

Control Structures

If-Else Statements for RPG Game Project

1. Player Health Check:

- Given a variable `playerHealth` (type: number), write an if-else statement that checks if `playerHealth` is greater than 10. Log "Player is strong" if true, and "Player is weak" otherwise.

2. Player Gold Range Check:

- Given a variable `playerGold` (type: number), write an if-else statement that checks if `playerGold` is between 1 and 100, inclusive. Log "Rich player" if it is, otherwise log "Poor player".

3. Nested If-Else for Weather and Time:

- With two variables, `weather` (type: string) and `timeOfDay` (type: string), create a nested if-else statement to determine if the player should "Continue the adventure" or "Seek shelter".

4. Comparing Weapon Strength:

- Compare two variables, `playerWeapon` (type: string) and `enemyWeapon` (type: string), for equality using an if-else statement. Log "Weapons clashing!" if they're the same, "Advantage" if the player's weapon is different.

5. Check Player Age for Game Content:

- Given a variable `playerAge` (type: number), use logical operators in an if-else statement to determine if the player is between 13 and 19 years old.

6. Game Score Evaluation with Ternary Operator:

- Use the ternary operator for a variable `gameScore` (type: number) to check if the score is greater than or equal to 50. Log

"High score" if true, and "Low score" otherwise.

7. Check Number of Enemies (Even or Odd) with Ternary Operator:

- Use the ternary operator to check if a variable `numOfEnemies` (type: number) is even or odd. Log "Even number of enemies" if it's even, and "Odd number of enemies" otherwise.

8. Character Age Group Classification:

- Use the ternary operator to classify a character's age variable `characterAge` (type: number) into "Youngling" (less than 12), "Adolescent" (between 13 and 19), or "Adult" (20 and above).

9. Player's Battle Skill Evaluation:

- Use the ternary operator for a variable `battleSkill` (type: number) to evaluate. Log "Skilled warrior" if `battleSkill` is greater than 70, otherwise log "Train harder" .

10. Game Session Duration Category:

- Given a variable `sessionDuration` (type: number, representing minutes), write an if-else statement that categorizes game sessions as 'Short' (less than 90 minutes), 'Moderate' (90 to 150 minutes), or 'Marathon' (more than 150 minutes).

11. Check Potion Availability in Inventory:

- With variables `totalPotions` (type: number) and `usedPotions` (type: number), write an if-else statement to check if there are potions available. Log 'Potions available' or 'Out of potions' , accordingly.

12. Validate Player Age for Restricted Quest:

- With variables `playerAge` (type: number) and `questAgeRestriction` (type: number), write an if-else statement to validate if the player is 18 years old or more. Log appropriate messages based on the age restriction.

13. Special Weekend Events:

- Given a variable `dayOfTheWeek` (type: string), write an if-else statement to check for special weekend events. Log `'Special event happening!'` for `'Saturday'` or `'Sunday'`, otherwise `'Regular gameplay'`.

14. Identify Era of Game Setting:

- Given a variable `gameSettingYear` (type: number), write an if-else statement to determine if the game setting is in the medieval era. Log `"Medieval era setting"` if the `gameSettingYear` falls between 900 and 1500 (inclusive), otherwise log `"Different historical era"`.
- Example input: `gameSettingYear = 1200`

15. Evaluate Character Class and Attributes:

- Given two variables, `characterClass` (type: string) and `characterLevel` (type: number), create an if-else statement to assess the strength of a character based on their class and level. If the `characterClass` is `"Warrior"` and the `characterLevel` is 10 or higher, log `"Powerful warrior"`, otherwise log `"Average character"`.
- Example inputs: `characterClass = "Warrior"`,
`characterLevel = 15`

16. Determine Battle Readiness:

- Given two variables, `playerHealth` (type: number) and `enemyCount` (type: number), write an if-else statement to determine if the player is ready for battle. Log `"Ready for battle"` if the `playerHealth` is above 50 and the `enemyCount` is less than 3, otherwise log `"Proceed with caution"`.
- Example inputs: `playerHealth = 70`, `enemyCount = 2`

17. Assess Quest Completion Status:

- Given two variables, `questProgress` (type: number) and `questDifficulty` (type: number), create an if-else statement to evaluate the completion status of a quest. If the `questProgress` is

100 and the `questDifficulty` is less than or equal to 5, log

"Quest completed", otherwise log "Quest in progress".

- Example inputs: `questProgress = 100`, `questDifficulty = 3`

18. **Check Player's Inventory:**

- Given a variable `inventorySize` (type: number), write an if-else statement to check the status of a player's inventory space. Log "Inventory almost full" if the `inventorySize` is greater than or equal to 90, otherwise log "Plenty of space in the inventory".
- Example input: `inventorySize = 80`