Reichman University

PDF Document for Project 9 Submission – Simon Game

Benjo Inbar, Peron Yuli

1. Concept / Idea of our game:

Simon Shapes is a single-player memory game based on the classic Simon game. Instead of colors, players must memorize and replicate sequences of shapes: Circle, Triangle, Square, and Arrow. The game offers 15 levels of increasing difficulty, with players required to recall longer sequences as they progress. The objective is to accurately reproduce the sequence of shapes presented on the screen by pressing the corresponding keys on the keyboard.

2. Architecture:

Explanation of Key Components:

- **Main.jack:**
  - **Purpose:** Serves as the main entry point and game controller.
  - **Responsibilities:** Initializes the game and starts the main game loop. Creates the game instance and manages program execution flow.
- **SimonGame.jack:**
  - **Purpose:** Implements the gameplay logic and manages levels.
  - **Responsibilities:** Controls the main game logic and manages the sequence of shapes and player interactions. Handles game state, level progression, and coordinates between different game components.
- **GameBoard.jack:**
  - **Purpose:** Manages visual elements of the game.
  - **Responsibilities:** Manages all visual elements of the game board including grid layout, shapes, and UI elements. Handles the drawing of shapes during player input and maintains game's visual interface.
- **GameFrames.jack:**
  - **Purpose:** Handles framing and highlighting specific shapes.
  - **Responsibilities:** Responsible for drawing and clearing frames around shapes during sequence display. Creates visual feedback by highlighting current shapes in the sequence through rectangular borders.

- **Random.jack:**
    - **Purpose:** Generates random shape indices for sequences.
    - **Responsibilities:** Implements a bag-based random number generator for fair shape distribution in the game. Uses shuffle algorithm to ensure balanced and unpredictable sequences of shapes.

- **RandSeed.jack:**
    - **Purpose:** Supplies a seed for the random number generator.
    - **Responsibilities:** Displays initial game instructions and generates a random seed based on player input timing. Sets up the starting conditions for the random number generation.

Key OS Class Integration:

- **Keyboard:** Detects player input, mapping keys to shapes:
    - 1 for Circle
    - 2 for Triangle
    - 3 for Square
    - 4 for Arrow
- **Screen:** Displays the sequence visually and highlights shapes.
- **Output:** Prints messages and feedback, such as game instructions and results.

3. Motivation:

We chose to create Simon Shapes because it combines memory skills and quick thinking, making it both engaging and challenging. The game offers an engaging way to demonstrate core programming concepts like memory management, user interaction, and timing control, while remaining simple enough to implement effectively in Jack.

4. Google Drive link to our video:

Link to the game explanation video.

5. Names and Emails:

Name :Benjo Inbar
Email: inbar.bengio@post.runi.ac.il

Name: Peron Yuli
Email: yulilidya.peron@post.runi.ac.il

Reichman University