PRD

Dynamic Vision Test Web Game

---

1. Product Overview

1.1 Product Definition

The Dynamic Vision Test Web Game is an educational web application designed to test and improve users' dynamic vision (the ability to recognize and track moving objects). It is provided in a game format where users recognize and remember Korean characters falling simultaneously from various positions on the screen to complete words.

1.2 Target Audience

- **Primary Target**: Students, general adults (age 10 and above)
- **Usage Purpose**:
- Dynamic vision improvement training
- Cognitive ability testing
- Enjoyable brain game experience
- **Technical Level**: Basic web browser usage skills

1.3 Core Purpose

The core purpose is to test and improve users' ability to recognize and remember multiple rapidly moving characters simultaneously to complete words. This comprehensively enhances concentration, memory, and dynamic vision.

---

2. Project Goals

2.1 User Goals

- Users can test and improve their dynamic vision in an enjoyable way
- Users can adjust difficulty levels to enjoy the game according to their skill level
- Users can check and improve their skills through scoring

2.2 Business Goals

- Provide basic structure and user experience for educational web applications

- Provide frontend development practice using HTML, CSS, and JavaScript
- Experience the complete process from product planning to implementation

2.3 Technical Goals

- Implement a complete game using only pure web technologies (HTML, CSS, JavaScript)
- Apply responsive design to work on various screen sizes
- Provide a clean and intuitive user interface

---

3. Feature List

3.1 Must-have Features (Required)

3.1.1 Game Start and Word Generation

- **Priority**: Must-have
- **Description**: Randomly select Korean words of 2-3 characters for use in the game
- **Detailed Requirements**:
- Build Korean word database (minimum 20 words each for 2-character and 3-character words)
- Implement random selection algorithm
- Process each character of the selected word individually

3.1.2 Difficulty Setting System

- **Priority**: Must-have
- **Description**: Classify difficulty into High, Medium, and Low based on character stroke count
- **Detailed Requirements**:
- High difficulty: Complex characters (10+ strokes)
- Medium difficulty: Medium complexity characters (5-9 strokes)
- Low difficulty: Simple characters (4 or fewer strokes)
- Provide UI for users to select difficulty level

3.1.3 Multi-directional Character Falling

- **Priority**: Must-have
- **Description**: Animation where each character falls from top, left, and right of the monitor
- **Detailed Requirements**:
- Each character starts from top, top-left, or top-right of the screen
- Natural falling animation with gravity effects applied

- Characters start from different positions and fall simultaneously
- Characters disappear when they reach the bottom of the screen

3.1.4 Falling Speed Control

- **Priority**: Must-have
- **Description**: Feature allowing users to adjust the speed at which characters fall
- **Detailed Requirements**:
- Provide speed control slider or buttons
- Provide 3-5 levels from minimum speed (slow) to maximum speed (fast)
- Allow real-time speed changes

3.1.5 Input System

- **Priority**: Must-have
- **Description**: Feature allowing users to input words after all characters have fallen
- **Detailed Requirements**:
- Display input field
- Display 1.5-second time limit timer
- Submit via submit button or Enter key
- Auto-submit on timeout

3.1.6 Scoring System

- **Priority**: Must-have
- **Description**: Score increases on correct answer, game ends on incorrect answer
- **Detailed Requirements**:
- Score increases on correct answer (different points based on difficulty)
- Return to initial screen on incorrect answer
- Real-time display of current score

3.1.7 Game Restart Feature

- **Priority**: Must-have
- **Description**: Feature to restart the game after it ends
- **Detailed Requirements**:
- Provide "Restart" button
- Automatically return to initial screen on incorrect answer

3.2 Should-have Features (Recommended)

3.2.1 Game Screen Transitions

- **Priority**: Should-have

- **Description**: Smooth transitions between main screen, game screen, and settings screen
- **Detailed Requirements**:
- Navigation menu between pages
- Transition animation effects

3.2.2 Visual Feedback

- **Priority**: Should-have
- **Description**: Provide visual feedback for user actions
- **Detailed Requirements**:
- Success message and effects on correct answer
- Failure message and effects on incorrect answer
- Visual timer display (progress bar or numbers)

3.2.3 Responsive Design

- **Priority**: Should-have
- **Description**: Layout that adapts to various screen sizes
- **Detailed Requirements**:
- Support desktop, tablet, and mobile screens
- Responsive CSS using media queries

3.3 Nice-to-have Features (Optional)

3.3.1 Sound Effects

- **Priority**: Nice-to-have
- **Description**: Sound effects for game actions
- **Detailed Requirements**:
- Correct/incorrect answer sounds
- Background music (optional)

3.3.2 Statistics Feature

- **Priority**: Nice-to-have
- **Description**: Record game play statistics
- **Detailed Requirements**:
- High score record
- Average accuracy rate
- Play count

---

4. Technical and Design Requirements

4.1 Technology Stack

4.1.1 Frontend

- **HTML5**: Web page structure and semantic markup
- **CSS3**: Styling, animations, responsive design
- **JavaScript (ES6+)**: Game logic, animation control, user interaction

4.1.2 Additional Libraries/Frameworks

- **Optional**: Implement with pure JavaScript, but consider the following if needed
- CSS Framework: Bootstrap or Tailwind CSS (optional)
- Animation: CSS animations or Web Animations API
- Fonts: Google Fonts (Korean fonts)

4.1.3 Development Tools

- **Version Control**: Git, GitHub
- **Code Editor**: Visual Studio Code or similar IDE
- **Browser Compatibility**: Chrome, Firefox, Safari, Edge (latest versions)

4.2 UI/UX Guidelines

4.2.1 Design Principles

- **Simplicity**: Intuitive and easy-to-understand interface
- **Clarity**: Clear purpose for all buttons and features
- **Consistency**: Consistent design language across all pages

4.2.2 Color Palette

- **Primary Colors**:
- Background: Light gray or white (#F5F5F5 or `#FFFFFF`)
- Primary Action: Blue or green (#4A90E2 or `#52C41A`)
- Warning/Error: Red (#FF4D4F)
- Text: Dark gray or black (#333333 or #000000)
- **Contrast**: Comply with WCAG 2.1 AA standards (minimum 4.5:1)

4.2.3 Typography

- **Korean Font**: Noto Sans KR or Nanum Gothic
- **English Font**: System font stack

- **Font Sizes**:
- Headings: 24px or larger
- Body: 16px
- Small text: 14px

4.2.4 Layout

- **Page Structure**:
- Header: Game title and navigation
- Main content area: Game screen
- Footer: Simple information
- **Grid System**: Use Flexbox or CSS Grid

4.2.5 Interaction Design

- **Buttons**:
- Minimum size: 44x44px (touch-friendly)
- Hover effects: Color change or shadow
- Click feedback: Visual change
- **Input Fields**:
- Clear borders
- Focus state indication
- Placeholder text

4.3 Performance Requirements

- **Loading Time**: Initial page load within 2 seconds
- **Animation**: Maintain 60fps (smooth animation)
- **Responsiveness**: Response time to user actions within 100ms

4.4 Accessibility Requirements

- **Keyboard Navigation**: All features accessible via keyboard
- **Screen Reader**: Use semantic HTML tags
- **Color Dependency**: Do not convey information using color alone

---

5. Milestones

Milestone 1: Project Planning and Design (Completed)

- **Duration**: 1 day
- **Tasks**:

- Write PRD document
- Determine technology stack
- Design project structure
- **Completion Criteria**: PRD document approved and project structure finalized

Milestone 2: Basic Website Structure Implementation

- **Duration**: 1 day
- **Tasks**:
- Write basic HTML structure (3 pages)
- Basic CSS styling
- Implement navigation between pages
- **Completion Criteria**: All pages display with basic layout

Milestone 3: Core Game Feature Implementation

- **Duration**: 2 days
- **Tasks**:
- Build Korean word database
- Random word selection feature
- Difficulty setting system
- Character falling animation
- Speed control feature
- **Completion Criteria**: Core game mechanics work

Milestone 4: Input and Scoring System Implementation

- **Duration**: 1 day
- **Tasks**:
- Implement input system
- 1.5-second timer feature
- Correct/incorrect answer determination logic
- Scoring system implementation
- **Completion Criteria**: Game works in complete cycle

Milestone 5: UI/UX Improvement and Final Testing

- **Duration**: 1 day
- **Tasks**:
- Design improvements
- Apply responsive design
- Browser compatibility testing

- Bug fixes
- **Completion Criteria**: All features work normally and design is complete

Milestone 6: Documentation and Deployment Preparation

- **Duration**: 1 day
- **Tasks**:
- Write README.md
- Organize code comments
- Set up and upload to GitHub repository
- **Completion Criteria**: Project uploaded to GitHub and documentation completed

---

6. Constraints and Assumptions

6.1 Constraints

- **Browser**: Support only latest browsers (no IE support)
- **Internet Connection**: Must work offline (minimize CDN usage)
- **Data Storage**: Use only local storage (no server)

6.2 Assumptions

- Users can read and input Korean characters
- Users know basic web browser usage
- Screen width is at least 320px

---

7. Success Metrics

7.1 Functional Success Metrics

- All Must-have features work normally
- Game works in complete cycle (Start → Play → Result → Restart)
- Works stably at various difficulty levels and speeds

7.2 User Experience Success Metrics

- Intuitive interface usable without additional explanation
- Smooth animation (60fps)
- Fast response time (within 100ms)

7.3 Technical Success Metrics

- Code readability and maintainability
- Consistency between PRD and implementation
- Clean code structure

---

8. Future Improvements

8.1 Short-term Improvements (Next Version)

- Add sound effects
- Add statistics feature
- Expand Korean word database

8.2 Long-term Improvements

- Multiplayer mode
- Leaderboard feature
- Mobile app version development