**Istanbul Atlas University**

**FACULTY OF ENGINEERING AND NATURAL SCIENCES**

**2024-2025 Spring Semester**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course title & Code | | Programming II (1400121012) | | |
| Instructor | | Dr. Muhammet ABDELATI | | |
| Teaching Assistants | | Muhammed Faruk Şahin | | |
| Department | | Computer Engineering | | |
| Section | | 2 | | |
| Student Full Name | | Kemal Mert Güney | | |
| Student ID | | 230508034 | | |
|  | |  | | |
| Assignment | Description | | Weight | Grade |
| P1 | Select a problem to solve with a GUI Java program and complete the following tasks: | | 1 |  |
| **Problem Description:**  Thoroughly describe the problem, identifying the inputs, required calculations, and expected outputs. | | 1 |  |
| **User Interface:**  Provide a screenshot of the user interface. Ensure the interface is intuitive, allowing users to easily understand what to enter and where to view the results. Specify the names of dynamic objects that have events and/or are updated during runtime. | | 1 |  |
| **Program Code:**  Write and include the complete program code. | | 2 |  |
| **Project Presentation: (3 points)**  Prepare a short video to present and demonstrate your project. | | 3 |  |
| **Video Upload: (1 points)**  Upload the video to YouTube. | | 1 |  |
| **Required Submittals:**   * A compressed file containing your NetBeans project and report.   **(Deadline: 02/03/2025 23:59:59)**   * A hard copy of your report.   **(Deadline: beginning of your lab session during the 5th week, 3-7/03/2025)** | | 1 |  |
| **Notes:**   * Your report must include this form as a properly filled cover page. In addition, it is expected to contain the problem description, user interface, well-formatted program code, and a link to your video. It should be handed as few A4 printouts stabled in the upper left corner. * If the video is not hosted on YouTube, you will lose 1 point. * Failure to meet both deadlines implies zero grade in this project. * Failure to meet only one deadline implies 50% deduction from your grade. | |  |  |
|  | | Total | 10 |  |

# **Number Prediction Game Documentation Kemal Mert Güney / 230508034**

## **1. Introduction**

The Number Prediction Game is a desktop application developed using Java Swing. The primary goal is to create an interactive game where the program generates a random number between 0 and 100, and the user must guess it. The game offers multiple difficulty levels and provides real-time feedback, making it an excellent project for understanding event-driven programming, GUI design, and basic algorithm implementation in Java.

## **2. Project Overview**

### **2.1 Objectives**

* **Develop a GUI-based application**: Utilize Java Swing to design a user-friendly interface.
* **Implement game logic**: Generate a random number and validate user input while providing hints.
* **Incorporate difficulty levels**: Offer three modes—Easy (unlimited attempts), Medium (10 attempts), and Hard (5 attempts).
* **Enhance user interaction**: Provide immediate feedback and include a restart option to replay the game.
* **Display comprehensive information**: Use an info button to guide the user on how to play the game.

### **2.2 Technologies Used**

* **Java**: The programming language used for development.
* **Swing (JFrame)**: For creating the graphical user interface.
* **Random Class**: For generating a random number.
* **Event Listeners**: Such as ActionListener to manage user interactions.

## **3. Application Features**

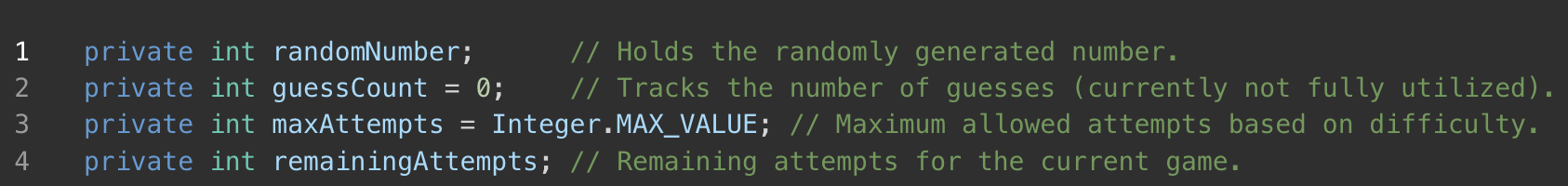
* **Random Number Generation**:  
  On startup, the game generates a random number between 0 and 100 using Java’s Random class.
* **User Input and Feedback**:  
  Users input their guess through a text field. The game compares this guess to the target number and provides hints:
  + If the guess is lower than the target, it displays: *"My number is higher."*
  + If the guess is higher, it displays: *"My number is lower."*
  + Upon guessing correctly, a congratulatory message is shown along with the number of attempts taken.
* **Difficulty Levels**:  
  A combo box allows users to select one of the three modes:
  + **Easy**: Unlimited attempts (simulated with Integer.MAX\_VALUE).
  + **Medium**: Limited to 10 attempts.
  + **Hard**: Limited to 5 attempts.
* **Game Reset and Information**:  
  The game includes a *Restart Game* button to reset the session and an *Info* button that provides detailed instructions about the game.

## **4. Code Structure and Detailed Explanation**

### **4.1 Main Class Overview**

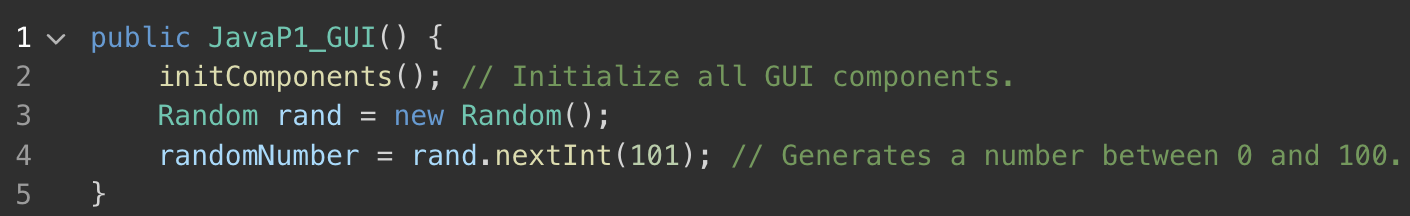
The project is encapsulated in the JavaP1\_GUI class, which extends javax.swing.JFrame. This class contains the GUI components, event handling, and game logic.

#### **Key Variables:**



### **4.2 Constructor and Random Number Initialization**

Upon instantiation, the constructor initializes the GUI and generates the random number:



### 

### 

### 

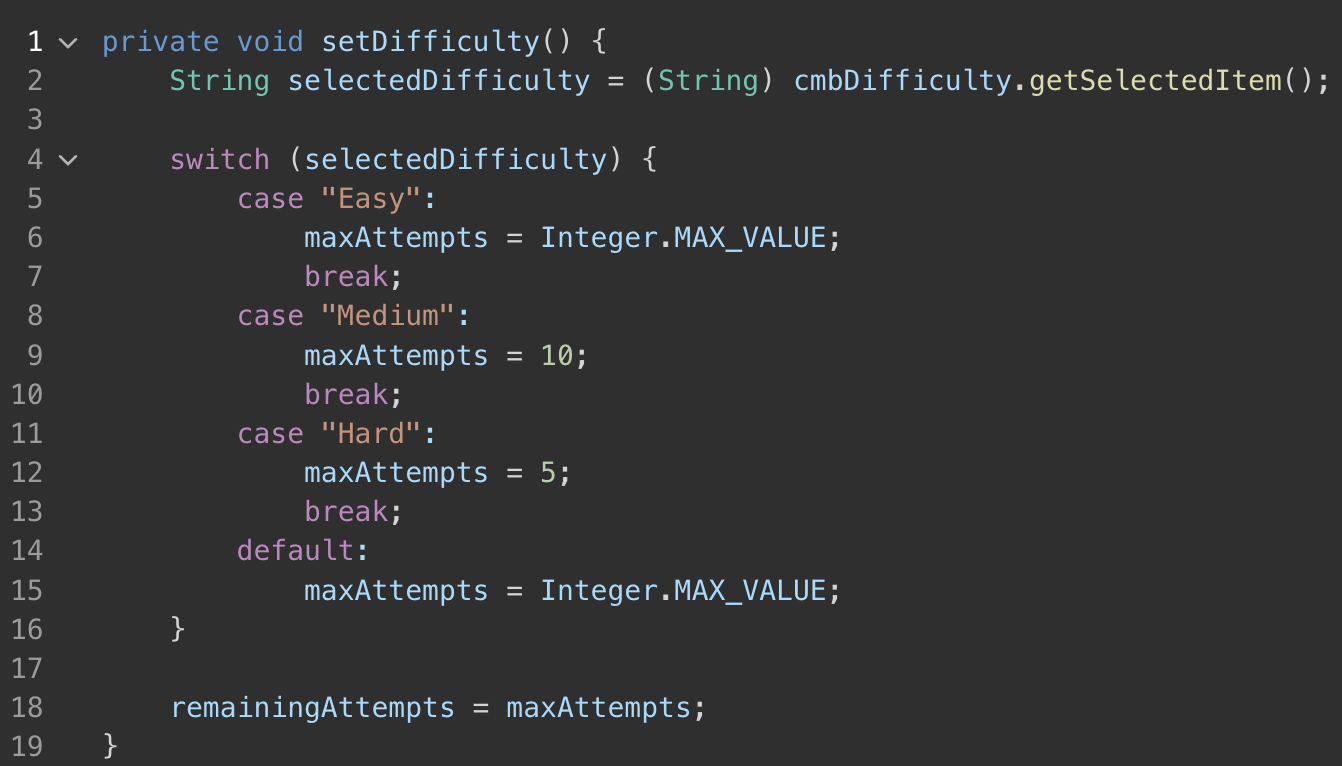
### 

### 

### 

### **4.3 Setting the Difficulty Level**

The setDifficulty() method sets the maximum attempts based on the user’s selection from the combo box:



*Explanation:*

* This method retrieves the current selection from the difficulty combo box and sets maxAttempts accordingly.
* The remaining attempts are reset to match the new maximum.

### 

### **4.4 Handling User Guesses**

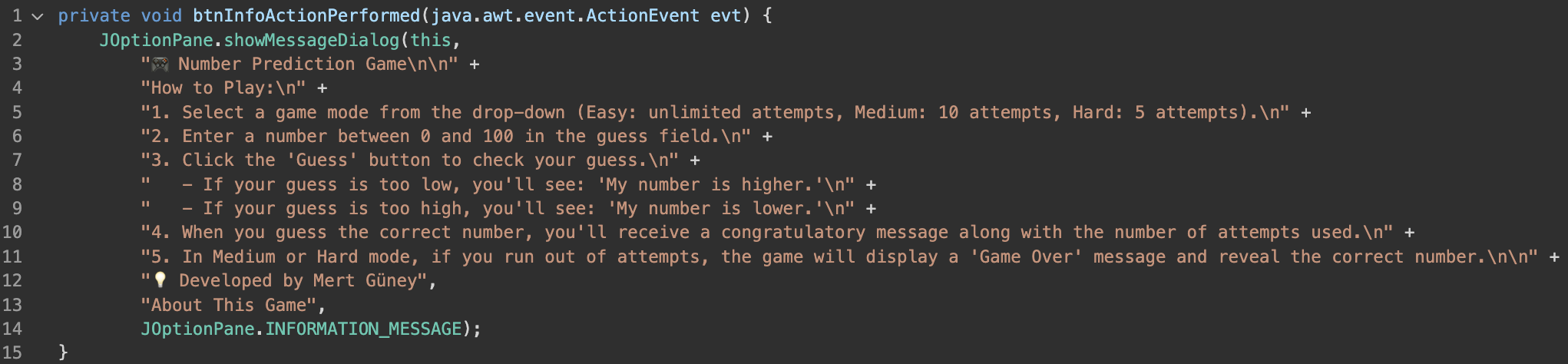
The btnGuessActionPerformed() method processes the user’s input, validates it, and provides feedback:

*Explanation:*

* The method first converts the user input into an integer.
* It checks if the guess is within the valid range and then provides hints or congratulatory messages based on the guess.
* The number of remaining attempts is updated, and if exhausted, a "Game Over" message is displayed.

### **4.5 Info Button Implementation**

The info button provides detailed instructions about the game. Below is the revised code block:

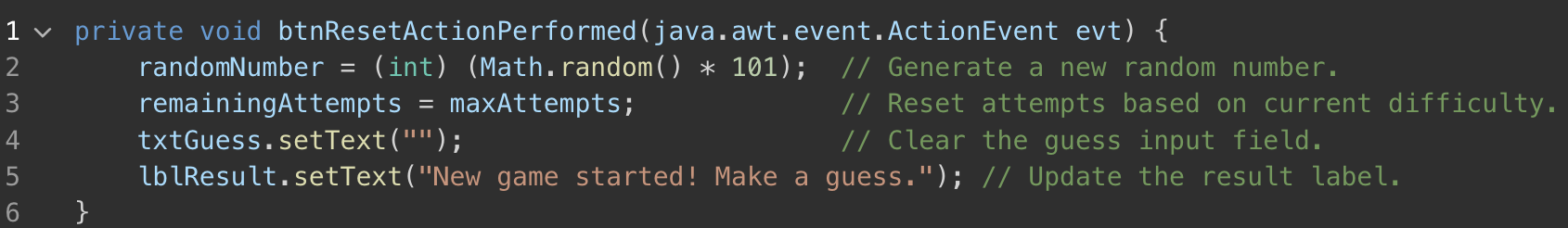


*Explanation:*

* This method uses JOptionPane.showMessageDialog() to display a detailed guide on how to play the game.
* It provides step-by-step instructions along with the difficulty mode explanations, ensuring that the user understands the gameplay before starting.

### **4.6 Restarting the Game**

The *Restart Game* button allows the user to start a new round by resetting the game state:



*Explanation:*

* When clicked, the method reinitializes the random number, resets the remaining attempts, and clears the user interface for a fresh start.

## 

## 

## **5. Graphical User Interface (GUI) Design**

The GUI was built using Java Swing components with a clear and organized layout:

* **Logo and Title**:  
  An image (e.g., Atlas University logo) is displayed on the left side, while the title “Atlas University Number Prediction Game” is prominently shown at the top.
* **Input Controls**:
  + A combo box for selecting the game mode (Easy, Medium, Hard).
  + A text field accompanied by a label prompting the user to "Guess a number between 0 and 100."
* **Action Buttons**:
  + **Guess Button**: Submits the user's guess.
  + **Restart Game Button**: Resets the game for a new round.
  + **Info Button**: Provides detailed game instructions.
* **Feedback Label**:  
  Displays hints, error messages, and success notifications based on user interaction.

This design ensures an intuitive and user-friendly interface, making the game accessible to all users.

## **6. Testing and Debugging**

During development, various techniques were employed to ensure robustness:

* **Input Validation**:  
  The code includes try-catch blocks to catch non-numeric entries and validate that the guess falls within the accepted range.
* **Console Debugging**:  
  Temporary console outputs (e.g., printing the random number, remaining attempts) were used during development to verify the correctness of the game logic.
* **User Experience Testing**:  
  The GUI was tested to ensure that all components are properly aligned and that the feedback is clear and immediate.

## **7. Areas for Future Improvement**

While the project meets its objectives, several enhancements could be implemented in future iterations:

* **Enhanced User Feedback**:  
  Display the remaining attempts directly on the GUI rather than only through console debugging.
* **Code Modularity**:  
  Separate game logic from GUI code into distinct classes or packages to improve maintainability.
* **UI Enhancements**:  
  Improve the visual design using modern UI components and theming libraries to offer a more polished experience.
* **Scoring and Leaderboard**:  
  Introduce a scoring system that tracks high scores and fastest guess times.

## **8. Conclusion**

The Number Prediction Game project demonstrates effective use of Java Swing for creating a graphical user interface, handling user input, and implementing game logic. This comprehensive project showcases skills in random number generation, event handling, and GUI design. The detailed documentation and code annotations reflect the effort and thought invested in creating a robust and user-friendly application.

metin, ekran görüntüsü, saat, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Presentation Link : https://www.youtube.com/watch?v=E0CTMgOj9VQ&ab\_channel=MertG%C3%BCney