

App Inventor Basic

Session 1: Calculator App

Objective: Introduce the basics of mobile app development and user interface design while building a functional calculator.

Content Overview:

1- Introduction to Mobile App Development:

- Explanation of mobile app fundamentals and their uses.
- Overview of the MIT App Inventor platform.

2 - Setting Up the Environment:

- Guide on signing in and navigating the App Inventor interface.
- Explanation of the main sections: Designer (UI) and Blocks (Logic).

3- User Interface Components:

- Detailed look at basic UI elements like buttons, textboxes, and labels.
- Customizing UI properties (size, color, text) to create a clear calculator layout.

4 - Working with Basic Arithmetic:

- Understanding how to perform arithmetic operations in blocks.
- Introduction to variables for storing and manipulating data.
- Displaying calculation results dynamically.

Session 2: Translator App

Objective: Introduce students to using functions and language handling to build a simple translator.

Content Overview:

1- Designing User-Friendly Interfaces:

- Best practices for creating a clear and intuitive user interface.
- Use of input boxes and selection controls (dropdowns or buttons) for language choices.

2- Introduction to Functions:

- What functions are and why they are essential in programming.
- How to use pre-built and custom functions in MIT App Inventor.

3- Handling User Input:

- Techniques for capturing and processing user-entered text.
- Displaying translations using labels or audio outputs (Text-to-Speech component).

4- Basic Translation Logic:

- Implementing simple translation logic for converting English to another language.
- Overview of using conditional statements to manage different languages.

Session 3: Paint App

Objective: Explore media components and interactivity by creating a paint application.

Content Overview:

1- Working with Layouts:

- Understanding different layout options (horizontal, vertical, table layouts).
- Organizing UI elements to create a clean and accessible interface.

2- Media Components:

- Introduction to media elements, including the Canvas component.
- Using images, sounds, and other media for interactive apps.

3- User Interaction and Drawing:

- Making the app responsive to user actions (touch and drag events).
- Implementing color selection for the paint tool.
- Exploring how the Canvas component can be used to draw lines, shapes, and more.

Session 4: Balloon Pop Game

Objective: Develop a simple game with interactive elements, focusing on timers, variables, and user feedback.

Content Overview:

1- Game Design Fundamentals:

- Basics of game design and creating engaging user experiences.
- Structuring the game layout with a clear play area for popping balloons.

2- Timers and Speed Management:

- Introduction to the Timer component for controlling time-based events.
- Setting up variables to manage balloon speed and game difficulty.

3- Using Arithmetic and Logic for Games:

- How arithmetic operations can control game elements like speed.
- Tracking scores using variables and updating them as the game progresses.

4- Game Feedback:

- Providing visual and audio feedback to enhance gameplay (e.g., popping sound effects).
- Displaying the score dynamically as balloons are popped.

Sessions 5 & 6: Snake Game

Objective: Create a classic snake game, focusing on gameplay mechanics, control flow, and condition handling.

Session 5:

Content Overview:

1- Designing the Snake Game Layout:

- Understanding the importance of layout for game clarity.
- Setting up the game grid and visual components for the snake and target.

2- Introduction to Control Flow:

- Basics of controlling movement using arrow keys or touch gestures.
- Implementing initial snake movement logic with direction controls.

3- Using Variables to Manage Game State:

- Introduction to variables for storing snake position, direction, and target.
- Managing game state with variables to track the snake's length and position.

Session 6:

Content Overview:

1- Completing the Game Logic:

- Handling collision detection for when the snake eats a target or hits a wall.
- Implementing scoring logic to increase the score as targets are collected.

2- Game Over Conditions:

- Creating a game-over screen that appears when certain conditions are met.
- Displaying final scores and giving the player options to restart the game.

3- Fine-Tuning Gameplay:

- Adjusting speed, difficulty, and control responsiveness for better gameplay.
- Ensuring smooth transitions and proper scoring updates.