100 Feet Ring Road, Banashankari Stage III, Dwaraka Nagar, Bengaluru, Karnataka-560085

PYTHON FOR COMPUTATIONAL PROBLEM SOLVING

(UE25CSI5IA)

MONOPOLY

Team Members :

1.	Prajna Sai Prasad	- PES1UG25AM274 (P3)
2.	Pranav Khuba	- PES1UG25EC191 (P3)
3.	Pranav P Jahagirdar	- PES1UG25CS375 (P3)
4.	Prerana Sravan Kumar	- PES1UG25CS394 (P3)

PROBLEMSTATEMENT:

To design and develop a simplified beginner level digital version of the Monopoly game using programming concepts, where players can roll dice, move across the board, purchase properties, pay rent, and manage their money, thereby understanding basic game logic, control structures, and object-oriented programming concepts.

APPROACH:

The project uses a **GUI-based, turn-by-turn approach** to simulate the Monopoly game. Players roll dice, move across the board, and perform actions based on the tile landed. Game rules such as property purchase, rent payment, jail are handled using conditional logic. The board and player status are updated dynamically after each turn.

METHODOLOGY:

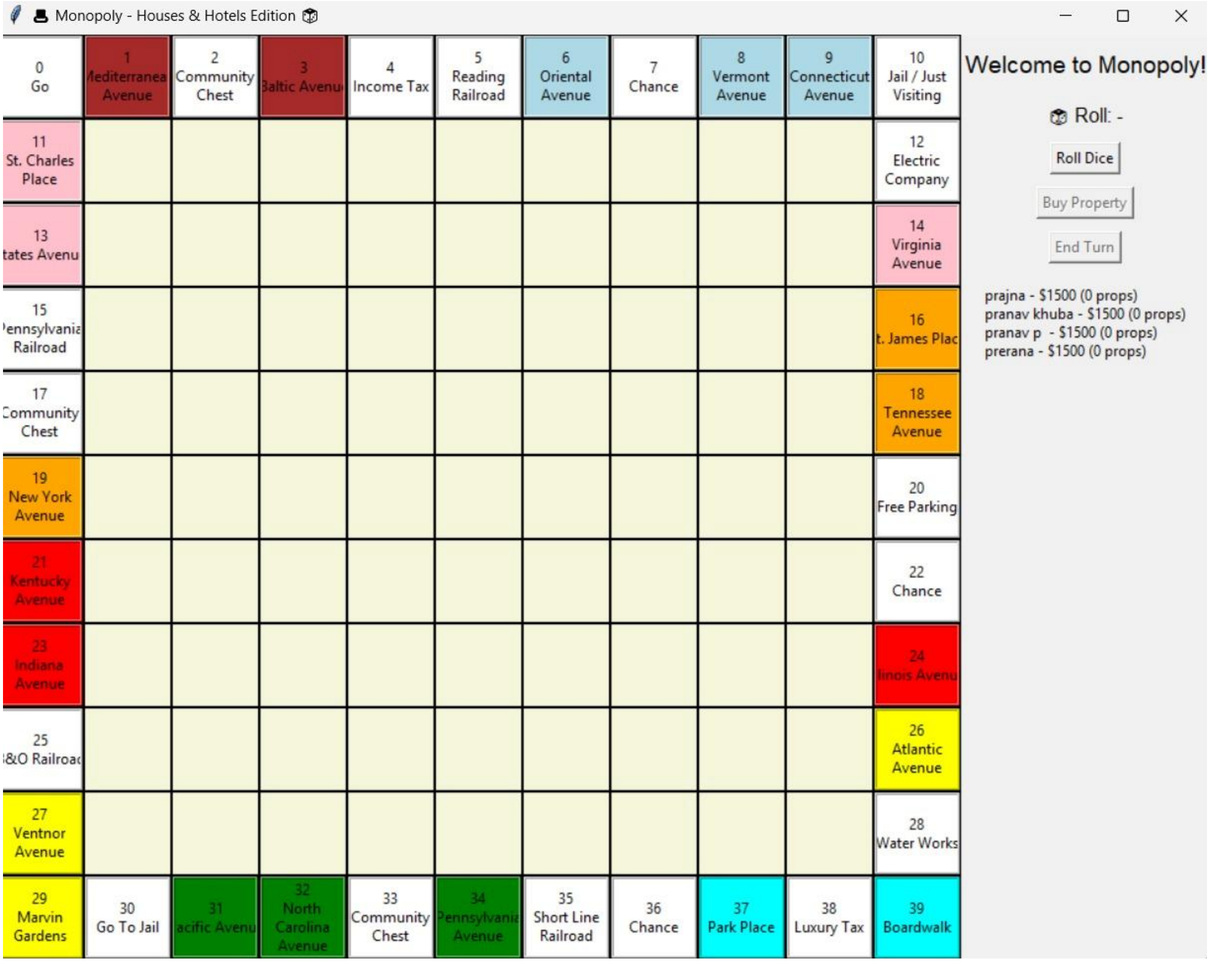
The project follows an **Object-Oriented and Event-Driven methodology**. Players and game logic are implemented using classes. User actions like rolling dice, buying property and ending turns are handled through GUI events using tkinter.

DATA STRUCTURES USED:

- **Lists** – To store board spaces, players, and buyable properties
- **Dictionaries** – To store property prices, ownership, house count, and color groups
- **Classes & Objects** – To represent players and game control logic
- **Set** – To store jailed players

- **Loops & Conditionals** – To control turns and apply game rules

SAMPLE OUTPUT :



Welcome to Monopoly!

Roll: -

prajna - $1500 (0 props)
pranav khuba - $1500 (0 props)
pranav p - $1500 (0 props)
prerana - $1500 (0 props)

CHALLENGES FACED :

- **Implementing game rules logically** was challenging, as Monopoly involves multiple conditions such as property ownership, rent calculation, jail rules, and card actions.
- **Managing turn-by-turn gameplay** for multiple players required careful control of loops and current player tracking.
- **Designing the GUI using Tkinter** was challenging due to layout management and updating the board dynamically after every move.
- **Handling property ownership and rent calculation** using data structures like dictionaries needed proper indexing and validation.
- **Debugging runtime errors and event-handling issues** was time-consuming, especially when multiple buttons and events interacted with the game logic.

SCOPE FOR IMPROVEMENT

- The game can be enhanced by implementing **complete Monopoly rules** such as trading properties, mortgages, auctions, and hotels.
- **AI-based computer players** can be added to allow single-player mode.
- The GUI can be improved with **better graphics, animations, and sound effects** for a more realistic gaming experience.
- **Saving and loading game progress** can be implemented so players can resume the game later.
- Additional features like **leaderboards, statistics, and customizable rules** can be included.
- Error handling and code optimization can be improved to make the application more efficient and user-friendly.