**✅ UNO Game Project: Phases 1–4 Documentation**

**📌 PROJECT GOAL**

Build a **complete browser-based UNO game** using **HTML, CSS, and JavaScript**:

* Fully playable in the browser
* Interactive UI
* Handles main UNO rules
* Good enough to be a portfolio project

**📌 DEVELOPMENT PRINCIPLES**

* Code everything from scratch (no frameworks)
* Semantic HTML
* Single CSS file
* Single JavaScript file (planned for now)
* All files in a single folder (e.g., “UNO”)
* No server or Python backend (pure JS in-browser)

**✅ PHASE 1: Planning & UI Design**

**✔️ 1.1 Define Core Rules & Scope**

* Colors: red, blue, green, yellow, black (for +4 and wild)
* Numbers: 0–9
* Specials: skip, reverse, +2, wild, +4
* For now: **no stacking or 7-0 rule** (basic rules only)
* Players: human vs. AI

**✔️ 1.2 Sketch/Wireframe Layout**

* Single-screen game
* UI sketch included:
  + Player hand (clickable)
  + AI hand (hidden backs)
  + Draw pile (clickable)
  + Discard pile (shows top card)
  + Play Card button
  + UNO button
  + Turn indicator
  + Message/log area

**✔️ 1.3 Decide Main UI Components**

* Game board container
* AI hand area
* Player hand area
* Draw pile
* Discard pile
* Turn indicator
* UNO button
* Play Card button
* Message/log area
* Card component for cards in hand/piles

**✔️ 1.4 Plan App Architecture**

* 1 HTML file for game screen (plus optional start/win/lose screens later)
* Single CSS file
* Single JS file
* No frameworks
* Semantic structure, IDs/classes as needed

✅ **Phase 1 complete**

**✅ PHASE 2: HTML Layout**

**✔️ 2.1 Set Up Project Folder Structure**

* Single folder (“UNO”)
* Contains: HTML, CSS, JS files

**✔️ 2.2 Create HTML Page with Semantic Layout**

* Used <main>, <section>, <aside>
* Clear IDs for game-board, ai-hand, player-hand, center-area, draw-pile, discard-pile, controls
* Buttons with IDs: draw-btn, play-btn, uno-btn
* Turn indicator and message area included

**✔️ 2.3 Add Placeholder Elements**

* Added visible text labels for all elements:
  + “AI Hand”, “PLAYER Hand”
  + “DRAW”, “DISCARD”
  + Buttons labeled: “Draw Card”, “Play Card”, “UNO!!!”
  + Turn indicator and message area labeled

✅ **Phase 2 complete**

**✅ PHASE 3: CSS Styling**

**✔️ 3.1 Define Color Scheme and Card Design**

* Card colors:
  + Red: #E74C3C
  + Blue: #3498DB
  + Green: #27AE60
  + Yellow: #F1C40F
  + Wild: #2C3E50
* Text color: white or black for contrast
* Card design:
  + Rounded corners
  + ~80x120px (prototype was ~200x300px for clarity)
  + Large number in center
* Board background: #222
* Buttons: bright accent colors

**✔️ 3.2 Style Cards**

* Created reusable .card class
* Included all color classes
* Layout with top/bottom small symbols and large center number
* Prototype tested with standalone HTML and CSS

**✔️ 3.3 Style Player Hands**

* Styled #player-hand and #ai-hand
* Used flexbox layout
* Horizontal display of cards with gap and padding
* AI hand shows card backs (brown with .ai class)

**✔️ 3.4 Style Draw/Discard Piles**

* Styled #draw-pile and #discard-pile
* Flex layout in center area
* Consistent sizing and borders matching card design
* Draw pile with card back color

**✔️ 3.5 Style Turn Indicators and Controls**

* #controls area with background color, padding
* Flex column layout with two rows:
  + Top row: buttons + turn indicator
  + Bottom row: message area
* Buttons styled consistently
* Turn indicator: circular, color-coded
* Message area clear with border and padding

**⚠️ 3.6 Make Layout Responsive/Scalable**

* *Skipped for now*
* Plan to revisit later for responsive breakpoints and scaling

✅ **Phase 3 complete (except 3.6 skipped for now)**

**✅ PHASE 4: JavaScript – Core Game Model**

**✔️ 4.1 Define Card Object / Structure**

* Card properties:
  + Color: red, blue, green, yellow, black
  + Value: 0–9, skip, reverse, +2, wild, +4
  + Type: number, action, wild

**✔️ 4.2 Define Deck Creation & Shuffling**

* Deck will be an array of card objects
* Loop over colors to add number/action cards
* Add wild cards separately
* Shuffle by randomizing array order in JS

**✔️ 4.3 Define Player Hands**

* Player vs AI (2 players)
* Each hand = array of card objects
* Plan allows future scaling to more players (turn array)
* Drawing = push to hand array
* Playing = remove from hand array
* Turn tracked with variable (turn flag)

**✔️ 4.4 Implement Draw/Discard Piles**

* Draw pile = array (stack)
  + Remove top card when drawn
  + Add to player/AI hand
* Discard pile = array
  + Add played card to top
  + UI shows last card
* If draw pile runs out:
  + Alert user (shuffling)
  + Take all discard cards except top
  + Shuffle and refill draw pile

**✔️ 4.5 Define Turn Order Logic**

* Plan for 2 players now but scalable to more
* Turn order stored as array of players
* Circular indexing for next player
* Special cards:
  + Skip: skip next player's turn
  + Reverse: reverse order (in 2-player acts as skip)
  + +2, +4: next player draws cards and loses turn
* Turn indicator in UI shows whose turn

**✔️ 4.6 Implement Rule Checking**

* Must access last played (top discard) card
* Player can play:
  + Same color
  + Same value/symbol
  + Any wild card
* Wild handling:
  + When played, prompt user to pick new color
  + Store chosen color in wild card
  + Future plays must match this chosen color
* Plan for UI color picker

✅ **Phase 4 complete**