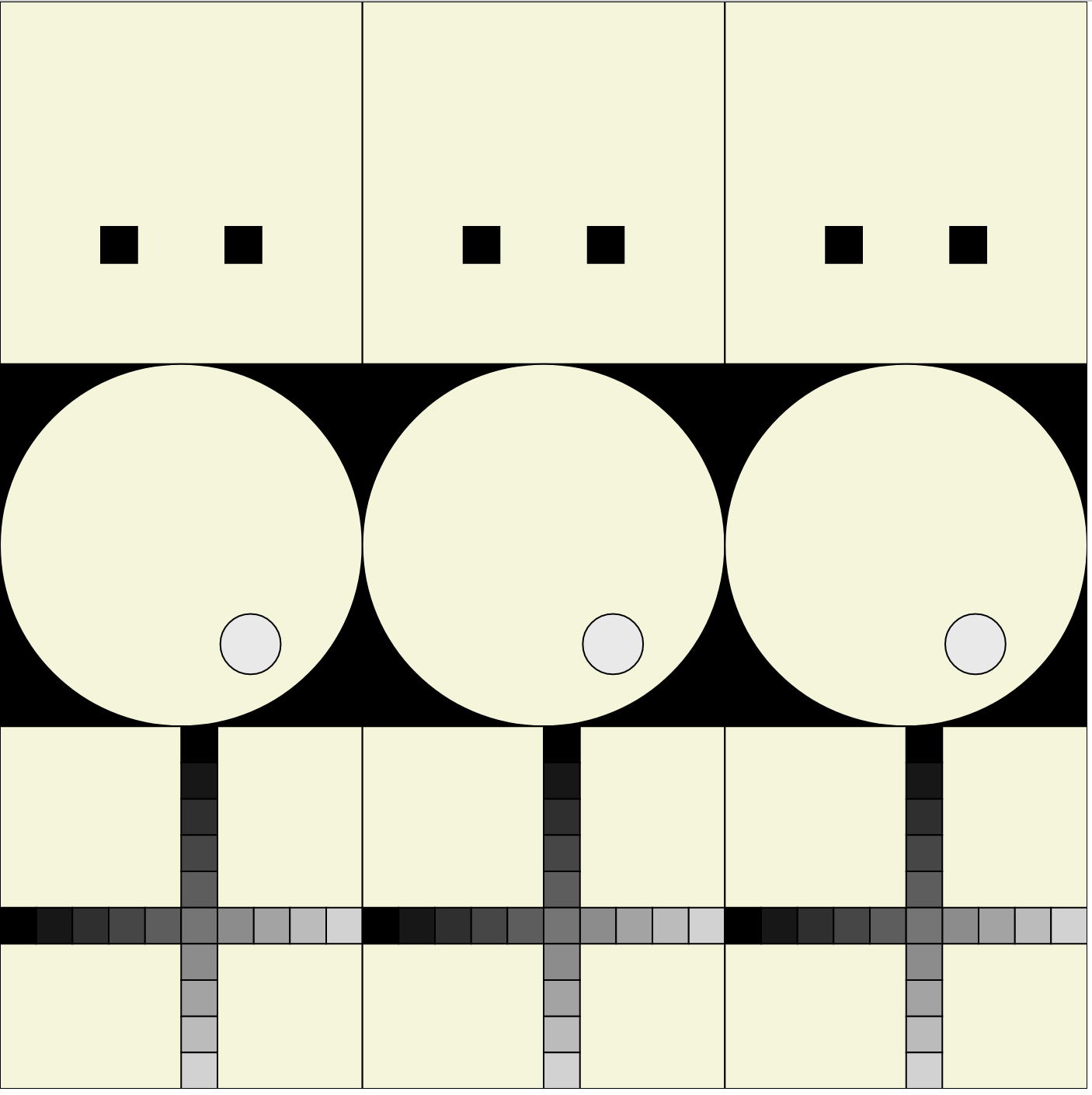
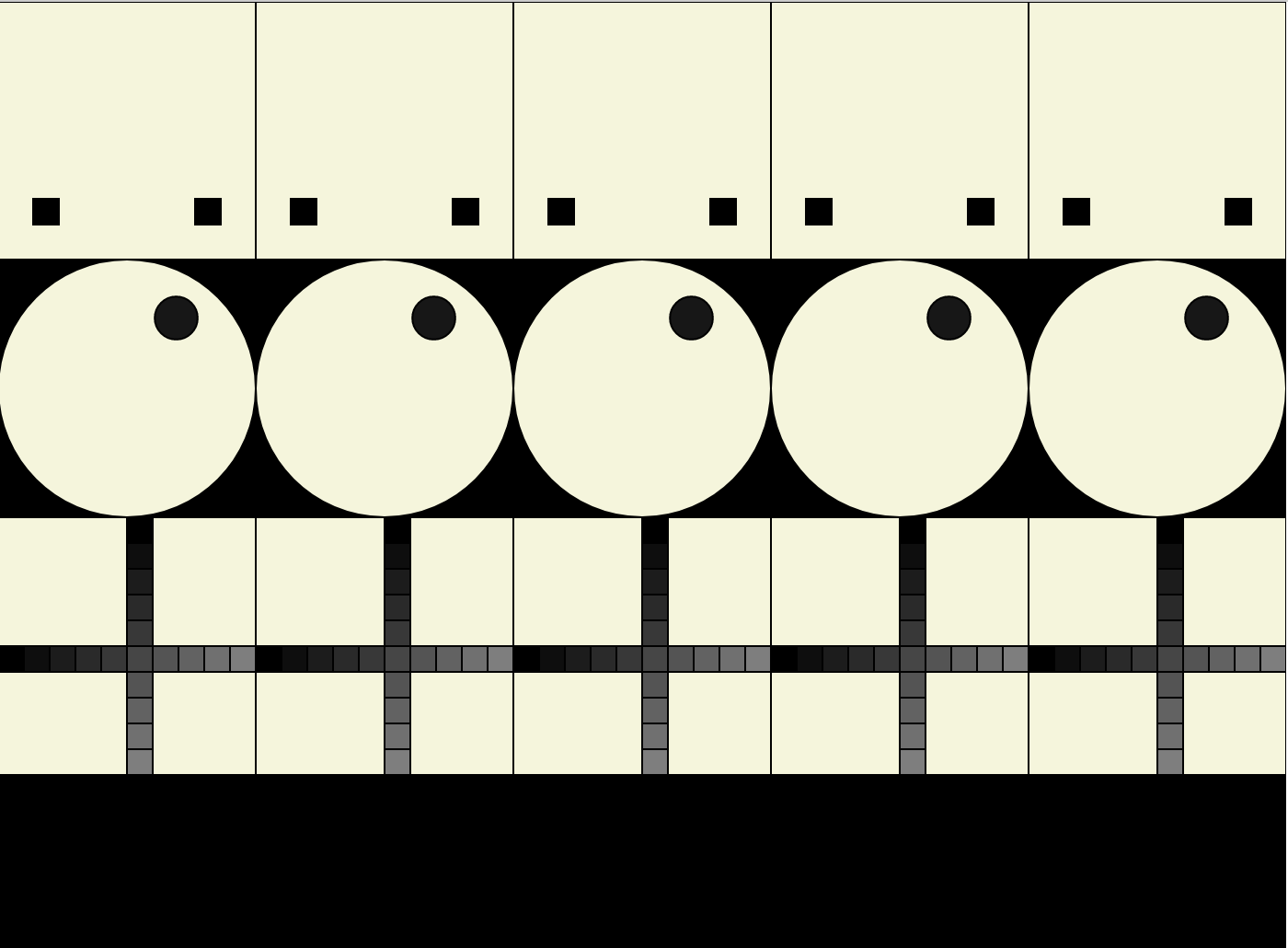
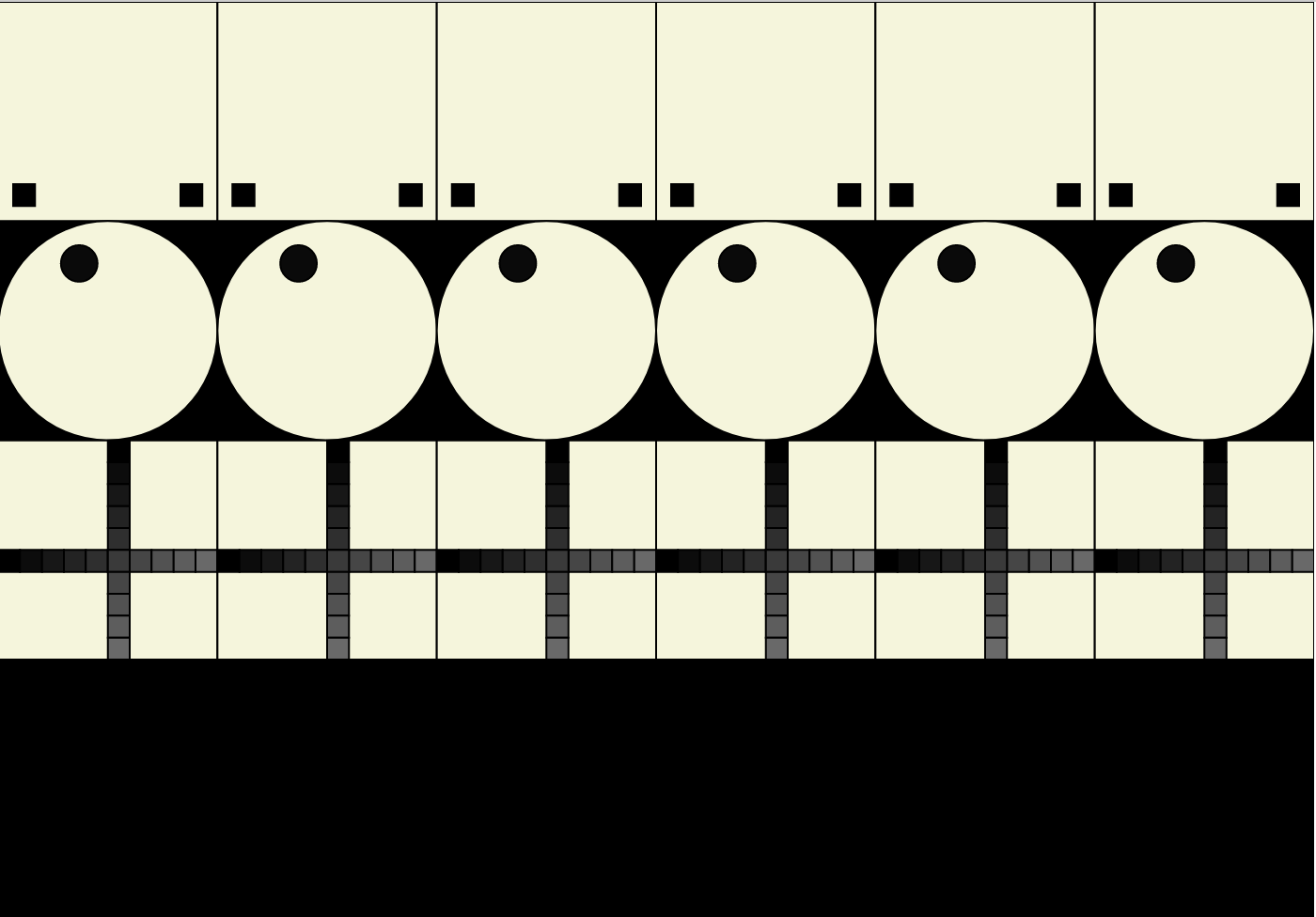
Coder's tools for this task: variables, conditionals, loops, and functions.

|  |
| --- |
| **Functionality Overview** |

As the program begins, you see a square canvas evenly divided into a 3x3 grid. Each column has 3 different decorated/active squares in it, and is repeated across the canvas. Two squares have continuous animated movement. Two squares have variations in color.

If a person types a number 3, 4, 5, or 6, the number of columns in the grid updates to match that number - still 3 squares tall, but the squares are smaller, and the size and motion of everything within the square scales down accordingly.

|  |
| --- |
| **Functionality Requirements** |

**Global Variables**

There should be only one global variable, for column count, which should be initialized to 3.

Setup Function

Size your square canvas at least 600 x 600 but not bigger than 1000 x 1000.

3 Interesting Square Functions

Design, plan, and create three distinct functions that create an interesting (but non-seizure-triggering) square. The three functions together form the basis for each column of the grid. Each function should accept 3 parameters:

1. x - coordinate for left side of square
2. y - coordinate for top of square
3. size - width and height of square

Function requirements:

1. Use a loop in at least 1 function.
2. Set color based on a variable (defined by p5 or by you) for at least 2 functions.
3. One function should have at least 2 continuously moving elements.
4. A different function has at least 1 continuously moving element (only one function may be non-moving).
5. Use p5's frameCount variable in at least 1 function.
6. Use p5's sin() or cos() functions in at least 1 function.
7. Never draw outside of the square.
8. All elements and motion scale to the size of the square.

Draw Function

1. Draw the background.
2. Calculate a local variable for the size of each square based on the global variable for column count and the width of the canvas.
3. Use a loop that will iterate across the columns (use the column count global variable) calling each of your three interesting square functions with the proper parameters. Use local variables to calculate x and y parameters to send along with the size of each square.

KeyTyped Function

If a person types "3", "4", "5", or "6", update the column count global variable to match.

|  |
| --- |
| **Reminders** |

Never forget:

1. Plan your program structure first, then code in small, testable chunks. Use the console to check for errors, test/print variables, and validate functions.
2. A header comment at the top of the main sketch IDs you, the date, the assignment. Cite any code references used that aren't directly from me.
3. Declare global variables above the setup() function definition. Use descriptive names.
4. The order of appearance for functions should be: setup(), draw(), then event handlers, followed by any custom functions.
5. Each function definition should be prefaced with a header comment explaining its purpose, its parameters (if any), and any return value.
6. Declare non-trivial local variables at the top of their code block. Use descriptive names.
7. Comment non-trivial variables with their purpose as you declare them.
8. Comment major sections of code, and also any tricky individual statements.
9. Format code with indentation (Format Document early and often) and blank lines between major sections and functions.

|  |
| --- |
| **Grading** |

|  |  |
| --- | --- |
| Only one global variable, column count, initialized to 3 | 5 |
| Setup function: |  |
| Reasonably sized square canvas | 5 |
| Draw function: |  |
| Background color set | 5 |
| Calculates size of a square properly | 5 |
| Loops across the canvas using column count: | 5 |
| x and y parameters for placing each square function properly calculated | 5 |
| Each of the 3 square functions are called with proper parameters | 5 |
| Event handler for key typed: |  |
| Checks if user typed "3", "4, "5", or "6" | 5 |
| Updates column count properly based on key | 5 |
| Interesting Square functions: |  |
| There are 3 square functions | 5 |
| Each function is interesting without potentially triggering seizures | 5 |
| Each accepts and uses 3 parameters: x, y, and size | 5 |
| A loop iterates properly | 5 |
| Color is based on a variable for at least 2 functions | 5 |
| A function has 2 continuously moving elements | 5 |
| A different function has at least 1 continuously moving element | 5 |
| frameCount is used properly | 5 |
| Sin() or cos() is used properly | 5 |
| Nothing is drawn beyond any function's square | 5 |
| All elements and motion scales to the size of the square | 5 |
| Style deductions? |  |
| Comments: header, variables, code sections (up to -5 pts) |  |
| Variable naming and declaration placement (up to -5 pts) |  |
| Code formatting: clean alignment, and v-spacing (up to -5 pts) |  |
| Total: | 100 |