2 Advanced topics of choice:

1. Advanced Object-Oriented Programming with Design Patterns. For example: Using the State Pattern to manage the overall application flow, switching between the menu, level select (map) screen, and the main game. A second State Machine is used within the arena to manage the match's flow and keep track of the game. The Observer Pattern is used for UI updates and observing the game state.

2. User Experience (UX). For example: different themes per stage (boat, beach, desert, forest, castle) with unique visuals (from assets/images), a "Back" button for navigation, "You Win" / "You Lose" screens for clear feedback, music (from assets/audio), and sound effects that give instant feedback for hits with the racket and table.

Name: Game window with Swing

Description: Establishes the main application window and rendering canvas using the Swing library, which serves as the container for all game scenes.

How to demo: Run the `Main.java` file. A window should successfully open. This window should display the menu scene, showing a "Continue" button and a "New Game" button.

Name: Rendering pingpong rackets and ball

Description: Renders the player's and opponent's paddle sprites onto the `ArenaScene` canvas in their default starting positions.

How to demo: Run `Main.java`. Click the "Continue" button. From the `LevelSelectScene`, click the first available stage icon. The `ArenaScene` should load, and two paddle sprites should be visible: one larger (player) at the bottom and one smaller (opponent) at the top. Furthermore, a ball should render and be in position to be served.

Name: Player racket movement

Description: Allows the player to control their paddle entity horizontally using keyboard input.

How to demo: Start a match, press the left arrow key: the player's paddle should move left. Press the right arrow key: the player's paddle should move right. The paddle should be constrained horizontally and not move off of the screen.

Name: Ball movement & collision

Description: Implements the ball entity's physics, including its 3D-perspective (feel) movement and collision detection with the player's paddle.

How to demo: Start a match. The ball will be served. Use the arrow keys to move the `PlayerPaddle` into the ball's path. When the ball sprite overlaps with the paddle sprite, it should reverse its direction (get reflected) and travel back toward the opponent's side of the table.

Name: Basic opponent

Description: Implements the basic opponent logic for the opponent paddle, allowing it to track the ball's position and attempt to return it.

How to demo: Start a match. Hit the ball back to the opponent's side. Observe the opponent paddel. It should move left and right, automatically trying to position itself under the ball to return the shot.

Name: Scoring system

Description: Tracks the score for the player and opponent within the arena. It updates the score when one side fails to return the ball.

How to demo: Start a match. Win Point: Play a rally and intentionally let the opponent miss the ball. The player's score display should increase by 1. Lose Point: Start a new rally and intentionally move your paddle away from the ball, letting it pass you. The opponent's score display should increase by 1.

Name: Stage progression

Description: Manages unlocking stages and saving progress, tracking which levels are beaten.

How to demo: Win a match on the first stage. You should be returned to the map level select screen. The second stage should now be unlocked and you should be able to play it. Close the entire application.

Name: Player Smash Hit

Description: Adds an advanced player mechanic. If the player presses the spacebar at the correct time (just as the ball is about to hit their paddle), the ball will be returned at a much higher speed.

How to demo: Start a match. Normal Hit: Hit the ball back to the opponent using only the arrow keys and observe its speed. Smash Hit: On the next return, press the spacebar just as the ball

makes contact with your paddle. The ball should accelerate and travel towards the opponent significantly faster than the normal hit.

Name: Sound effects

Description: Plays audio cues from the `assets/audio` folder for game events like hitting the ball.

How to demo: Start a match. Hit: Hit the ball with your paddle; a hit sound effect should play, when the ball hits the table a sound effect should also play. Furthermore, a winning or losing sound effect should play at the end of each stage.

Name: Camera shake

Description: A visual effect that slightly and briefly shakes the game camera when the player hits the ball.

How to demo: Start a match. Move your paddle to hit the ball. Exactly when the paddle and ball entities collide, the entire screen (background, paddles, ball) should visibly shake for a moment to add impact.

Name: Back Button

Description: Adds a clickable "Back" button to the arena that allows the player to quit the match and return to the level select map.

How to demo: Start a match. A "Back" button icon should be visible on the screen (in the top right corner). Click this button. The game should exit the arena and transition back to the level select map.

Name: Win/Lose Match Screens

Description: When a player or opponent reaches the winning score (5 points), the game transitions from the ArenaScene to a "You Win" or "You Lose" screen, using a transition.

How to demo: Win: Start a match and score 5 points against the opponent. As soon as the 5th point is scored, the game should stop and display a "You Win" message/animation. Lose: Start a match and let the opponent score 5 points. The game should stop and display a "You Lose" message/animation.

Name: Background music per stage

Description: Loads and plays a unique audio track from `assets/audio` for each different stage (arena).

How to demo: Stage 1: Start a match on the first stage. Listen to the background music. Stage 2: Win the match, return to the level select map, and start a match on the second (now unlocked) stage. A different background music track should begin playing.