

2a. My program is essentially a memory game board created within Studio Code in JavaScript. The purpose of the program is to allow the user to play Memory, a game involving two players taking turns trying to match cards by flipping them over two at a time. The program itself creates 12 cards with 6 pairs of color dots to be matched randomly put on the 12 cards. It allows two users to take turns flipping two cards over and then either keeping them flipped over if they are a match or flipping them over once more to cover the color dot. The video illustrates that the cards can be flipped to play the game and it shows the random card placement.

2b. My project was created independently, so I had to break down the steps to create this in a manageable way for myself. I started with just creating the card which involved a cover, a background and a color dot. My main difficulty even at this early step was figuring out how to make the color dots random which started to take up the time I had to make this project, so I decided to make that a separate part of the program to not inhibit the creation process and first focused on making the blank card objects and covers. From there I had to go a step up and create the whole board of blank cards as a function utilizing the previous functions I made to draw a single card. At this point I was able to create a function to solve my earlier problem that shuffled the card order made in the board because the cards were stored as an array to be put on the screen rather than just directly put on the screen.

2c.

```
function drawCard(color, cardNumber, cardCoverNumber)    {

    createCanvas(cardNumber, 86, 92);
    setActiveCanvas(cardNumber);
    setPosition(cardNumber, cardXPosition, cardYPosition);
    drawCardBase(cardXPosition, cardYPosition);
    drawSymbol(color);
    createCanvas(cardCoverNumber);
    setActiveCanvas(cardCoverNumber);
    setPosition(cardCoverNumber, cardXPosition, cardYPosition);
    drawCardCover(cardXPosition, cardYPosition);

}
```

This function draws an individual card including a symbol (the color dot), a background and a cover. Each of these aspects is created using its own function to make the individual part, but by combing all of these into a single function, I was able to later use this to create a usable entity instead of three different shapes. This function also allows the cards to be placed based on the inputted cardNumber so as to have an organized board.

2d.

```

function drawBoard() {
  var cardColors = [];
  for (var card=0; card<6; card++) {
    cardColors[2*card] = 2*card;
    cardColors[2*card+1] = 2*card+1;

  }
  cardColors = shuffle(cardColors); //Shuffles the cards
  for (card=0; card<12; card++) { //sets twelve cards
    if (cardColors[card] === 0 || cardColors[card] === 10) { //This should color each card their respective
color
      cardColor = "red";
    } else if (cardColors[card] === 1 || cardColors[card] === 5) {
      cardColor = "green";
    } else if (cardColors[card] === 2 || cardColors[card] === 6) {
      cardColor = "blue";
    } else if (cardColors[card] === 3 || cardColors[card] === 7) {
      cardColor = "yellow";
    } else if (cardColors[card] === 4 || cardColors[card] === 8) {
      cardColor = "purple";
    } else if (cardColors[card] === 9 || cardColors[card] === 11){
      cardColor = "orange";
    } else {
      cardColor = "black";
    }
  }

  if (card === 1) { //this will name the card, so it can be its own canvas
    cardNumber = "One";
    cardCoverNumber = "1";
  } else if (card === 2) {
    cardNumber = "Two";
  }
}

```

```
    cardCoverNumber = "2";
} else if (card == 3) {
    cardNumber = "Three";
    cardCoverNumber = "3";
} else if (card == 4) {
    cardNumber = "Four";
    cardCoverNumber = "4";
} else if (card == 5) {
    cardNumber = "Five";
    cardCoverNumber = "5";
} else if (card == 6) {
    cardNumber = "Six";
    cardCoverNumber = "6";
} else if (card == 7) {
    cardNumber = "Seven";
    cardCoverNumber = "7";
} else if (card == 8) {
    cardNumber = "Eight";
    cardCoverNumber = "8";
} else if (card == 9) {
    cardNumber = "Nine";
    cardCoverNumber = "9";
} else if (card == 10) {
    cardNumber = "Ten";
    cardCoverNumber = "10";
} else if (card == 11) {
    cardNumber = "Eleven";
    cardCoverNumber = "11";
} else {
    cardNumber = "Zero";
```

```
    cardCoverNumber = "0";  
  }  
  drawCard(cardColor, cardNumber, cardCoverNumber); //This part will draw all of the cards onto the  
board  
  if (cardXPosition < 180) {  
    cardXPosition = cardXPosition + 106;  
  } else {  
    cardXPosition = 10;  
    cardYPosition = cardYPosition + 112;  
  }  
}  
}
```

My main abstraction was splitting the board down into its component parts before using the drawBoard function to combine all of the abstracted parts such as the shuffling of the cards or the creation of the cards. This allowed me to focus on parts that would not directly interact at first, the making of the cards and the shuffling of the cards, before then making the program more complicated.