pearson. - Comprehensive resource for understanding network programming principles, socket communication, and client-server architectures.*

[2] *Gorelick, L., & Ozsvald, I. (2020). High Performance Python: Practical Performant Programming for Humans. O'Reilly Media. - Provides insights into Python's networking capabilities, concurrent programming, and performance optimization techniques.*

[3] *Bader, D. A. (2017). Python Network Programming Cookbook: Practical Solutions to Network Programming Problems. Packt Publishing.*