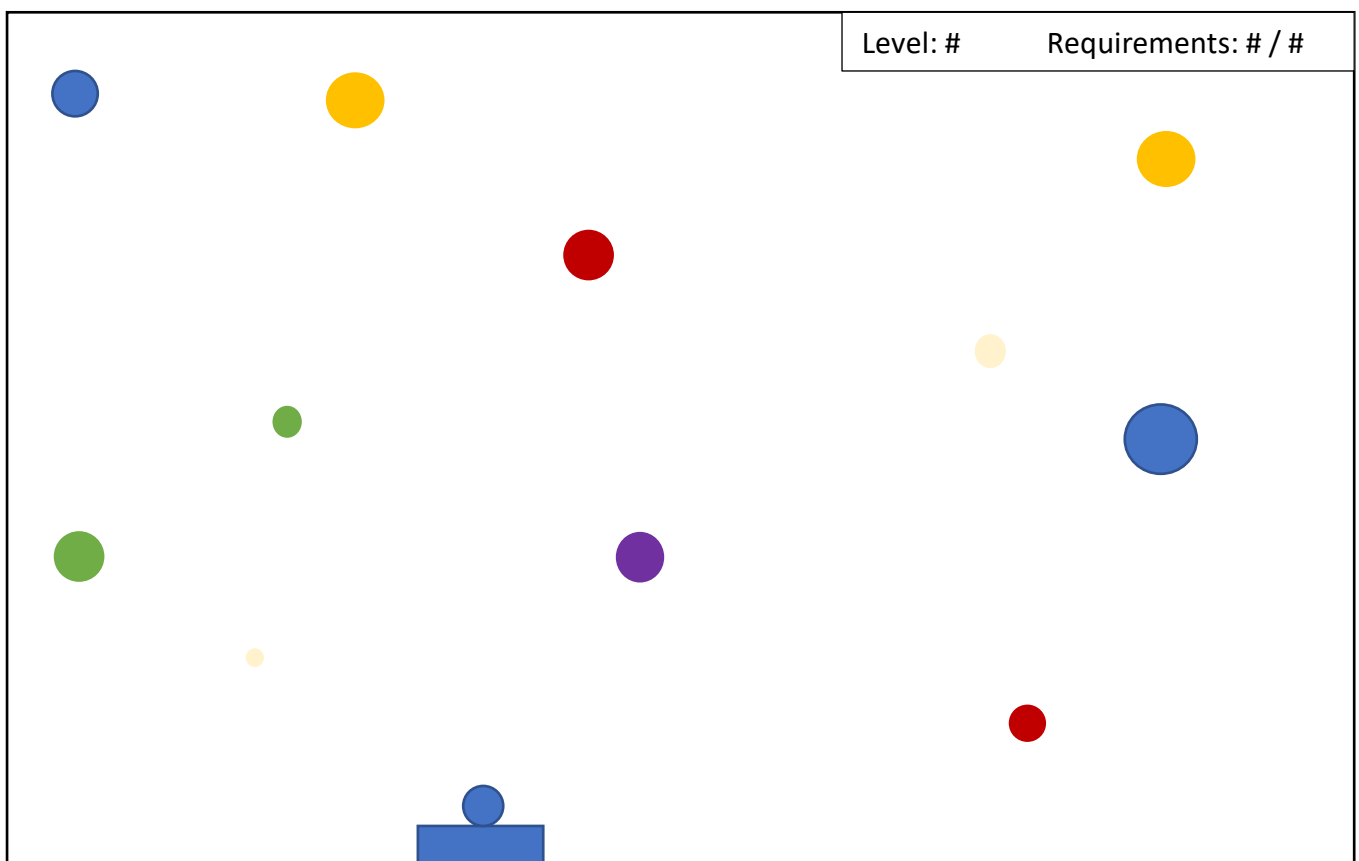


## 1. Overview:

- a. My project is about a game of combination bounce and collision detection.
- b. Description
  - i. For my game, there's 6 level: red, orange, yellow, green, blue, purple
  - ii. There will have 6 different colors of circle start at a random position from the top of the canvas and fall to the bottom of the canvas. The circle drop speed will increase in each level.
  - iii. There will have a box below, each level the computer will choose a color of the paddle, and the box should get the same color as the circle that falls down from the top.
  - iv. Each level, there's a requirement for the number of the circle that the box should get.
    1. If the paddle touch different color, it will go a level below, which means the paddle color will change back.
      - a. If it's in level one, then the color will keep in the same
    2. If the paddle touch the same color, the circle bounce back, the computer will also Record the number of times when the paddle gets the same color
      - a. If the number of times reaches the requirement, level up, which means the paddle color will change and there will have a new requirement.
      - b. If user reach highest level requirement, the canvas color will change
  - v. The user can press Q or click QUIT bottom to stop the game, which means the canvas will be clear except the paddle below
  - vi. The user can press S or click START key to start the game

## 2. Sketches:



### 3. Components

- a. Walls – the four sides of the canvas
  - i. canvasWidth
  - ii. canvasHeight
  - iii. CanvasColor
- b. Text – requirement shows right top corner
- c. Ball – Ball – 6 different color and different size; ball bounce off the top and sides of walls; ball bounce off the top of the paddle; ball falls off the bottom of the screen; balls drop from the top.
  - i. size
  - ii. x
  - iii. y
  - iv. color
  - v. speed
  - vi. next x
  - vii. next t
  - viii. drawBall()
    - 1. draw random circle with random size and 6 random color
- d. Paddle - 6 different colors; paddle moves horizontally at the bottom of the canvas by using the mouse position; paddle stays inside of the left and right boundaries and never goes off.
  - i. Hight
  - ii. width
  - iii. X
  - iv. y
  - v. Color
  - vi. Speed
  - vii. drawPaddle()
    - 1. draw a rectangle using x, y, height, width, color
- e. Start button - a button that when the user clicked, the ball will start to drop down and the game starts
- f. Clear button – a button that when the user clicks, the ball will disappear and the game will end
- g. Level- a while loop if the user reaches the level requirement.
- h. Game Over – the message that displays when the game is the end or user reach the highest level.

### 4. Interactions

- a. Ball movement - the ball starts at a random position and starts move down with a requirement speed. If requirement ball touch the paddle it will bounce off and disappear in any side of walls.
  - i. MoveBall()
    - 1. If the color is not a requirement color

- a. Change the Y position base on move Y
  - b. If ball reach the paddle, set requirement = False
- 2. if the color is requirement color
  - a. change ball x and y base on move x and move y

- b. Paddle Movement – the paddle moves left an right across the bottom of the canvas; the paddles stays within the boundaries of the left and right walls; paddle moves along a defined on the y axis and moves based on the location of the y position of the mouse.
- c. Ball and Paddle Collision – if the requirement ball collides with the paddle, the ball bounce off, else requirement = false

## 5. Game Control Algorithm

- a. Draw canvas
  - i. Clear the canvas
  - ii. Draw shapes
  - iii. Move shapes
  - iv. Collison checks
  - v. Requirement checks
- b. Game loop
  - i. Animation frame
  - ii. Game over check
  - iii. Level checks
  - iv. Number of ball records

## 6. Build steps

- a. Paddle
  - i. Define the paddle
  - ii. Draw the paddle
  - iii. Move the paddle
- b. Balls
  - i. Define the balls raw the balls
  - ii. Get the ball moves
  - iii. Get the ball falls off the canvas
- c. Paddle and ball collision - get the ball bounce off the paddle
- d. Clear button - setup the clear button to reset the ball, paddle and level to starting position
- e. Start the button – setup the start button to start the ball moving
- f. Game over- setup a game over Boolean variable; set the game over to false when non requirement balls collision the paddle; set the game over bottom when the highest level requirement = true; set the game over button when the clear button clicked; develop a game over function that draws a gave over message on the canvas; modify the game loop to any call draw canvas when the game over Boolean is true.