Unit 4: Game Development

In this unit, you will use CodePuppy to learn about:

1. How to create and manipulate objects on a 2 dimensional cartesian grid
2. How to handle user interactions to make purposeful playable games
3. How to check code for and correct logical and syntax errors

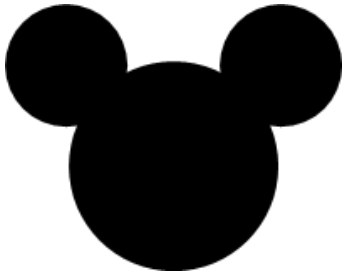
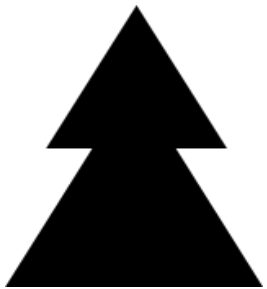
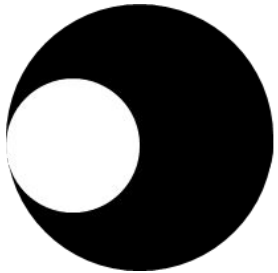
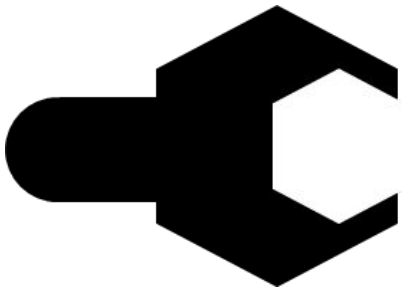
Exercise 1

Enter and run the following code:

```
circle(100,100,50,"black")
```

Study Drills:

Write the code to produce the following shapes. You'll need to use black and white shapes...

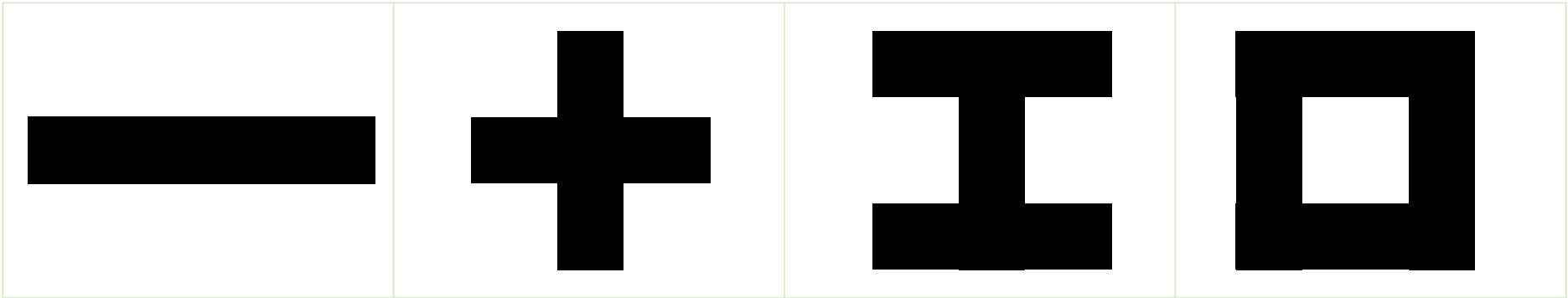
Exercise 2

Enter and run the following code:

```
rectangle(100,100,400,200,"black")
```

Study Drills:

Write the code to produce the following shapes. You must only use rectangles to make these shapes.



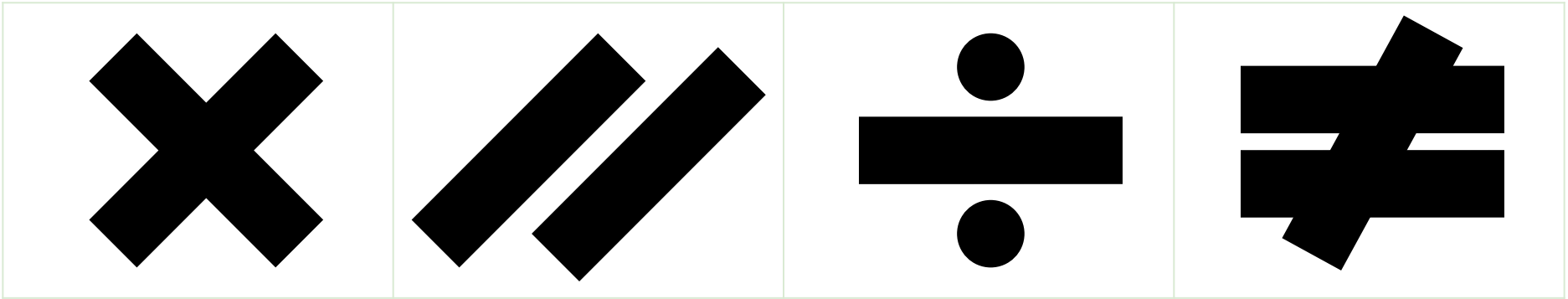
Exercise 3

Enter and run the following code:

```
rectangle(100,100,400,200,"black").rotate(45)
```

Study Drills:

Write the code to produce the following shapes



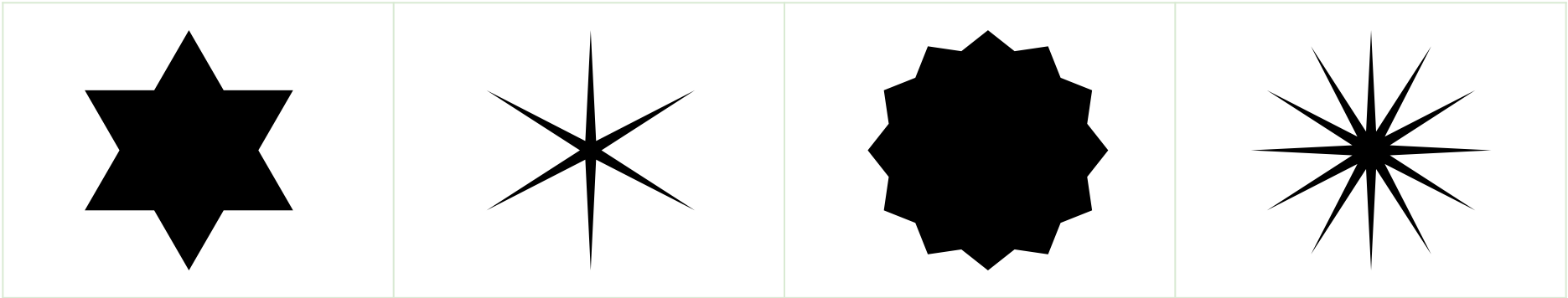
Exercise 4

Enter and run the following code:

```
complexstar(200,200,100,50,5,"black")
```

Study Drills:

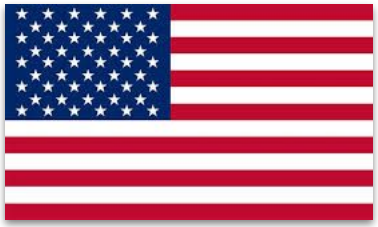
Write the code to produce the following shapes



More Study Drills





```
canvas("lightblue")
head(400,200)
mouth(400,250)
eye(230,100)
eye(230,200)
carrot(200,300)
body(200,450)
arm(400,450,90)
arm(400,500,-90)
coal(400,520)
coal(370,550)
coal(430,550)
foot(100,100)
foot(100,200)
```

Exercise 5

This code contains all of the bits you need to make Olaf.

Study Drills:

Copy the code into Code Puppy.

Many of the values in the program are wrong. Work out what these numbers should be to make the picture of Olaf display properly.

Exercise 6a

Enter the following code:

```
function onMouseDrag(event) {
    circle(event.point.x, event.point.y, 10, pencolor);
}
```

Study Drills:

1. Make sure that the code allows the user to draw a line on the canvas
2. Change the code to make the line thicker
3. Change the code to make the line thinner

Now, move on to Exercise 5b. You are going to continue to add code to this program

Exercise 6b

Add the following code to your program. This is going to allow the user to choose a line colour:

```
blue = square(25,25,30,"blue");
red = square(25,70,30,"red");
green = square(25,115,30,"green");
```

Study Drills:

1. Add nine more squares so that the user has a choice of twelve colours

Now, move on to Exercise 5c. You are going to continue to add code to this program

Exercise 6c

Add the following code to your program. This is going to allow the user to choose a line colour:

```
function onMouseDown(event) {
    if (blue.hitTest(event.point)) {pencolor = "blue";}
    if (red.hitTest(event.point)) {pencolor = "red";}
    if (green.hitTest(event.point)) {pencolor = "green";}
}
```

Study Drills:

1. Add the ability to select from any of the twelve colours in the palette
2. Improve the program, to allow the user to choose the thickness of the line they draw
3. Improve the program, to allow the user to use an eraser to remove lines they have drawn
4. Improve the program, to allow the user to draw a solid rectangle by clicking at the top left and bottom right corners

Exercise 7a

Enter the following code:

```
apple = new Layer()
#write some code here to draw an apple

basket = new Layer()
#write some code here to make a basket
```

You are going to use this code to make an apple drop game.

Study Drills:

1. Write the code to make an apple where the comment appears in the program
2. Write the code to make a basket where the comment appears in the program

Exercise 7b

Add the following code to your program:

```
function onMouseMove(event) {
    basket.position.x = event.point.x;
}

function onFrame() {
    apple.position.y += 5;
}
```

Study Drills:

1. Can you modify this code to make the apple fall faster?
2. Can you modify this code the make the apple fall slower?

Exercise 7c

The following code will make the apple respawn whenever it touches the paddle

```
if (basket.hitTest(apple.position)) {
    apple.position.y = 0;
    apple.position.x = randomBetween(0,600);
}
```

Study Drills:

1. Add this code to your program in an appropriate place, so that whenever the player catches an apple the apple will respawn at the top of the screen
2. Test your program, to make sure that it plays infinitely - each time the user catches an apple, it must respawn
3. Test your program to make sure that the apple spawns in a different random location each time

Exercise 7d

This code will create a score variable, so that you can keep track of the player's score

```
score = text(40,40,0,"black")
```

This code will increase the score by one

```
score.content = parseInt(score.content) + 1;
```

Study Drills:

1. Add the score functionality to your program
2. Make it so that red and green apples fall, but when the user catches a green apple they lose 10 points
3. Make it so that each time the user catches an apple, the next apple moves slightly faster

```
bird = new bird()
pipe = new pipe()
score = 0
function onFrame() {
    bird.position.y += [redacted]
    pipe.position.x -= [redacted]
    if (pipe.position.x < [redacted]) {
        pipe.position.x = [redacted]
        pipe.position.y = randomBetween([redacted],[redacted])
        score = score + 1
    }
    if (pipe.hitTest(bird.position)) {
        displayMessage("Game Over!\n Final Score: " + score)
    }
}
function onMouseDown() {
    bird.position.y -= [redacted]
}
```

Exercise 8

This code is the start of a simple Flappy Bird game.

Study Drills:

Copy the code into Code Puppy.

Some of the values in the program have been hidden. Work out what these numbers should be to make the game play properly.

```
addTopBumper("Red")
addBottomBumper("Red")
addLeftBumper("Red")
addRightBumper("Red")
addBall("Blue")
addPaddle("Green")
addBreakoutBlocks(   ,   ,"Black")

function onFrame() {
    moveForward(ball,   )
    bounceOnWalls()
    bounceOnPaddle()
    checkBlocks()
}

function onMouseMove(event) {
    movePaddle(event)
}
```

Exercise 9

This code is the start of a simple Breakout game.

Study Drills:

Copy the code into Code Puppy.

Some of the values in the program have been hidden. Work out what these numbers should be to make the game play properly.