

USER INTERFACE AND DESIGN

EXPERIMENT 2

Aim : To Design a UI where users recall visual elements (e.g., icons or text chunks). Evaluate the effect of chunking on user memory.

Procedure :

A. Home Screen (It contains Instruction Page)

Step 1: Create a Frame:

- In Figma, create a new frame (File → New Frame). Set the size to **1024x768px** for a standard desktop view.
- This will be your **Home Screen** where users start the task.

Step 2: Add Instructions:

- Use the **Text Tool (T)** to add a heading like "Memory Recall Task."
- Add a smaller body of text with instructions such as:
 - "You will be shown several groups of icons or text. After viewing, recall the items you remember."
- Use the **Text Tool (T)** to add more detailed instructions like "You will have 5 seconds to view the items. Then, recall them in the next screen."

Step 3: Start Button:

- Create a button at the bottom of the screen. To do this:
 - Draw a **Rectangle (R)** for the button.
 - Use the **Text Tool (T)** to add "Start."
 - Style the button (color, border radius) to make it stand out.

- Use **Figma's Prototyping Tools** (top bar → Prototype) to link this button to the next screen (Chunking Phase).
- You can also use **interactive components** like hover effects for more realism.

B. Chunking Phase (It Display Chunked Items)

Step 1: Create a New Frame:

- Create a new frame for the **Chunking Phase** (the second screen). This frame will display the icons or text.

Step 2: Design Chunked Items:

- Use **icons** or **text blocks** that users will have to recall. If you're using text, it could be short phrases or words. If you're using icons, you can either import them from **Figma's resources** or draw simple shapes using Figma's drawing tools.
- **For Chunking with Borders:**
 - Group 3-5 icons or text together in a box (use the **Rectangle Tool (R)**) to visually represent a chunk. You might want to create 3-4 groups.
 - Space these chunks out with some empty space in between them to ensure users can identify each chunk.
- **For Chunking without Borders:**
 - Place the elements next to each other without clear separation. This can be done by not using boxes and just visually mixing the items.

Step 3: Set the Viewing Time:

- **Time Simulation:** Figma does not have true timers, but you can simulate a fixed time by setting the next screen transition after 5 seconds:
 - Select the entire **Frame (Chunking Phase)**.
 - Under the **Prototype** tab, link this frame to the next screen (Recall Phase).
 - Set the interaction to "After Delay" and enter 5000ms (5 seconds).

C. Recall Phase

Step 1: Create a New Frame for Recall:

- This is where the user will recall the items they saw in the previous chunking phase.

Step 2: Recall Input (Multiple-choice or Text Input):

- **Option 1: Multiple-Choice:**
 - Create multiple options for the user to select (e.g., 4-5 icons or text options).
 - Use **Checkboxes** or **Radio buttons** to allow users to select what they remember.
 - Add a question at the top: "Select the items you remember seeing."
- **Option 2: Text Input:**
 - Create **Text Input Fields** where users can type what they remember. Create 3-5 input fields depending on how many chunks you showed.
 - This can be done by selecting the **Text Tool (T)**, adding a label ("Item 1", "Item 2"), and setting up input boxes.

Step 3: Submit Button:

- Create a **Submit** button at the bottom using the **Rectangle Tool (R)** and adding text like "Submit Recall."
- Add an interaction to move to the **Feedback Screen** after submission.

D. Result Screen

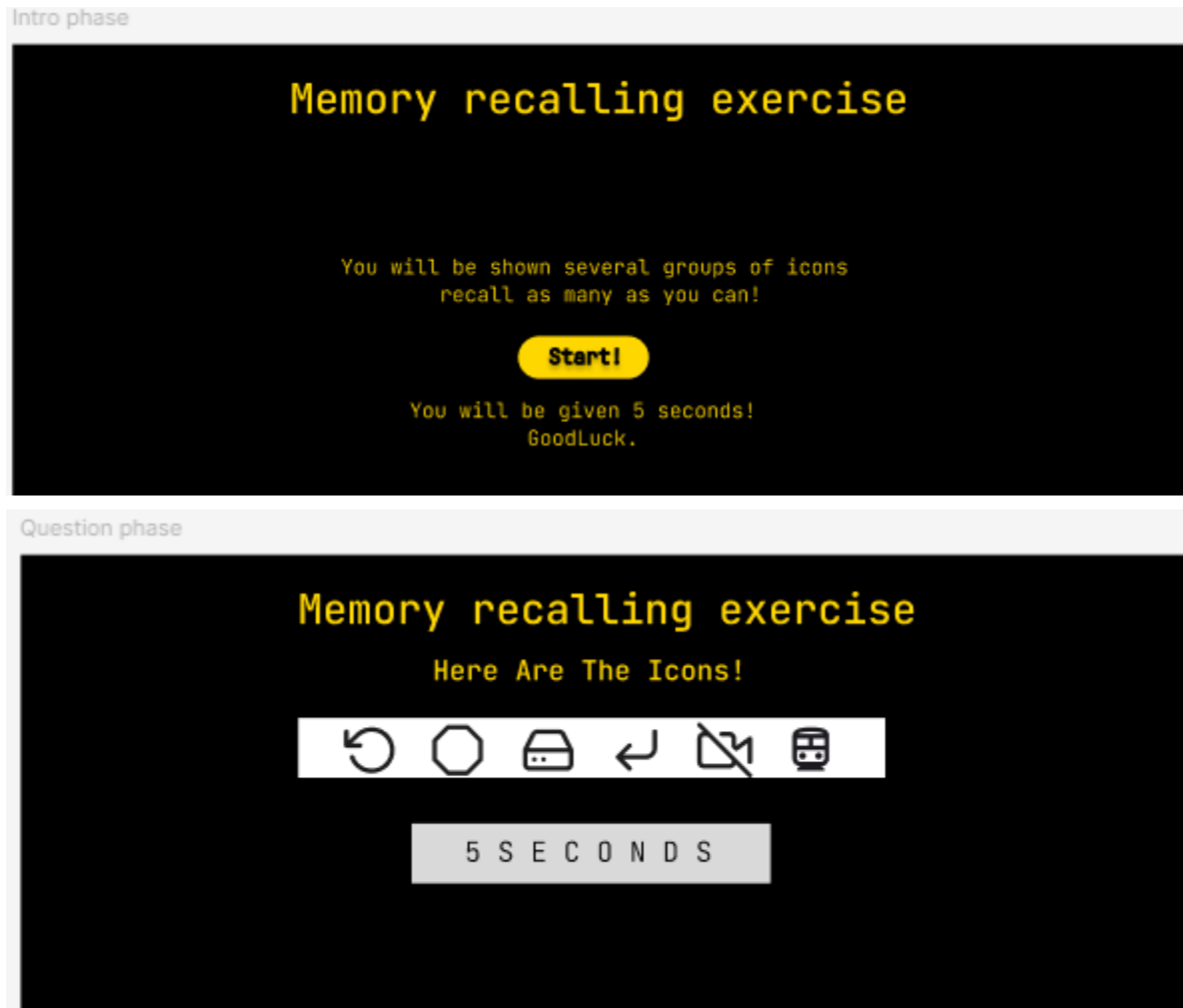
1. Create a Feedback Screen:

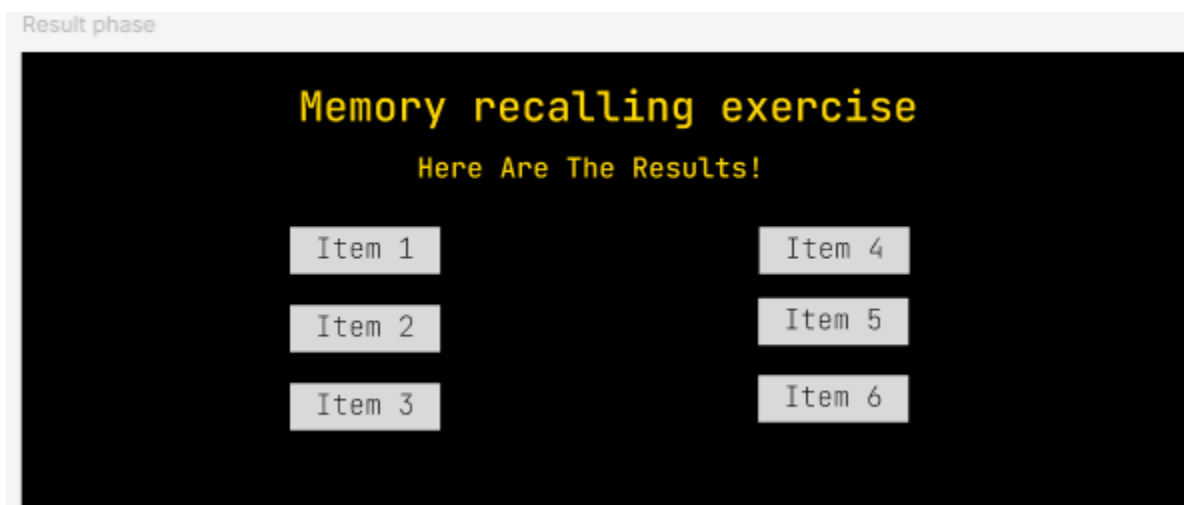
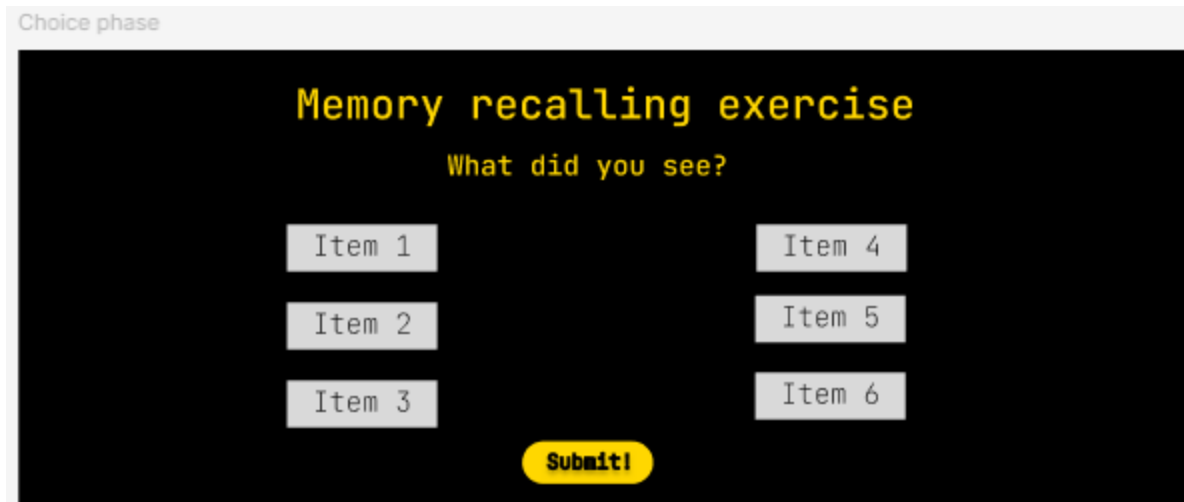
- After the user submits their recall, provide feedback.
- Add text like: "You recalled 4/5 items correctly!" or "Good job, you remembered 3 out of 5 items."

2. Analyze:

- For your experiment, you can vary the **chunk size** (3 vs. 5 items per chunk) and the **chunk type** (icons vs. text) across different test sessions to evaluate their impact on recall.

Output :





Result : Thus, a figma design for creating a game using memory recall and prototyping along with the use of chunking was successfully studied and executed.