Make a project using code and design skills at Barnes and Noble for fun!

Tools

1. Laptop
2. VS Code or Cursor
    a. Cursor Requires Account (Free Trial Available)
    b. Cursor has AI Built In, but can be a little clunky to use
    c. VS Code starts off bare bones and you add features as you realize you want them
3. VS Code/Cursor Add Ons
    a. Live Server
    b. Anything else you want within reason
4. Github Desktop
    a. Need Github Account to sync code to server and share project with each other

Platform/Technology

1. Web Browser
2. HTML/JS/CSS
3. External Libraries are fine as long as you can figure out how to use them

Method

1. Choose Project
2. Define Timeframe 2-4 Hours?
3. Break Down into Small Pieces
4. Assign Tasks
5. Work/Share Progress/Sync Project/Test/Discuss
    a. Discuss
    b. Work
    c. Share Progress
    d. Sync Project
    e. Test
    f. Go back to 5a. And repeat

Project Ideas

1. Simple Games
    a. **Taco Food truck Chaos** - Manage a busy food truck by preparing ingredients, cooking meals, and serving them to customers under time pressure.
    b. **Dog catches frisbee** - a dog is sitting in the middle of a dog park. When the frisbee flies above, jump to catch it. Avoid birds. Cute background music and pixel art

c. **Color Match** - Two circles of color are shown side by side, and the player must quickly decide if they are the same color. The color values will be extremely close to trick the user.

Caveats

- Keep it simple, Get it done, Have some fun

Collaborate on code using **GitHub Desktop**:

Step 1: Create a GitHub Repository

1. One developer should create a repository on GitHub.
2. Go to GitHub and click the "+" icon to create a new repository.
3. Give the repository a name and select "Private" or "Public" (depending on your needs).
4. Add a `README.md` file to initialize the repository.
5. Click "Create Repository".

Step 2: Invite the Other Developer

1. In the repository's settings, click "Manage access."
2. Add the second developer as a collaborator by their GitHub username or email.
3. The second developer will receive an invitation to join the repository.

Step 3: Clone the Repository to Both Computers

1. Both developers should download **GitHub Desktop** and sign in with their GitHub accounts.
2. Clone the repository:
    - In GitHub Desktop, go to "File" > "Clone Repository."
    - Select the newly created repository from the list.
    - Choose a local folder to clone it to.

Step 4: Create Separate Branches

1. Each developer should work on a separate branch to avoid conflicting changes. In GitHub Desktop:
    - Click the current branch dropdown (top left) and select "New Branch."
    - Name the branch something descriptive (e.g., `feature-x`, `bugfix-y`).
2. Both developers now have separate branches for their work.

Step 5: Make Changes and Commit

1. Make changes to the code on each branch.

2. After making changes, commit them in GitHub Desktop:
    - Add a commit message describing the changes.
    - Click "Commit to [your branch name]."
3. Push the changes to the GitHub repository:
    - Click "Push origin" in GitHub Desktop to send the changes to the shared repository.

Step 6: Pull Changes from Each Other

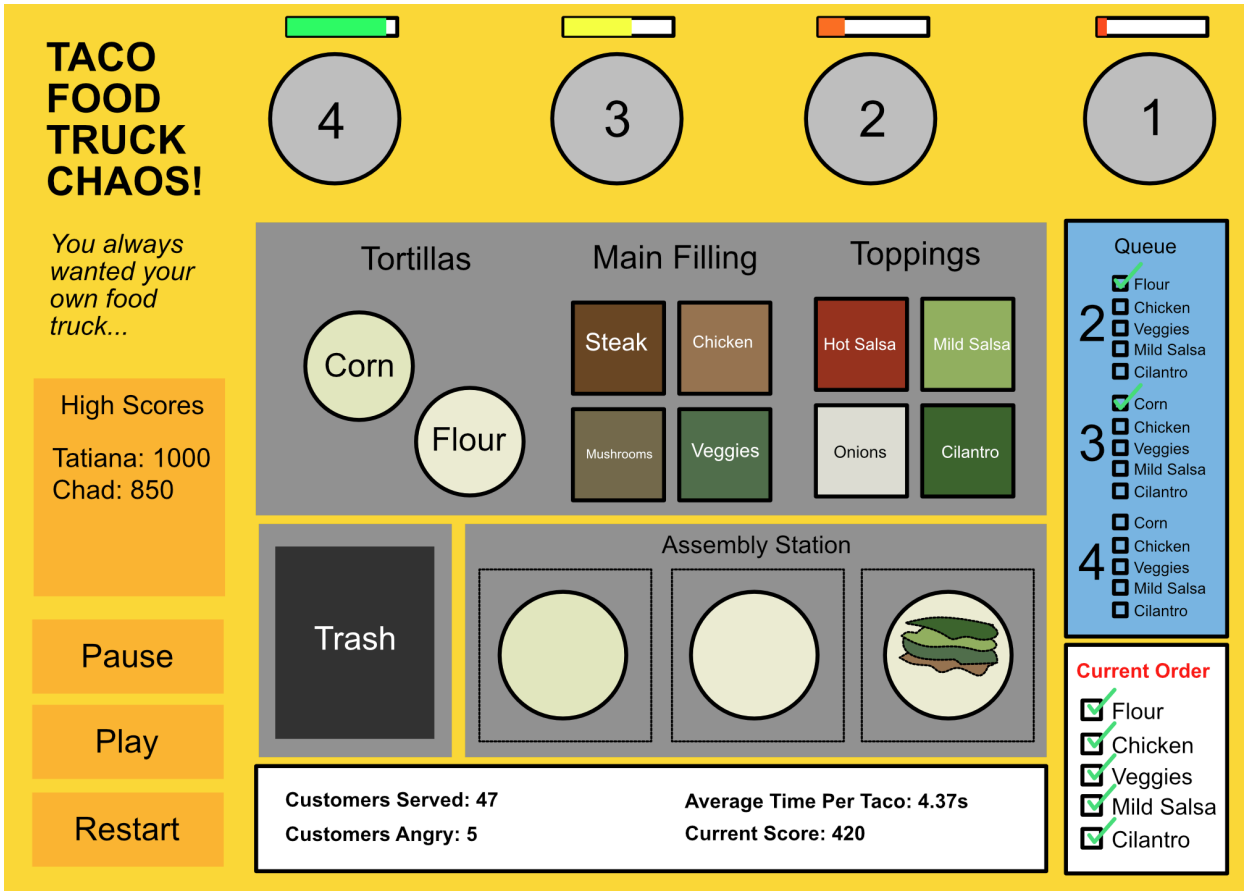
1. If one developer finishes their changes, they should push the branch to GitHub.
2. The other developer can then **pull the latest changes**:
    - Click "Fetch origin" and "Pull" in GitHub Desktop to download the latest updates from GitHub.

Step 7: Create Pull Requests

1. When ready to merge changes back into the `main` branch, create a pull request:
    - Go to GitHub (online) and open the repository.
    - Click the "Pull Requests" tab and create a new pull request from the branch.
2. Review each other's code and approve the merge.
3. Once reviewed, click "Merge Pull Request" to merge the changes into the main branch.

Summary:

1. **Create a repository.**
2. **Invite the second developer as a collaborator.**
3. **Clone the repository to both computers.**
4. **Work on separate branches.**
5. **Commit and push changes.**
6. **Pull changes from each other.**
7. **Create pull requests to merge changes.**



Game: TACO FOOD TRUCK CHAOS (Simplified)

Genre: Time Management/Simulation

Platform: PC (Browser-based, HTML5)

Game Overview: In this simplified version of TACO FOOD TRUCK CHAOS, players manage a taco food truck and serve customers while dealing with the chaos of a busy day.

Core Gameplay:

- **Players click on customers to take their orders, which appear in an "Order Queue."**
- **Players must click on ingredients to prepare tacos and serve them to the corresponding customers.**
- **Customers have patience meters that deplete over time. If a customer's patience runs out, they will leave.**

Game Mechanics:

- **Three types of ingredients: tortilla, meat, and salsa.**
- **Customers can order any combination of these ingredients.**
- **Players must serve customers in the order they arrive.**

Art Style:

- **Simple, 2D graphics with minimal animations.**
- **Bright colors to create a fun atmosphere.**

Sound and Music:

- **Basic sound effects for taking orders and serving customers.**
- **Optional: Simple background music loop.**

User Interface:

- **Click-based controls for taking orders, preparing tacos, and serving customers.**
- **Displays for the "Order Queue" and customer patience meters.**
- **Basic start and pause buttons.**

Game Loop:

- **The game consists of a single, endless level.**
- **Customers arrive with increasing frequency as the game progresses.**
- **The game ends when a specified number of customers leave due to impatience (e.g., 3 customers).**
- **Players aim to serve as many customers as possible before the game ends.**

Development Scope:

- **One endless level with increasing difficulty.**
- **Three types of taco ingredients.**
- **Basic customer and ingredient graphics.**
- **Simple sound effects and optional background music.**
- **Minimal UI elements (Order Queue, patience meters, start/pause buttons).**

This simplified spec focuses on the core gameplay elements and should be achievable within a 2-4 hour timeframe for a game jam. The emphasis is on creating a fun and challenging experience while keeping the scope manageable for inexperienced developers.

Game: DOG CATCH FRISBEE

Genre: Timing/Skill-based

Platform: PC (Browser-based, HTML5)

Game Overview: In DOG CATCH FRISBEE, players control a cute pixel art dog and must time their jumps to catch frisbees thrown by an off-screen owner. The game features a peaceful, outdoor scene with a grassy field and blue sky, reminiscent of the classic game Duck Hunt.

Core Gameplay:

- **Players press a button (e.g., spacebar) to make the dog jump.**
- **Frisbees are thrown from the bottom of the screen towards the dog at varying speeds and heights.**

- **Players must time their jumps to catch the frisbees when they are directly above the dog.**
- **Successfully caught frisbees earn points, while missed frisbees result in lost points.**

Game Mechanics:

- **The dog can only jump straight up and down, not side-to-side.**
- **Frisbees are thrown in a randomized pattern, with variations in speed and height.**
- **The game gradually increases in difficulty by throwing frisbees more frequently and at more challenging heights.**

Art Style:

- **Cute, pixelated 2D graphics.**
- **Colorful, simple background depicting a grassy field and blue sky.**
- **Animated dog jumping and wagging its tail.**
- **Frisbees with simple, recognizable pixel art design.**

Sound and Music:

- **Peaceful, looping background music to create a relaxing atmosphere.**
- **Sound effects for jumping, catching frisbees, and missing frisbees.**
- **Optional: Playful barking sound when the dog successfully catches a frisbee.**

User Interface:

- **Simple one-button control scheme (e.g., spacebar to jump).**
- **Score display at the top of the screen.**
- **Start, pause, and restart buttons.**

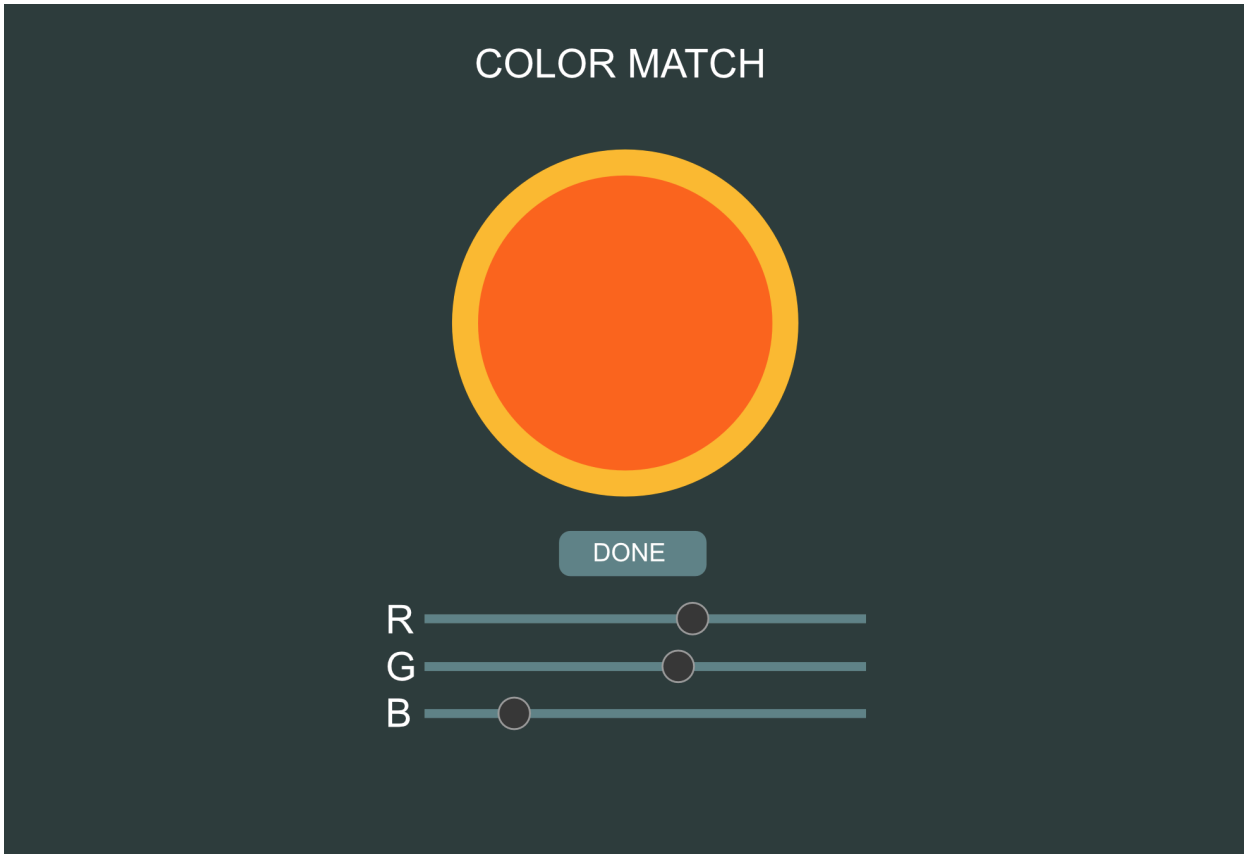
Game Loop:

- **The game consists of a single, endless level.**
- **Frisbees are thrown continuously, with increasing difficulty as the game progresses.**
- **Players aim to achieve the highest score possible before missing too many frisbees (e.g., 3 misses).**

Development Scope:

- **One endless level with increasing difficulty.**
- **Pixel art dog, frisbee, and background graphics.**
- **Simple jumping and frisbee catching animations.**
- **Peaceful background music and basic sound effects.**
- **Minimal UI elements (score display, start/pause/restart buttons).**

This simple and cute game concept focuses on timing and skill, making it an enjoyable and rewarding experience for players. The pixel art style and peaceful atmosphere create a charming and relaxing game that can be developed within a 2-4 hour timeframe for a game jam.



Game: COLOR MATCH

Genre: Puzzle/Educational

Platform: PC (Browser-based, HTML5)

Game Overview: COLOR MATCH is a puzzle game designed to help players, particularly graphic designers, improve their color matching skills. The game presents players with a target color and challenges them to mix primary colors (red, green, and blue) to create a matching color.

Core Gameplay:

- **Players are presented with a target color swatch.**
- **Using sliders or buttons, players adjust the levels of red, green, and blue to mix a color that matches the target.**
- **Players submit their mixed color for evaluation.**
- **The game provides feedback on how closely the mixed color matches the target, using a percentage score.**
- **Players earn points based on the accuracy of their color matching and the time taken to submit an answer.**

Game Mechanics:

- **Three sliders or buttons to control the levels of red, green, and blue, each ranging from 0 to 255.**
- **A submit button to evaluate the mixed color against the target.**
- **A timer that tracks how long players take to submit a color match.**
- **Bonus points awarded for quick and accurate color matching.**

Art Style:

- **Simple, clean, and professional design to appeal to graphic designers.**
- **Minimalistic user interface that focuses on the color swatches and mixing controls.**
- **Clear, easy-to-read text for instructions, feedback, and scores.**

Sound and Music:

- **Optional: Subtle, ambient background music that enhances concentration.**
- **Sound effects for submitting colors, receiving feedback, and earning bonus points.**

User Interface:

- **Prominent display of the target color swatch and the player's mixed color swatch.**
- **Sliders or buttons for adjusting red, green, and blue levels.**
- **Submit button for evaluating the mixed color.**
- **Score display and timer.**
- **Start, pause, and restart buttons.**

Game Loop:

- **Players progress through a series of increasingly challenging target colors.**
- **Each level consists of a set number of color matching challenges (e.g., 5 colors per level).**
- **Players must achieve a minimum accuracy score (e.g., 80%) to advance to the next level.**

- **The game ends when players complete all levels or fail to meet the minimum accuracy score.**

Development Scope:

- **5 levels, each with 5 color matching challenges.**
- **Simple color swatch and user interface graphics.**
- **Basic color mixing functionality using sliders or buttons.**
- **Scoring system based on accuracy and time.**
- **Minimal sound effects and optional background music.**

COLOR MATCH provides an engaging and educational experience for players looking to improve their color matching skills. The game's focus on a core mechanic and minimalistic design makes it well-suited for development within a 2-4 hour timeframe for a game jam, while still offering a valuable learning experience for graphic designers and other players interested in color theory.