Session 11 — Extending Your Game

Choose Your Adventure 🎯

You've built a complete trivia game and deployed it to the internet — now it's time to make it uniquely yours! Ready to add your personal touch and practice new skills? Let's go!

Table of Contents

☁️ Accessing Your Codespace

Visit github.com/codespaces to relaunch your Codespace from Session 10.

🎯 Choose Your Own Adventure

Choose **up to 4 extensions** to implement — try 2 Starter + 1 Power + 1 Beast, or mix and match based on your interests.

Use the GitHub Copilot Workflow to get AI help with implementation, and the Git Workflow to save and deploy your changes.

🚀 Starter Mods: Fast Wins

Extension	Description	What You'll Practice
Custom Fonts	Replace existing fonts with your choices for game logo, zone labels, and UI elements	Importing Google Fonts, updating component styling, creating visual consistency

Character Swap	Replace character-1.png with character-2, 3, 4, or 5 from available sprites	Working with image assets, updating component references, testing visual changes
Game Instructions Modal	Create instructions modal explaining game rules and scoring	Building reusable components, showing/hiding content, creating user-friendly interfaces

⚡ Power Mods: Build Skills

Extension	Description	What You'll Practice
Visual Progress Indicator	Replace "Question X of Y" text with colored dots showing progress	Managing component state, transforming arrays into visual elements, styling based on user actions
Color & Font System	Convert hardcoded colors/fonts in main.css to reusable CSS variables	Organizing stylesheets, creating reusable design tokens, making code easier to maintain
Interactive Sound Effects	Add audio feedback for quiz interactions using existing useAudio pattern	Using custom hooks, working with browser audio, triggering sounds on user interactions

🦁 Beast Mods: Go Big

Extension	Description	What You'll Practice
Character Selection Screen	Add character selection screen before entering game map	Building new screens, managing game-wide state, composing complex interfaces
Animated Character Movement	Animate character sprite moving between zone positions when completed	Creating smooth animations, coordinating with game state, providing visual feedback
Dark/Light Mode Toggle	Add dark/light mode toggle affecting colors throughout game	Building theme systems, persisting user preferences, creating toggleable interfaces

⚡ GitHub Copilot Workflow

Use this workflow from Session 9 for each extension:

How to Use Copilot Chat Effectively

1. Use a Copilot chat command like `/fix`, `/explain`, or `/test`
2. Write a clear, focused prompt describing what you want
3. Review the suggestion Copilot generates
4. Apply the change if it meets your needs
5. Test the update to confirm it works

Example Prompt Sequence

```
/fix Import Orbitron, Roboto, and Inter from Google Fonts into fonts.css.
```

followed by

```
/fix Update all fontFamily references:
- Orbitron for game logo title
- Roboto for zone map labels
- Inter for UI elements
```

Good Prompts Follow the 3S's:

- **Simple:** Use clear, direct language
- **Specific:** Say exactly what you want
- **Short:** Focus on one task at a time

🔄 Git Workflow

Every time you finish an extension, use this Git workflow to save and deploy your changes:

The Essential Commands

```
# 1. Stage your changes (prepare them for committing)
git add .

# 2. Commit your changes (create a permanent snapshot)
git commit -m "feat(extension): add custom fonts to game logo"

# 3. Push your changes (share with the world and trigger deployment)
git push
```

Commit Message Examples

Extension Type	Example Message
Custom Fonts	`feat(typography): update Google Fonts for game logo`
Character Swap	`feat(character): replace default sprite with archer`

Visual Progress Indicator	`feat(quiz): add visual progress indicator with dots`
Interactive Sound Effects	`feat(audio): add feedback sounds for answers`
Character Selection Screen	`feat(character): add character selection screen`
Animated Character Movement	`feat(animation): animate character movement between zones`
Dark/Light Mode Toggle	`feat(theme): implement dark/light theme system`

Your live game will automatically update each time you push changes thanks to the CI/CD pipeline you set up in Session 10.