1. Game Planning and Design
   - Duckling - Character
   - Anshul - Sage who guides the duckling.
   - Starts with Animation about the DSA World.
   - World 1, 2, 3, 4, 5
   - Each world - 2 Levels.
   - The duckling will move according to the theme of the level e.g Arrays - Binary Search.
   - The duckling can smash - solve a question (MCQ) - and get a coin.
   - The duckling will face obstacles - CodeSmells/bugs if solved, gets a coin. If not, they lose their heart.

   - World 1 || Basics of Programming :
     - The theme is a code editor with an array block setup where the duckling will be running.
     - Level 1: Arrays
       - Array blocks are the platforms. Clear blocks are passable, and filled blocks must be smashed.
       - Smashing a filled block prompts a question, and solving it will earn a point (Leetcode coin).
       - Obstacles will prompt a question, and solving it will earn a point, while failure costs a heart.
     - Level 2: Strings
       - Instead of collecting coins, the duckling will collect characters.
       - Each time a character is collected, it gets added to the result string.
       - Completing the result string successfully finishes the level.

   - World 2 || Linear Data Structures:
     - The theme is garden setup with enemy mushrooms which can only be beaten when attacked with larger size than the numbers floating above.

     - Level 1 : Stacks and Queues:
       - The new stack of enemy duckies means that the duckling has to attack the top one in order to move past to the bottom.
       - Wrong move or enemy duckies attack the duckling -> lose a heart
       - If the enemy duckies were killed in Fifo order, doubling-size power-ups would be granted, which would later help in killing those duckies and gaining coins.

     - Level 2 : LinkedList:
       - New blocks with starting block having value on top of LL, and succedding nodes with addr value on top with pointer value inside

- Start with head node, pop the next node add value, choose the correct next node.
- Successful arrival at the end of LL grants Bug Smasher power up.

- World 3 || Trees and Hierachial Structures:
  - The theme is Data Forest with tree structures, each representing different tree types like binary trees, AVL trees and Red-Black trees

  - Level 1 : Traversals:
    - **Preorder Region (Root → Left → Right):** The player begins at the root node of a towering tree. The mushrooms blocking the path must be defeated in the Preorder sequence. The player starts by attacking the root, then moves to the left side, and finally the right.
    - **Inorder Region (Left → Root → Right):** A balanced tree with branching paths. Players must defeat the mushrooms in Inorder sequence, starting from the leftmost branch, then the root, and finally the rightmost.
    - **Postorder Region (Left → Right → Root):** A dense forest where all mushrooms on the left must be defeated before moving to the right, and the root boss at the end must be tackled last.

  - Level 2 : Balancing Trees:
    - Given an unbalance tree, should be adding or removing nodes.
    - Fix an unbalanced tree by adding or removing nodes. Clear mushrooms blocking paths. Correct balancing grants a power-up. Restore balance quickly or face stronger enemies and decay damage.

- World 4 || Graphs:
  - The theme revolves around an aquatic landscape, featuring islands that symbolize nodes on a graph.The objective is to navigate through this water-themed terrain, connecting the islands in an effective manner to achieve overall success.

  - Level 1 : Traversals and shortest path:
    - BFS Mechanics: Duckie explores nodes level-by-level, with glowing paths indicating the shortest unvisited neighbors. Rewards include coins for visiting all nodes efficiently under a time limit.
    - DFS Mechanics: Duckie dives deep along trails to leaf nodes before backtracking. Dead ends add challenge, and rewards include treasure at leaf nodes and bonuses for minimizing backtracking.

- Dijkstra’s Algorithm: Duckie navigates weighted paths to find the least-cost route to the treasure. Optimal paths are highlighted, and competing against an AI rival adds urgency.
- Prim’s Algorithm: Duckie connects nodes by selecting the minimum-weight edges dynamically. Players avoid unnecessary moves and earn coins for completing an MST.
  Or
- Kruskal’s Algorithm: Duckie selects edges in ascending weight order to form an MST without creating cycles, with visual guidance for legal moves and bonuses for efficient connections.

- Level 2 :Graph Node Coloring and Bipartiteness
  - Duckie must traverse the graph to determine if it is bipartite by attempting to virtually "color”,nodes during exploration, ensuring no two adjacent nodes share the same color.
  - Duckie marks nodes dynamically as it explores, with visual feedback for conflicts (e.g., red-glowing edges if adjacent nodes cannot be colored distinctly).
  - Players navigate strategically to test bipartite properties, with penalties for errors or backtracking and rewards for completing the exploration without conflicts.
  - Coins are earned for accurate exploration, and a summary screen visualizes the graph's bipartite status, highlighting correctly marked nodes and any conflicts.

- Level 3: Heap:
  - Duckie organizes a list of scattered numbers into a binary heap, with visual feedback ensuring the heap property (min-heap or max-heap) is maintained. Bonus rewards for efficiency.
  - Duckie adds new elements to the heap, triggering "heapify-up" animations as nodes adjust. Points awarded for completing insertions correctly.
  - Players remove the root node, with "heapify-down" animations restoring the heap structure. Rewards given for handling deletions without errors.
  - Timed puzzles with multiple insertions and deletions test players’ ability to maintain the heap property under pressure.

- World 5 || Algorithms:
  - Theme: Optimize paths through a tree by choosing the best route.
  - Level 1 : Sorting algorithms:
    - Bubble Sort: Duckie moves along an array of numbered platforms, comparing adjacent numbers. If the numbers are out of order, Duckie swaps them by jumping between them. The largest number moves towards the end of the array as the process repeats.

- Selection Sort: Duckie selects the smallest (or largest) number from the unsorted part of the array and moves it to the correct position by dragging it to the front. The player must navigate Duckie across the array, identifying and moving the correct number.
- Merge Sort: Duckie splits the array of numbers into smaller sections, visually breaking it into halves. Then, Duckie merges these sections back in sorted order, placing numbers in the correct spots as they combine. The array visually updates as smaller groups of numbers join together to form a fully sorted array.
- Insertion Sort: Duckie walks along an array of numbers, picking up each number and inserting it into the correct position in the sorted section of the array. As Duckie places each number, the remaining numbers on the right shift to make room.

- Gameplay: Use DP to calculate the shortest or most rewarding path, avoiding costly branches and enemies.
- Objective: Reach the end with the least cost.