

# Sprint Timeline Document

## *Santa's Workshop Puzzle – Project 3*

---

### Sprint-Based Development Timeline

This project was developed iteratively following a sprint-based workflow. Each sprint focused on a specific set of features, allowing incremental progress, testing, and refinement.

---

### Sprint 1: Core Puzzle Logic (Foundation Sprint)

**Duration:** Day 1

#### Goals:

- Implement the core sliding puzzle mechanics
- Support dynamic grid sizes (3×3, 4×4, etc.)
- Detect valid tile movements
- Detect puzzle completion (win condition)

#### Tasks Completed:

- Designed grid rendering logic using JavaScript
- Implemented tile adjacency and movement rules
- Created empty tile handling
- Implemented win detection logic
- Ensured puzzle initializes in a solved state

#### Outcome:

A fully functional sliding puzzle that works correctly for multiple sizes.

---

## Sprint 2: Game State Tracking (Gameplay Sprint)

**Duration:** Day 2

### Goals:

- Track number of moves
- Track elapsed time
- Reset state correctly between games

### Tasks Completed:

- Implemented move counter tied to tile movement
- Implemented game timer (start on first move, stop on win)
- Reset timer and moves on shuffle or size change
- Added visual status indicators for time and moves

### Challenges:

- Timer starting too early or restarting incorrectly

### Resolution:

- Timer starts strictly on first valid move
- Timer resets only on new game initialization

---

## Sprint 3: Backend Integration (Persistence Sprint)

**Duration:** Day 3

### Goals:

- Persist gameplay data
- Track sessions on backend
- Enable leaderboard storage

### Tasks Completed:

- Created backend API endpoints using PHP
- Designed MySQL tables for users, sessions, and moves
- Implemented session lifecycle (start, record moves, end)
- Stored puzzle size, duration, and move count per session

### **Outcome:**

Game data persisted reliably across sessions.

---

## **Sprint 4: Authentication & Security (User Sprint)**

**Duration:** Day 4

### **Goals:**

- Enable user login and registration
- Restrict leaderboard updates to authenticated users

### **Tasks Completed:**

- Implemented user registration and login
- Added session-based authentication
- Displayed user identity in UI
- Allowed anonymous play but restricted leaderboard persistence

### **Challenges:**

- Session validation issues during backend calls

### **Resolution:**

- Centralized authentication checks in middleware
  - Ensured consistent session handling
-

## **Sprint 5: Leaderboard & Analytics (Graduate Feature Sprint)**

**Duration:** Day 5

### **Goals:**

- Display leaderboard sorted by performance
- Separate leaderboard results by puzzle size

### **Tasks Completed:**

- Implemented size-based leaderboard queries
- Sorted leaderboard by time, then moves
- Synced leaderboard size selector with puzzle size
- Fixed first-completion persistence bug

### **Outcome:**

Accurate, size-specific leaderboards updated in real time.

---

## **Sprint 6: UI/UX & Theming (Polish Sprint)**

**Duration:** Day 6

### **Goals:**

- Improve usability and visual clarity
- Apply a consistent theme
- Enhance user feedback

### **Tasks Completed:**

- Redesigned UI layout for clarity
- Improved tile visibility and interaction cues
- Added win overlay and feedback messages
- Applied a festive Christmas/Santa theme
- Ensured accessibility and readability

## **Outcome:**

A polished, visually engaging application suitable for presentation.

---

## **Final Review & Testing Sprint**

**Duration:** Final Day

### **Tasks:**

- End-to-end testing of all features
  - Verified leaderboard accuracy
  - Tested multiple puzzle sizes
  - Verified authentication flows
  - Cleaned up unused code and debug logs
- 

## **Conclusion**

Using a sprint-based approach allowed structured progress and continuous improvement throughout the development cycle. Each sprint focused on a clear goal, ensuring both functional correctness and graduate-level complexity were achieved.