

# Color Plus

**Part 1** Due in class on **11/20**

**Part 2** Due before class on **11/29**

**Part 3** Due before class on **12/6**

**Final** Due before class on **12/13**

For this project, you will create a two-player action puzzle game. This project is intended to help you increase your skills in the following areas:

- Resourcefulness and Persistence
  - Searching for answers online, bookmarking commonly used webpages
  - Asking for help from Ira, the TAs, or other students when you're stuck
- Logical Thinking
  - Carefully thinking through step-by-step logic for the program
- Using Unity
  - Creating new projects, files, and assets
  - Attaching scripts to objects
- Using GitHub
  - Storing and sharing your program as it changes over time
- Publishing a game to your own website
  - How to create a basic website and the tools required
  - FTP, basic text editor, Unity
- Organization
  - Well-organized and well-named files, folders, assets, and variables
  - Nicely formatted code with matching {} and indentation
  - Comments to help understand the code
- Types of variables
  - int, float, bool
  - How to create them and modify them
- Data structure: Arrays, including 2D arrays
  - What they are, how to create them and modify them
- Mathematical expressions
  - Comfortable with adding, subtracting, multiplying and dividing
- Conditional expressions
  - if(), else if(), else()
- Iterative statements
  - for() loops

Wow, that's a lot! It's amazing what you'll be able to learn in just a few weeks of effort!

## What To Hand In

For each part of this project, **deliver your work as follows:**

- a) Within Unity, make a WebGL build (File -> Build Settings)
- b) Upload the .html file, Build folder, and TemplateData folder to your website.
- c) Upload your project files (especially .cs) to GitHub.
- d) Find your tab on the spreadsheet below and add your website and GitHub links for the appropriate part of the project:

<https://docs.google.com/spreadsheets/d/1o8qh3B-h3RI6rQMotEtjJlbzv-dOxfVamxa-ZUVYU>

## Part 1

By the deadline, prepare a short (~4 minute) presentation to give in class. Update your slides here:

[https://docs.google.com/presentation/d/1ZFf-Z\\_QrighFVbEO25EmafPM-x8MbNVZxyv1dksb7YU/edit#slide=id.g29cfbf6b94\\_0\\_45](https://docs.google.com/presentation/d/1ZFf-Z_QrighFVbEO25EmafPM-x8MbNVZxyv1dksb7YU/edit#slide=id.g29cfbf6b94_0_45)

The presentation should include:

- 1) A description of what your main Update() loop will do.
- 2) A list of each method you intend to implement with arguments, return values, and a one sentence description of what it does (i.e. the method header). For example:
  - a. // this method will check if the player should score any points now  
void CheckScore()
  - b. // Starting from the given position in the grid, this method will return  
// how many cubes in a row match the given cube's color  
int FindColorMatchInRow (int x, int y)

You will be evaluated on the quality of your content as well as the polish of your presentation. You may collaborate with any number of people for this part of the project, as long as you credit them appropriately. If you want to present in combination with someone else, that's fine too, as long as you notify Ira in advance and all people have a full understanding of what's being presented.

You may work with a partner, but ensure you can write all the code individually. If you finish this section have extra time, start programming your game!

### **Part 2**

By the deadline, make significant progress toward the final game. Here are some features that should be completed, and parenthetical features should be done soon:

- Grid of cubes
- Turns exist, including showing new random cube
- Keyboard input is processed properly (including row-is-full situations)
- Mouse input is processed properly (including all situations)

### **Part 3**

By the deadline, the game should be nearly complete. In addition to all the features in Part 2, here are additional features that should be done:

- Matches are detected properly
- Players can score points and score is tracked
- The game ends properly (either win or lose)
- If players fail to place a cube in the grid in time, they are penalized correctly

### **Final**

By the deadline, the game should be complete and bug free. Spend the final week implementing any missing features and debugging problems. In a game this complicated, there may be strange bugs. Ensure that your game works properly in all cases and there are no bugs. Include a link to your game and GitHub repository in the spreadsheet on your tab, as usual.

### **A Note about the Game Design**

The description below is something that you might get from a game designer, if you were working at a game company. Once you read it and start thinking about it, if you have questions about the design, you may feel free to ask questions to get clarifications of design intent by e-mailing Ira.

### **Useful Link on GUI in Unity**

As you'll see in the game design below, there are several places where it would be desirable to display information to the player on screen. This walkthrough Ira made might be useful: <https://youtu.be/9aO0BvBLrys>

Here's an older link too: <https://unity3d.com/learn/tutorials/topics/user-interface-ui>

And specifically #5 on UI Text. You can experiment within Unity by choosing  
GameObject -> UI -> Text.

# Color Plus

## Game Design Document

*Color Plus* is a cooperative two-player action puzzle game, playable in a web browser. One player uses the keyboard, and the other uses the mouse.

### Gameplay

The game is timed, and takes 60 seconds (tunable).

The game is played on an 8 x 5 grid of large white cubes, and takes place in turns. One turn takes 2.0 seconds (tunable).

Above the grid, there is a "Next Cube" area. At the beginning of each turn, a random cube appears in that area. The valid colors are:

- 1) Blue
- 2) Green
- 3) Red
- 4) Yellow
- 5) Magenta

### Keyboard Input

If a player presses 1, 2, 3, 4, or 5, the cube should immediately disappear from the Next Cube area, and appear in the 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>, or 5<sup>th</sup> row. The cube should appear in a random, available column of the appropriate row. A particular location is "available" if it contains a white cube. The Next Cube area will be temporarily empty until the next turn starts.

If a player fails to press one of those numbers before the turn ends, then a random available cube (i.e. white) turns black. In that case, a new cube appears in the Next Cube area as usual.

For example, from the start of the game:

0.0 seconds: Whole grid is white. Yellow cube appears in the Next Cube area

1.2 seconds: A player presses 2. Yellow cube disappears from Next Cube area and appears in a random location in row 2.

2.0 seconds: A Blue cube appears in the Next Cube area

4.0 seconds: Since the player failed to press 1, 2, 3, 4 or 5, a random cube in the grid turns black, the Blue cube disappears, and a new random color cube appears in the Next Cube area.

### Mouse Input

Any time a player clicks a colored cube (non-white, non-black) in the grid, the cube should activate.

An activated cube should highlight in some way, like enlarging a little bit or having a spotlight shine on it. Only one cube can be active at a time. If a player clicks an active cube, it should deactivate.

If a player clicks a white cube that is adjacent to the active cube (including diagonals), the active cube moves to that location instantly (and remains active). The location that the active cube just vacated should become a white cube.

Cube movement due to clicking happens instantly and not on turn boundaries. Therefore, with fast clicking, it's possible to make many moves in a single turn.

Note: The players can only move a colored cube to a white cube, so if the players click a colored cube and then another colored cube, no switching happens, and the second colored cube ends up being the active one. For example:

- a) Player clicks a colored cube. It activates.
- b) Player clicks an adjacent white cube. The colored cube and the white cube seem to swap, and the colored cube remains active.
- c) Player clicks a colored cube. The recently-clicked colored cube becomes active.
- d) Player clicks a far-away white cube. Nothing happens.
- e) Player clicks a black cube. Nothing happens.
- f) Player clicks the currently active cube. It deactivates.

### Scoring

If at any point there are 5 colored cubes all of 1 color in a + configuration, those cubes all turn black, they are no longer movable, and the players score 10 points (tunable). If one of the cubes that just scored happened to also be the active cube, deactivate it.

If at any point there are 5 colored cubes, one of each color, in a + configuration, those cubes all turn black, they are no longer movable, and the players score 5 points (tunable). If one of the cubes that just scored happened to also be the active cube, deactivate it.

Players also lose 1 point whenever the players fail to type 1, 2, 3, 4, or 5 before the turn ends. Score can never go below 0, though.

### End of Game

The game can end in the following ways:

- 1) The time runs out. The players win as long as they have a positive score!
- 2) By pressing 1, 2, 3, 4, or 5, the Next Cube attempts to go to a row that has no available spots (i.e. white cubes). The players lose instantly.
- 3) By not pressing 1, 2, 3, 4 or 5, at the end of a turn, the game attempts to turn a random cube black, but all locations are full. Note: this isn't possible with current tuning, but if we change some parameters, this could happen. In this case, the game ends instantly.

**! Challenge by Choice:** Do any or all of the following additional tasks. I've included categories below, but you can pick and choose however you want.

#### Game Flow

- Add a Start screen that has a large "Play" button. If the player clicks it, the game starts.
- Add a Game Summary screen that appears when the game ends. It should include:
  - A big "Victory!" if the players won, or a bit "Defeat!" if the players lost.
  - The players' score.
  - A big [Play Again] button that will return the players to Gameplay.

#### UI (User Interface)

- The current time remaining should be clearly visible to players.
- When the time gets to 10 seconds or less, the timer should turn red.
- The Next Cube area should be labeled with the words "Next Cube"

#### Game Design Complexity

- 1) When picking a random Next Cube, avoid repeating the same color that was picked the previous turn.
- 2) It's possible for players to create two plusses simultaneously, if the middle cube is part of two plusses. In that case, give the players bonus points and turn all 9 cubes black. Each group of 5 cubes must be all the same color, or one of each color (but all 9 cubes need not be the same color).
- 3) Make the scoring more interesting so the game is more fun. Possible ideas:

- a. Add some sort of streak bonus, if the players can score multiple plusses within a few turns of each other.
- b. Reward plusses in the middle of the grid more than plusses on the edge of the grid.
- c. Give a score bonus at the end based on how well the players partitioned the grid. Lots of separate partitions should yield more points (since it's hard to play if there are lots of separate partitions).

#### Feedback to the Player

- Account for color blind players by adding a letter to the front of each colored cube, based on the color. Perhaps A, B, C, D, E.
- On the Start Screen, include some instructions about how to play
  - For extra challenge: When showing instructions, don't just use text. Include images of some sort to communicate more clearly.
- Whenever players mouse-over any location (but before clicking on it), give a visual indication that it's clickable (change the color a bit, enlarge it slightly, etc. Your choice of what to do, but do something.)
- Somewhere on screen, display the player's current score.
  - Along with the player's current score, show the word "Score."
- On the Game Summary screen, show additional information like:
  - Total cubes that appeared of each color
  - High score
  - Scores from the 10 most recent games
  - A tip on how to play better
  - Anything else potentially interesting to the player

#### Sound Effects

- Whenever the player activates a cube, deactivates a cube, moves a cube, the players fail to place a cube in the grid during the turn, a cube is placed in a row, a new cube appears, the players score, or the game ends in defeat or victory, play a sound effect (SFX).
- Play music that loops, so there's constantly music playing during the game.

#### Visual Polish

- Make the game look prettier, by any means you deem appropriate.