Process & Decision Documentation

Project/Assignment Decisions

Side Quests and A4 (Individual Work)

Keep this section brief, typically 2 to 4 sentences.

Focus on:

I built the game using a **game-state structure with multiple files**, organizing each screen into its own file (start, game, win, lose) instead of putting everything into one large script. This made the code easier to manage, reduced confusion between screens, and allowed the project to scale as I added more story beats and player stats.

Role-Based Process Evidence

Entry Header

Name: Suyao Liu

Role(s): Designer, Developer

Primary responsibility for this work: Designing the interactive story, organizing the project using a multi-file game-state structure, and implementing player stats and branching logic.

Goal of Work Session

- Build an interactive story that unfolds through multiple game states
- Organize code using separate files for each screen
- Track Trust, Health, and Energy across scenes and use them to determine endings

Tools, Resources, or Inputs Used

- Lecture slides on Organizing Code with Multiple Files and Game States
- GenAI (ChatGPT 5.2) for support when extending and balancing the game

GenAI Documentation

If GenAI was used (keep each response as brief as possible):

Date Used: Feb 1st 2026

Tool Disclosure: Codex (Chatgpt Agent in VScode)

Purpose of Use: GenAI was used to support building and extending an interactive story system, including organizing code into multiple game states, expanding story beats, and refining player stat logic (Trust, Health, Energy).

Summary of Interaction: I used GenAI by giving specific instructions related to the assignment requirements, such as building a multi-file game-state structure, adding branching story beats, tracking player stats across scenes, and unlocking endings based on those stats. The tool provided suggestions for structuring files, created and expanding story content, and balancing stat changes.

Human Decision Point(s): I decided how the game was structured, what story content to keep, how stats affected outcomes, and when to revise choices that felt too predictable. I adjusted stat values, added or removed choices, and refined logic based on playtesting.

Integrity & Verification Note: I verified all changes by manually running the game, checking that stats updated correctly across states, and confirming that win and lose conditions triggered as intended.

Scope of GenAI Use: GenAI supported implementation and iteration but did not independently design the project. All final decisions, structure, and code integration were done by me.

Limitations or Misfires: Some suggestions initially made success too obvious, which required further balancing and the addition of randomized stat ranges.

Summary of Process (Human + Tool)

I built the project iteratively, starting with a basic multi-file interactive story and gradually expanding it through multiple versions. I tested the game after each change, revised story beats, adjusted stat logic, and reworked choices when outcomes felt too predictable. When uncertainty came up around structure or balance, I used GenAI for support while making final decisions myself through testing and revision.

Decision Points & Trade-offs

Options considered: Keeping a simple Trust-only system versus adding additional player stats

What changed: Health and Energy were added alongside Trust, and later randomized stat ranges were introduced

Why that choice was made: This reduced obvious “correct” choices and made decision-making more complex and engaging while still fitting the story context

Verification & Judgement

I evaluated my decisions primarily through repeated playtesting, checking whether choices led to varied outcomes and whether stats behaved as intended across game states. I also compared the final structure against lecture concepts on organizing code with multiple files and game states to ensure the implementation aligned with course material.

Limitations, Dead Ends, or Open Questions

- Early versions of the game made the top choice too consistently successful, requiring additional balancing
- No sound or animation was added due to time constraints
- Stats reset between playthroughs instead of persisting across reloads

Appendix

Build an interactive story that unfolds through multiple game states and files, branching like a small decision tree. • Bonus: Track a player stat (e.g., trust, health, karma) across scenes and unlock endings based on it.

Built a multi-file branching interactive story with a tracked Trust stat that gates endings, using shared logic in common.js and scene files under scenes/.

What changed

- Added story pages and branching: index.html, scene1.html, scene2a.html, scene2b.html, gate.html, ending-good.html, ending-bad.html.
- Implemented Trust stat persistence and choice handling in common.js.
- Styled the experience in styles.css with a distinct visual mood.

Files

- index.html
- styles.css
- common.js
- scene1.html
- scene2a.html
- scene2b.html
- gate.html
- ending-good.html
- ending-bad.html

If you want, I can:

1. Add more branches/endings or a neutral ending.
2. Add sound/ambient effects and transitions.

The Solution: Game States Organize code by screen/state - one file per screen. Load them all in index.html : • index.html → Loads all your scripts in order. p5.js runs what is inside. • main.js → Master controller with setup() , draw() , input routing. • start.js → Start screen with menu buttons. Defines drawStart(). • game.js → Game screen with gameplay. Defines drawGame(). • win.js / lose.js → Outcome screens. Each defines drawWin() , drawLose() .

Set up the game-state structure you asked for: index.html now loads p5.js
plus start.js, game.js, win.js, lose.js, and main.js, with a master controller that routes setup()/draw() and state transitions.
Details

- main.js owns setup(), draw(), state switching, and UI helpers.
- start.js, game.js, win.js, lose.js each define their draw...() and enter...() flows.
- styles.css updated to style the p5 canvas + UI buttons.

Files created/updated

- index.html
- main.js
- start.js
- game.js
- win.js
- lose.js
- styles.css

Note: the earlier multi-page HTML story files in scenes/ and common.js are still present but not used by this game-state setup. If you want, I can clean those out or merge the story content into the p5 screens.
Next steps if you want:

1. Add input routing (mouse/key) per state and highlight current choices.
2. Expand the decision tree to 3–4 beats with more trust thresholds.

Track a player stat (e.g., trust, health, karma) across scenes and unlock endings based on it. and give me an read me file

Added Trust stat visibility on start/win/lose screens and wrote a README describing the stat gating and file structure.
Files updated/added

- start.js
- win.js
- lose.js
- README.md

If you want the Trust stat persisted across reloads (localStorage) or a more complex decision tree, tell me the rules and I'll wire it up.
can you make the game longer

How much longer do you want it, and in what way?

- More story beats in game.js (e.g., 4–6 total choices)?
- New mid-states (e.g., game2.js, game3.js)?
- More endings with extra thresholds?

Tell me the number of beats and any themes/choices you want, and I'll extend it.
I want more story beats and for the read me file use this format Project Title Give your project a short, clear title (e.g., "Side Quest Week 1: Interactive Shape Switcher"). Group Members List the full name, WatID, and student number for each group member. If the project is individual, include only your own information. Description Briefly describe what your project does. Explain what the user will see, how they can interact, and what inspired the design. Example: This project lets users switch between shapes using keyboard inputs. It explores how visual feedback can communicate accessibility states. Interaction Instructions Describe how to use or play your project. How does the user start? What controls or keys do they use? Any important visual or audio feedback to watch for? Example: Press S to switch shapes and R to reset the canvas. Assets List any images, sounds, or other files used in your assets/ folder. Include in-text citations for all assets that are not your own, and connect them to your reference list. Example:

assets/sound/beep.mp3 (created in GarageBand) assets/image/background.png (original artwork) References Connect your in-text citations to the reference list along with additional sources that informed your design but were not directly cited. Provide all citations using the ACM reference format.

Extended the story to 5 beats in game.js and rewrote README.md in the exact format you requested. Please send your **Full Name, WatID, and Student Number** so I can replace the placeholders in README.md.
maybe put the content on the big screen instead of having it showing the same "the frog shift with every choice" and make a bit more complex game with health and energy incorperated and displayed on the big screen

Editedmain.js

```
let trust = 50;
let health = 75;
let energy = 75;
let uiRoot;
const TRUST_MAX = 100;
const STAT_MIN = 0;
const STAT_MAX = 100;


const clampTrust = (value) => Math.min(TRUST_MAX, Math.max(TRUST_MIN, value));
const clampStat = (value) => Math.min(STAT_MAX, Math.max(STAT_MIN, value));
```

Editedstart.js

```
  title.addClass("ui-title");
  addCaption(`Trust: ${trust}`);
  addCaption(`Trust: ${trust}  |  Health: ${health}  |  Energy: ${energy}`);
  addCaption("A short tale about trust. Choose how to approach the beacon.");
      trust = 70;
      health = 75;
      energy = 75;
      setState("game");
```

Editedgame.js

```
    options: [
      { label: "Offer your tools", delta: 10 },
      { label: "Hold the tools back", delta: -8 },
      { label: "Offer your tools", trust: 10, health: 0, energy: -5 },
```

```
    { label: "Hold the tools back", trust: -8, health: 0, energy: -2 },
  ],
  options: [
    { label: "Ask about your crew", delta: 8 },
    { label: "Take the beacon and run", delta: -10 },
    { label: "Ask about your crew", trust: 8, health: 0, energy: -4 },
    { label: "Take the beacon and run", trust: -10, health: -6, energy: -8 },
  ],
  options: [
    { label: "Follow the echo", delta: 6 },
    { label: "Stay and secure the beacon", delta: -4 },
    { label: "Follow the echo", trust: 6, health: -4, energy: -6 },
    { label: "Stay and secure the beacon", trust: -4, health: 2, energy: 3 },
  ],
  options: [
    { label: "Read the logbook together", delta: 7 },
    { label: "Pocket the key quietly", delta: -6 },
    { label: "Read the logbook together", trust: 7, health: 0, energy: -3 },
    { label: "Pocket the key quietly", trust: -6, health: 0, energy: 2 },
  ],
  options: [
    { label: "Share the map coordinates", delta: 9 },
    { label: "Erase the trail behind you", delta: -8 },
    { label: "Share the map coordinates", trust: 9, health: 0, energy: -4 },
    { label: "Erase the trail behind you", trust: -8, health: -3, energy: -2 },
  ],
},
{
  text: "The path narrows. The fog drains your strength.",
  options: [
    { label: "Push forward together", trust: 6, health: -8, energy: -10 },
    { label: "Rest and make a plan", trust: 2, health: 6, energy: 10 },
  ],
```

```
},
{
  text: "A final broadcast requests a response code.",
  options: [
    { label: "Share the code", trust: 7, health: 0, energy: -5 },
    { label: "Jam the signal", trust: -9, health: -4, energy: -6 },
  ],
trust = clampTrust(trust);
health = clampStat(health);
energy = clampStat(energy);
renderGameUI();
clearUI();
addCaption(`Trust: ${trust}`);
addCaption(`Trust: ${trust}  |  Health: ${health}  |  Energy: ${energy}`);

  addButton(option.label, () => {
    trust = clampTrust(trust + option.delta);
    trust = clampTrust(trust + option.trust);
    health = clampStat(health + option.health);
    energy = clampStat(energy + option.energy);
    storyIndex += 1;
    if (storyIndex >= storyBeats.length) {
      setState(trust >= 60 ? "win" : "lose");
      const survived = health > 0 && energy > 0;
      const earnedTrust = trust >= 60;
      setState(survived && earnedTrust ? "win" : "lose");
    } else {
    }
    , option.delta < 0
    , option.trust < 0
  ;
noStroke();
fill(11, 95, 122, 25);
```

```
  fill(11, 95, 122, 18);
  rect(30, 30, width - 60, height - 60, 24);
  fill(31, 26, 23);
  textSize(20);
  textAlign(LEFT, TOP);
  text("The fog shifts with every choice.", 50, 50, width - 100, height - 100);
  textSize(22);
  text("The Signal in the Glade", 50, 50);

  textSize(16);
  text(`Beat ${storyIndex + 1} of ${storyBeats.length}`, 50, 88);

  const beat = storyBeats[storyIndex];
  const storyText = beat ? beat.text : "The story concludes.";
  textSize(20);
  text(storyText, 50, 130, width - 100, height - 220);

  textSize(16);
  text(`Trust: ${trust}`, 50, height - 90);
  text(`Health: ${health}`, 200, height - 90);
  text(`Energy: ${energy}`, 360, height - 90);
 ;
```

Editedwin.js

```
  title.addClass("ui-title");
  addCaption(`Trust: ${trust}`);
  addCaption(`Trust: ${trust}  |  Health: ${health}  |  Energy: ${energy}`);
  addCaption("Your trust unlocked the hidden camp. The crew steps out.");
```

Editedose.js

```
  title.addClass("ui-title");
  addCaption(`Trust: ${trust}`);
  addCaption(`Trust: ${trust}  |  Health: ${health}  |  Energy: ${energy}`);
  addCaption("The beacon fades. The forest keeps its secrets.");
```

Editedwin.js

```
trust = 50;
health = 75;
energy = 75;
setState("start");
```

Edited lose.js

```
trust = 50;
health = 75;
energy = 75;
setState("start");
trust = 30;
health = 55;
energy = 55;
setState("game");
```

Updated the game so the big screen (canvas) shows the current story beat and all stats, and expanded the story with more beats plus Health/Energy tracking.

What changed

- Added health and energy stats alongside Trust, with clamping and win/lose based on trust + survival.
- Expanded to 7 story beats with per-choice impacts on Trust/Health/Energy.
- Canvas now displays the current beat text and live stats instead of the repeated “fog shifts” line.
- Start/Win/Lose screens show all three stats and reset them appropriately.

Files updated

- main.js
- game.js
- start.js
- win.js
- lose.js

If you want, I can also:

1. Add mid-run stat thresholds that branch to different beats.
2. Add a “game over” when Health or Energy hits 0 before the final beat.

5 files changed+66-20

UndoReview

game.js

lose.js

main.js

start.js

win.js

shift the choices rn it's clear keep clicking the top choice would lead to succes

Editedgame.js

```
    options: [
      { label: "Offer your tools", trust: 10, health: 0, energy: -5 },
      { label: "Hold the tools back", trust: -8, health: 0, energy: -2 },
      { label: "Offer your tools", trust: 6, health: 2, energy: -6 },
      { label: "Hold the tools back", trust: -2, health: 0, energy: 4 },
    ],
    options: [
      { label: "Ask about your crew", trust: 8, health: 0, energy: -4 },
      { label: "Take the beacon and run", trust: -10, health: -6, energy: -8 },
      { label: "Ask about your crew", trust: 4, health: -2, energy: -3 },
      { label: "Take the beacon and run", trust: -4, health: -3, energy: -1 },
    ],
    options: [
      { label: "Follow the echo", trust: 6, health: -4, energy: -6 },
      { label: "Stay and secure the beacon", trust: -4, health: 2, energy: 3 },
      { label: "Follow the echo", trust: 5, health: -6, energy: -4 },
      { label: "Stay and secure the beacon", trust: -1, health: 4, energy: 6 },
    ],
    options: [
      { label: "Read the logbook together", trust: 7, health: 0, energy: -3 },
      { label: "Pocket the key quietly", trust: -6, health: 0, energy: 2 },
      { label: "Read the logbook together", trust: 6, health: 0, energy: -4 },
      { label: "Pocket the key quietly", trust: -3, health: 2, energy: 4 },
    ],
    options: [
      { label: "Share the map coordinates", trust: 9, health: 0, energy: -4 },
      { label: "Erase the trail behind you", trust: -8, health: -3, energy: -2 },
      { label: "Share the map coordinates", trust: 7, health: -2, energy: -5 },
      { label: "Erase the trail behind you", trust: -2, health: 3, energy: 5 },
    ],
    options: [
```

```
    { label: "Push forward together", trust: 6, health: -8, energy: -10 },
    { label: "Rest and make a plan", trust: 2, health: 6, energy: 10 },
    { label: "Push forward together", trust: 5, health: -9, energy: -6 },
    { label: "Rest and make a plan", trust: 1, health: 6, energy: 8 },
  ],
  options: [
    { label: "Share the code", trust: 7, health: 0, energy: -5 },
    { label: "Jam the signal", trust: -9, health: -4, energy: -6 },
    { label: "Share the code", trust: 6, health: -2, energy: -4 },
    { label: "Jam the signal", trust: -4, health: -1, energy: -2 },
  ],
```

Balanced the choices so the top option isn't always the obvious "best" path. In game.js I mixed trade-offs between Trust, Health, and Energy across beats so each option has pros/cons.
If you want it even less predictable, I can add occasional "risky good" / "safe bad" beats or add a small random range to stat changes.

1 file changed

Undo Review

game.js

yes make it more unpredictable but at the same time make sense to the content of the game

Edited game.js

```
const roll = (min, max) =>
  Math.floor(Math.random() * (max - min + 1)) + min;

const applyDelta = (value, delta) => {
  if (Array.isArray(delta)) return value + roll(delta[0], delta[1]);
  return value + delta;
};

const storyBeats = [
  options: [
    { label: "Offer your tools", trust: 6, health: 2, energy: -6 },
    { label: "Hold the tools back", trust: -2, health: 0, energy: 4 },
    { label: "Offer your tools", trust: [3, 8], health: [1, 4], energy: [-8, -3] },
```

```
  { label: "Hold the tools back", trust: [-4, 1], health: [-1, 2], energy: [2, 6] },
],
options: [
  { label: "Ask about your crew", trust: 4, health: -2, energy: -3 },
  { label: "Take the beacon and run", trust: -4, health: -3, energy: -1 },
  { label: "Ask about your crew", trust: [2, 6], health: [-4, -1], energy: [-5, -2] },
  { label: "Take the beacon and run", trust: [-6, -1], health: [-5, -1], energy: [-2, 1] },
],
options: [
  { label: "Follow the echo", trust: 5, health: -6, energy: -4 },
  { label: "Stay and secure the beacon", trust: -1, health: 4, energy: 6 },
  { label: "Follow the echo", trust: [3, 7], health: [-8, -3], energy: [-6, -2] },
  { label: "Stay and secure the beacon", trust: [-2, 2], health: [2, 6], energy: [4, 8] },
],
options: [
  { label: "Read the logbook together", trust: 6, health: 0, energy: -4 },
  { label: "Pocket the key quietly", trust: -3, health: 2, energy: 4 },
  { label: "Read the logbook together", trust: [4, 8], health: [-1, 2], energy: [-5, -2] },
  { label: "Pocket the key quietly", trust: [-4, -1], health: [1, 4], energy: [2, 6] },
],
options: [
  { label: "Share the map coordinates", trust: 7, health: -2, energy: -5 },
  { label: "Erase the trail behind you", trust: -2, health: 3, energy: 5 },
  { label: "Share the map coordinates", trust: [5, 9], health: [-4, -1], energy: [-6, -3] },
  { label: "Erase the trail behind you", trust: [-3, 1], health: [2, 6], energy: [3, 7] },
],
options: [
  { label: "Push forward together", trust: 5, health: -9, energy: -6 },
  { label: "Rest and make a plan", trust: 1, health: 6, energy: 8 },
  { label: "Push forward together", trust: [4, 7], health: [-10, -5], energy: [-8, -4] },
  { label: "Rest and make a plan", trust: [0, 3], health: [4, 8], energy: [6, 10] },
],
options: [
```

```
    { label: "Share the code", trust: 6, health: -2, energy: -4 },
    { label: "Jam the signal", trust: -4, health: -1, energy: -2 },
    { label: "Share the code", trust: [4, 8], health: [-3, 0], energy: [-6, -2] },
    { label: "Jam the signal", trust: [-5, -2], health: [-2, 1], energy: [-3, 0] },
  ],
  addButton(option.label, () => {
    trust = clampTrust(trust + option.trust);
    health = clampStat(health + option.health);
    energy = clampStat(energy + option.energy);
    trust = clampTrust(applyDelta(trust, option.trust));
    health = clampStat(applyDelta(health, option.health));
    energy = clampStat(applyDelta(energy, option.energy));
    storyIndex += 1;
```

Made the outcomes more unpredictable but still thematically consistent by turning each choice into a stat range. So “trusting” options usually boost Trust but can drain Energy/Health, while cautious options conserve resources but may lower Trust—now with variability inside sensible bounds.

Updated file

- game.js

If you want, I can also show the exact roll results after each choice (e.g., “Trust +5, Health -2”) so players understand why their stats shifted.

1 file changed

UndoReview

game.js

give some beats more than 2 choice

Editedgame.js

```
    { label: "Hold the tools back", trust: [-4, 1], health: [-1, 2], energy: [2, 6] },
    { label: "Ask them to fix it together", trust: [1, 5], health: [0, 3], energy: [-6, -2] },
  ],
    { label: "Pocket the key quietly", trust: [-4, -1], health: [1, 4], energy: [2, 6] },
    { label: "Leave the cache untouched", trust: [-1, 3], health: [0, 2], energy: [0, 4] },
  ],
    { label: "Erase the trail behind you", trust: [-3, 1], health: [2, 6], energy: [3, 7] },
    { label: "Split up and compare routes", trust: [0, 4], health: [-2, 2], energy: [-2, 2] },
  ],
    { label: "Jam the signal", trust: [-5, -2], health: [-2, 1], energy: [-3, 0] },
```

```
    { label: "Send a partial reply", trust: [1, 5], health: [-1, 1], energy: [-2, -1] },
  ],
```

Added a third option to four beats so some scenes now have 3 choices.
Updated file

- game.js

If you want, I can add 3–4 choice beats to every scene or just a couple more.

1 file changed

UndoReview

game.js

update the read me file

Updated README.md to reflect the new Trust/Health/Energy system, randomized stat ranges, and win/lose conditions.
Send your full name, WatID, and student number and I'll fill them in.