Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

COMP 342

Project Concept Note

On

"The Imposter Game"

Submitted by:

Lujaw Dhunju (19)

Jatin Madhikarmi (32)

Submitted to

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submission Date: January 9, 2026

Project Overview: The Imposter Game

1. Introduction

The Imposter Game is a digital implementation of a social deduction word game (inspired by games like *The Chameleon* and *Undercover*). It is designed as a local multiplayer "pass-and-play" experience where players use a single device to uncover a hidden traitor within their group.

The game blends verbal communication with digital secrecy. At the start of each round, a "Secret Word" is chosen from a specific category. While the majority of players (the "Civilians") are shown this word, one player (the "Imposter") is left in the dark.

The Primary Objectives:

- For Civilians: To describe the secret word vaguely enough to prove their identity to other players, but not so clearly that the Imposter can guess the word. Their ultimate goal is to identify and vote off the Imposter.
- For the Imposter: To listen to the descriptions, blend in by giving convincing clues, and deduce the secret word. The Imposter wins if they avoid being voted off or if they successfully guess the secret word at the end of the game.

2. **Language and Library**

To achieve a balance between logic handling and custom visual rendering, this project is developed using the following stack:

- Programming Language: Python
    - Python was chosen for its efficient handling of game states and its ability to manage dynamic lists of players and word libraries with ease.
- Graphics Library: Pygame

- Direct Rendering: Pygame allows for low-level control over the "Game Loop," giving us the ability to manage frame rates and event handling (mouse clicks, keyboard input).
- Transformations: The project utilizes Pygame's transform module to implement key Computer Graphics concepts, such as Scaling and Rotation matrices. These are used to create the visual "Card Flip" effect when players reveal their secret roles.
- Surface Manipulation: We utilize Pygame Surfaces to manage the UI layers, ensuring smooth transitions between the role-assignment screen, the voting phase, and the final game-over screen.

3. **Key Technical Focus (Computer Graphics)**

While the game is text-based in nature, the project focuses on the following graphical implementations:

- 2D Affine Transformations: Simulating 3D depth by manipulating the width and height of 2D sprites during the card-reveal phase.
- Dynamic UI Rendering: Real-time rendering of text and buttons based on player count and game state.
- Visual Feedback: Implementing color-blending and alpha-transparency (fading) for a polished voting and elimination sequence.