Java Code Explanation

Java Code Explanation

```
import java.util.*;
import java.util.stream.Collectors;

// Importing necessary packages from java.util library
class Player {
    String name;
    List<String> cards;

    // Player class with a name and a list of cards they hold
    Player(String name) {
        this.name = name;
        this.cards = new ArrayList<>();
    }

    // Method to add a card to the player's hand
    void addCard(String card) {
        cards.add(card);
    }
}

// Location class with a name attribute
class Location {
    String name;
```

```
    Location(String name) {
        this.name = name;
    }
}

// Room class with a name attribute
class Room {
    String name;

    Room(String name) {
        this.name = name;
    }
}

// Game class managing the game's logic and state
class Game {
    List<Player> players = new ArrayList<>();
     List<String> locations = Arrays.asList("Under Vase", "Secret Drawer", "Behind Picture", "Inside Box", "Under Table", "On Top of Closet");
    List<String> rooms = Arrays.asList("Greenhouse", "Billiard Room", "Study Room", "Living Room", "Bedroom", "Piano Room", "Dining Room", "Kitchen", "Library");

    Map<String, String> cards = new HashMap<>();

    // Initialize game by setting up players, shuffling cards, and distributing them
```

Java Code Explanation

```
void initializeGame() {
    players.add(new Player("Emma"));
    players.add(new Player("Liam"));
    players.add(new Player("Jack"));
    players.add(new Player("Sophia"));
    players.add(new Player("Emily"));
    players.add(new Player("Ella"));

    // Combining all cards into one list
    List<String> allCards = new ArrayList<>();
    allCards.addAll(players.stream().map(player -> player.name).collect(Collectors.toList()));
    allCards.addAll(locations);
    allCards.addAll(rooms);

    Collections.shuffle(allCards); // Shuffling all cards

    // Set aside one card from each category as the correct answer
    String correctPlayer = players.get(new Random().nextInt(players.size())).name;
    String correctLocation = locations.get(new Random().nextInt(locations.size()));
    String correctRoom = rooms.get(new Random().nextInt(rooms.size()));

    cards.put("Player", correctPlayer);
    cards.put("Location", correctLocation);
    cards.put("Room", correctRoom);
```

Java Code Explanation

```
    // Distributing remaining cards to players
    int index = 0;
    for (Player player : players) {
        while (player.cards.size() < 3 && index < allCards.size()) {
            String card = allCards.get(index++);
            if (!cards.containsValue(card)) {
                player.addCard(card);
            }
        }
    }
}

// Method to start the game
void startGame() {
    Scanner scanner = new Scanner(System.in);
    int currentPlayerIndex = 0;

    while (true) {
        Player currentPlayer = players.get(currentPlayerIndex);
        System.out.println(currentPlayer.name + "'s turn. Roll the dice (enter 'roll'): ");
        String input = scanner.nextLine();

        if (input.equals("roll")) {
            int diceRoll = rollDice();
            System.out.println("You rolled a " + diceRoll);
```

```
            // Implement movement logic here


            System.out.println("Enter your guess (format: Player,Location,Room): ");
            String guess = scanner.nextLine();
            String[] guessParts = guess.split(",");
            if (guessParts.length == 3) {
                checkGuess(currentPlayer, guessParts[0], guessParts[1], guessParts[2]);
            }
        }

        currentPlayerIndex = (currentPlayerIndex + 1) % players.size();
    }
}

// Method to roll a dice
int rollDice() {
    Random rand = new Random();
    return rand.nextInt(6) + 1;
}

// Method to check the player's guess
    void checkGuess(Player player, String guessedPlayer, String guessedLocation, String
guessedRoom) {
    boolean correctGuess = guessedPlayer.equals(cards.get("Player")) &&
```

```
            guessedLocation.equals(cards.get("Location")) &&
            guessedRoom.equals(cards.get("Room"));

        if (correctGuess) {
            System.out.println("Congratulations " + player.name + "! You found the hidden diamond!");
            System.exit(0); // End the game if the guess is correct
        } else {
            System.out.println("Incorrect guess. The game continues.");
        }
    }
}

// Main class to run the game
public class Main {
    public static void main(String[] args) {
        Game game = new Game();
        game.initializeGame();
        game.startGame();
    }
}
```