Guessing Game

CS501

Yol-Emma Veillard

Game Mechanics

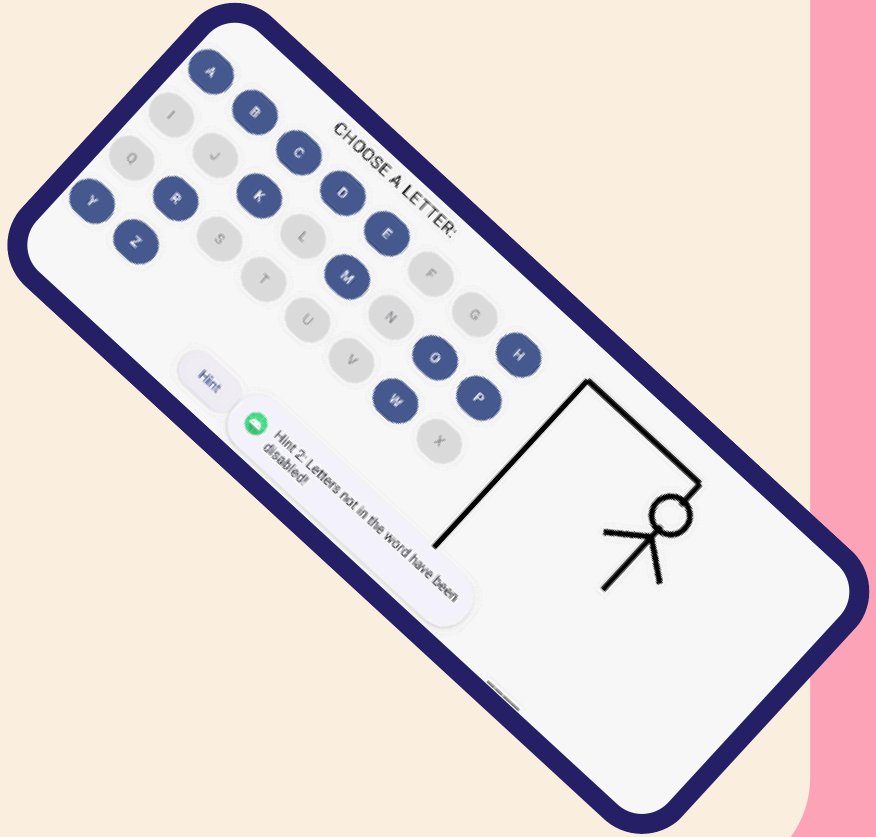
- A word is randomly selected, and the player is shown blank spaces representing the letters of the word.
- The player guesses letters one at a time from an on-screen keyboard.
- Correct guesses fill in the corresponding blanks, while incorrect guesses result in a body part being added to the figure.
- The game ends when the player either guesses the word or the Hangman figure is fully removed after six incorrect guesses.
- The player has a maximum of 3 hints. Each hint counts as an incorrect guess, so a part of the figure is added.

Core Features

- Drawing: Highlight how the figure is drawn and parts of the body are removed for each wrong guess. Mention the modular way you approached drawing the body (e.g., legs, arms, torso).
- Real-time Feedback: The game immediately updates to show whether a guess was correct or incorrect. The state of the word and the hangman drawing are updated in real-time as guesses are made.

Technical Highlights

- State Management
  - ViewModel
- Compose UI
  - HangmanPanel()
  - LetterPanes()
  - HintPane()



Hint 2: Letters not in the word have been disabled!

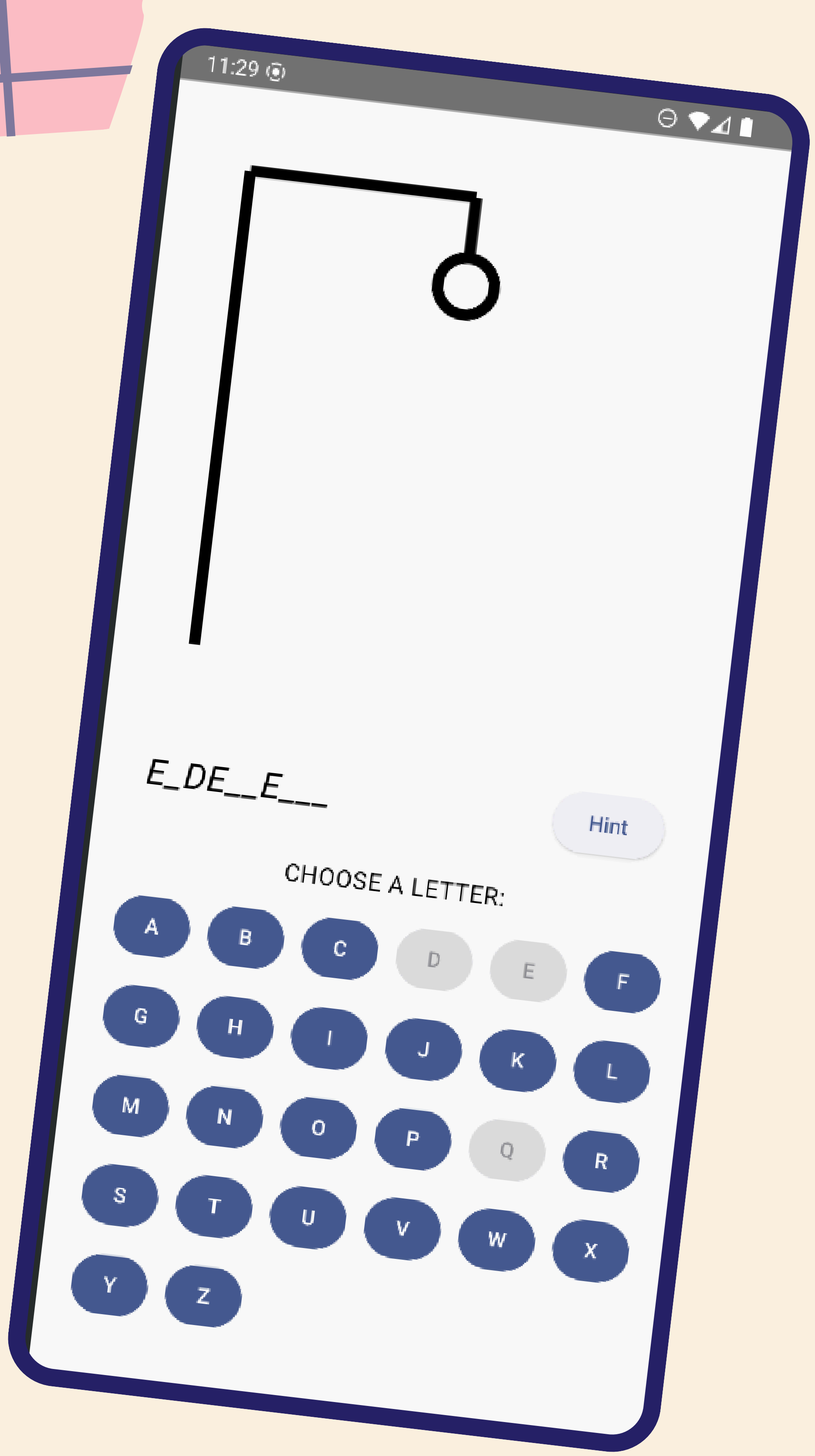
Challenges & Solutions

- State Management & Interaction:
  - Challenge: Managing game state (word to guess, letters used, wrong guesses) efficiently.
  - Solution: Used Jetpack Compose’s state handling via remember and collectAsState functions to manage UI updates, and ViewModel to hold the game's state and logic outside of the Composables for better separation of concerns.

Congrats! You guessed the word: PEAR! :D

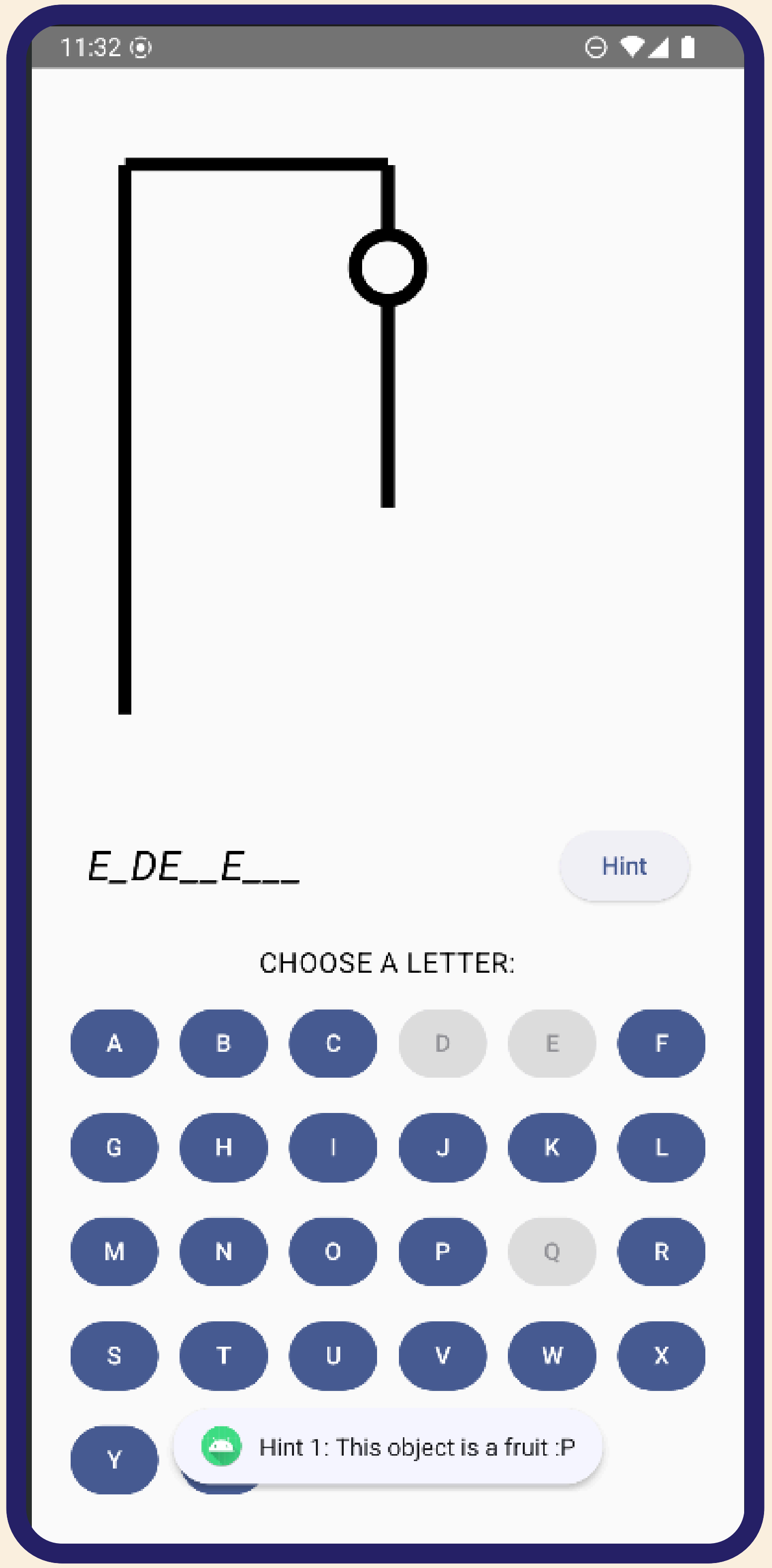
- Responsive Design:
  - Challenge: Handling different screen sizes, especially wide screens vs narrow screens (landscape vs portrait orientation), while keeping the UI functional and visually appealing.
  - Solution: Used Modifier functions like .fillMaxWidth() and .weight() to ensure that elements are scaled proportionally across different screen sizes.
    - WideScreenLayout for landscape mode, where the elements are displayed in a horizontal arrangement.
    - NarrowScreenLayout for portrait mode, where elements are stacked vertically.

App Demo



E_DE__E___

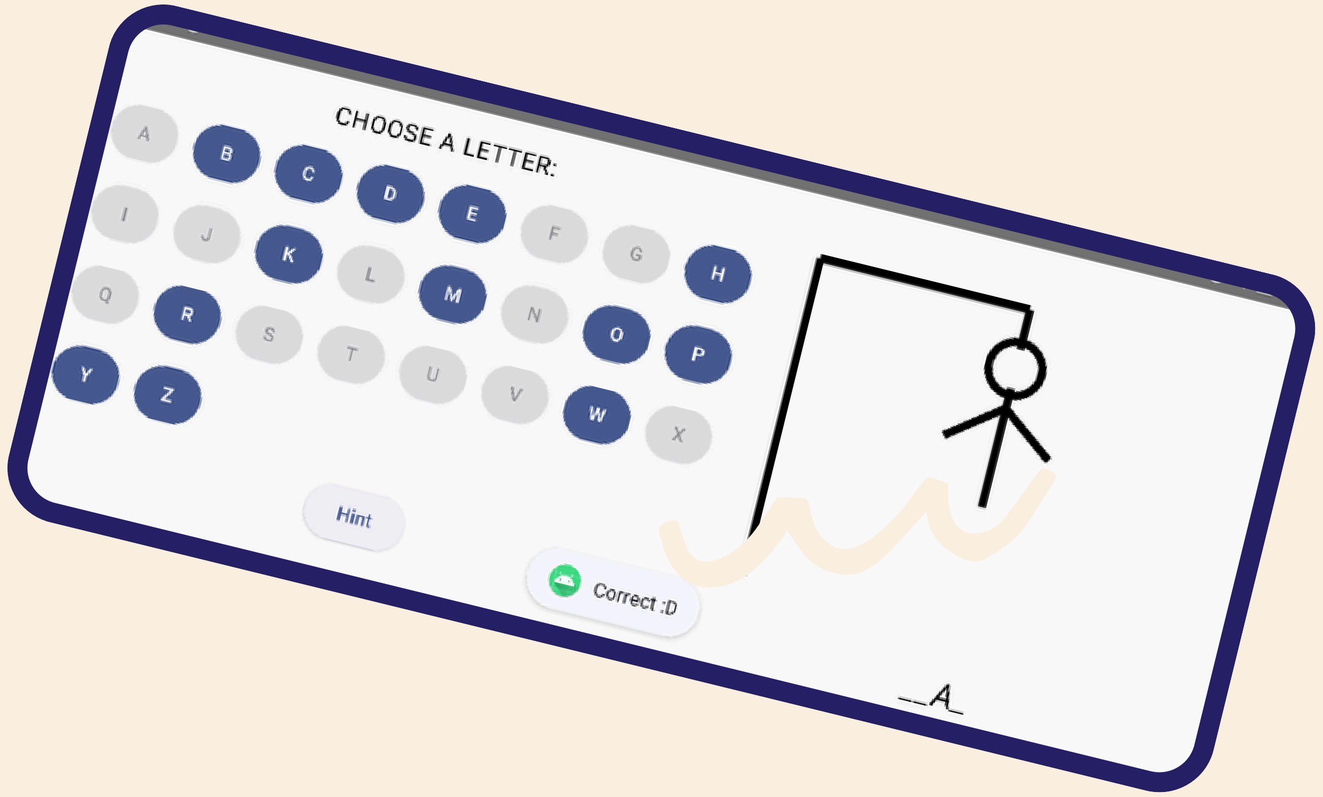
CHOOSE A LETTER:



E_DE__E___

CHOOSE A LETTER:

Hint 1: This object is a fruit :P



CHOOSE A LETTER:

Correct :D

__A_