

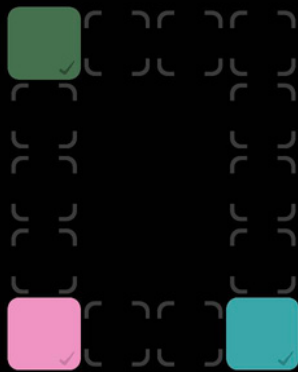
# 15-112 TP PROPOSAL

BY  
JOANNA TAN



**PALETTE**

**GAME BOARD**



**FINISHED GAME**



# BLENDOKU

## PROJECT DESCRIPTION

Blendoku is a color hue puzzle game. In the game, players will get a palette of colors blocks. The goal of the game is to arrange the color blocks such that they are in order according to their hues. Each level of the game starts with the color block palette and a game board onto which players can drag the color blocks. The game board may start with one or more color blocks already in position and may have more than one right answer. When the blocks are in the right color hue order, the player has passed the level.

## SIMILAR PROJECTS

This project is a recreation of the game Blendoku2. The basic gameplay rules of dragging color blocks into place and the basic game board will be similar.

However, instead of the set levels in Blendoku2, this project will allow players to select a difficulty level, then randomly generate the game board and the closeness of color hues between the color blocks. As the level of difficulty increases, the variety of game board shapes increases and the closeness of color hues between the color blocks increases.

After the player has passed a level, instead of going to the next level, the player can simply regenerate a game board with their preferred difficulty level.

# STRUCTURAL PLAN

The finalized project will be organized into the following functions:

- `generateGameBoard`: creates an instance of the `gameBoard` class that contains a randomly chosen game board shape.
- `generatePalette`: creates a palette by generating instances of the `colorBlock` class, with each `colorBlock` object representing a color block to be placed on the game board.
- `checkBoard`: checks if the board has all the color blocks in the right place

Other basic game play 15–112 graphics functions such as `appStarted`, `mouseDragged`, and `timerFired` will also be used.

Note: no external modules will be used.

# ALGORITHMIC PLAN

The most difficult part of this project is the generation of the game board and the examination of the player's answer.

The generation of the game board is random, but each game board has to have at least one right answer such that the color blocks are in the right order based on their color hues. To do so, color blocks will be stored as its RGB color code. Possible game board shapes will be stored as a 2D list, with a default value marking places where a color block can be filled, in a list. When the game starts, a board will be generated by randomly choosing a game board shape from the 3D list, then generating a starting color block. After this, other color blocks with close color hues will be generated by slightly changing one of the RGB values by a set value, and these blocks will fill the game board. Then, these color blocks will be stored in a list and be displayed as the palette. This way, there will be at least one right answer to the game board.

Because there may be more than one right answer, the game must constantly check if the player's answer is among the right answers. Instead of generating all the right answers, the game will just check that the player has placed all empty spots on the game board has been filled and that the color blocks differ by the previously set RGB change value.

# TIMELINE PLAN

**11/20 (TP 1)**

gameBoard class and colorBlock class

basic game board interface

- with hard-coded gameBoard instance and colorBlock instances
- game screen with palette at the top, empty game board at the bottom
- player can drag block from palette onto game board

start generateGameBoard and generatePalette

**11/30 (TP 2)**

finish generateGameBoard and generatePalette

checkBoard

**12/07 (TP 3)**

try adding new functions such as:

- generate the board such that several starting color blocks are already in place, and there is only one correct answer
- give hints: place a color block in the correct position

# VERSION CONTROL PLAN

I will create a branch and push to this GitHub repository for every new function added as well as any other major changes. I will commit the changes to keep track of them.

