# Game Structure: CodeBot Quest

## Main Story (expanded on the DOC “STORYLINE AND SCRIPT: CODEBOT QUEST”)

In the year 3025, *CodeBot*, a young robot, embarks on a quest to fix and rebuild his world after a rogue AI virus corrupts the system. The only way to fix it is by mastering the basics of coding, solving programming challenges, and defeating virus bosses.

## Core Gameplay

* Players control a friendly robot navigating through different levels.
* Each level teaches a different programming concept.
* Gamification elements: life system, rewards, and knowledge quizzes.

## Game Mechanics & Motivational Features

* **Life System**: Players start with three lives per level.
* **Scoring System**: Points for completing tasks and solving coding problems.
* **Hints & Feedback**: The game provides hints when players struggle.
* **Power-ups**: Speed boosts, extra jumps, and temporary shields.
* **Unlockable Skins**: New robot outfits upon completing levels.
* **Scoring System**: Points for correct code execution and bonus rewards for efficiency.

## Accessibility Features (expanded on the DOC “ACCESSIBILITY FEATURES: CODEBOT QUEST”)

* **Dyslexia-friendly fonts** (OpenDyslexic).
* **Colorblind-friendly design**.
* **Audio cues** for visual elements.
* **Adjustable difficulty levels**.

## Game Progression & Levels

Each level will introduce new programming concepts, gradually increasing in complexity.

### Level 1: Basics of Movement (Sequential Commands)

* **Objective**: Teach players how to issue basic commands.
* **Concepts Covered**: Sequential execution, simple movement.
* **Gameplay**: Players control the robot using basic directional commands (left, right, jump).
* **Challenge**: Reach the goal while avoiding obstacles.

### Level 2: Variables and Data Storage

* **Objective**: Introduce variables to store and modify values.
* **Concepts Covered**: Assigning values, using variables.
* **Gameplay**: Players collect energy cells while controlling the robot.
* **Challenge**: Solve small puzzles using stored values.

### Level 3: Loops (Repeating Instructions)

* **Objective**: Teach loops for repeated actions.
* **Concepts Covered**: for and while loops.
* **Gameplay**: Players program the robot to repeat actions.
* **Challenge**: Create a loop to collect multiple items efficiently.

### Level 4: Conditional Statements (Decision Making)

* **Objective**: Teach if-else logic.
* **Concepts Covered**: Basic conditional logic.
* **Gameplay**: The robot must decide which path to take based on conditions (e.g., if there is an obstacle, jump).
* **Challenge**: Players write simple conditions to navigate through obstacles.

### Level 5: Functions (Reusability of Code)

* **Objective**: Teach function creation.
* **Concepts Covered**: Writing and calling functions.
* **Gameplay**: Players create reusable movement functions.
* **Challenge**: Use functions to solve a maze efficiently.

### Level 6: Debugging (Problem Solving)

* **Objective**: Identify and fix errors in a code snippet.
* **Concepts Covered**: Debugging strategies.
* **Gameplay**: Players troubleshoot pre-written code that has errors.
* **Challenge**: Fix the robot’s path by correcting logic errors.