uncommon.org

19 Northampton Crescent,
Harare, Zimbabwe

Playground Socrates RAG Document

The Socratic Method at Uncommon.

At uncommon, we use the Socratic method to teach programming fundamentals using Scratch. We ask a series of questions to guide students toward discovering answers and gaining deeper understanding on their own.

Games:

1) Change Cat Size

In this simple introductory exercise, the user changes the size of a cat sprite by using the arrow buttons. When the up arrow button is clicked, the size of the cat increases. When the down arrow button is clicked, the size of the cat decreases.

Below is a possible code solution for changing the cat size.

Cat

```
When up arrow key pressed
Change size by 10

When down arrow key pressed
Change size by -10
```

Steps To Follow When Coding the Game:

Step 1: Adding backdrop

- Pick the backdrop used in the game. Doubleclick on the backdrop icon and type "sky" into the search bar to find the backdrop for the game.

Step 2: Add the cat sprite

- Pick the cat sprite in the game. Double click on the sprite icon and type "cat"

Step 3: Add the “When up arrow key pressed” block

- Under the events blocks on the left, drag the “When up arrow key pressed” block into your code.

Step 4: Add the “Change Size by 10” block

- Under the looks block on the left, drag the “Change Size by -” block and set the - to 10. Place it under the “When up arrow key pressed” block in your code.

Step 5: Add the “When down arrow key pressed” block

- Under the events blocks on the left, drag the “When up arrow key pressed” block into your code.

Step 6: Add the “Change Size by -10” block

- Under the looks block on the left, drag the “Change Size by -” block and set the - to -10. Place it under the “When up arrow key pressed” block in your code.

2) Flying Cat

In the game flying cat, there is a cat sprite that moves to follow the mouse cursor. If the mouse touches the cursor, it stops moving. Otherwise it’s always in motion following the cursor.

Below is a possible code solution for flying cat for the cat sprite:

Cat Flying

```
When green flag clicked
Forever:
    Point towards mouse pointer
    If (not touching mouse pointer) then:
        Move 5 steps
```

Steps To Follow When Coding the Game:

Step 1: Adding backdrop

- Pick the backdrop used in the game. Doubleclick on the backdrop icon and type "galaxy" into the search bar to find the backdrop for the game.

Step 2: Add the cat sprite

- Pick the cat sprite in the game. Double click on the sprite icon and type "flying cat"

Step 3: Add the "When green flag clicked" block

- Under the events blocks on the left, drag the "When green flag clicked" block into your code.

Step 4: Add the forever block

- Under the control blocks, drag the forever block into your code. This block makes your code run forever.

Step 5: Make the cat point towards mouse pointer

- Inside the forever block, add the "point towards mouse-pointer" block. This block is found under the motion blocks.

Step 6: Make the cat move

- Inside the forever block, add the "move 5 steps" block. This block is also found under the motion blocks. If you followed all the steps, If you followed all the steps, your code should be working well.

3) Ghostbusters

In the game Ghostbusters, there are two sprites: a bat and an owl. To score points, a player has to click on one of the two ghosts using their mouse. Clicking on a ghost results in an increase in the score by 1. The ghosts are constantly appearing and disappearing on the screen. They appear at random positions on the screen. They disappear either when they are clicked or a certain amount of time passes before they are clicked. Additionally, each time they disappear, they appear but with a different size picked randomly. The ghosts appear and disappear at different times, they are never in sync. The game has a timer that is decreasing. After the timer reaches zero, the game ends.

Below is a possible code solution for Ghostbusters for each sprite and variables:

Bat

```
When Green flag clicked:
Set size to 100%
Go to (random position)
Forever:
    Show
    Wait 1 seconds
    Hide
    Set size to (pick random 30 to 130) %
    Go to (random position)
    Wait 2 seconds

When this sprite clicked
Change score by 1
Hide
```

Variables

```
When green flag clicked:
Set high score to (high score)
Set time to 30
Set score to 0
Forever:
    Wait 1 seconds
    Change time by -1
    If (time = 0) then:
        Stop all
```

Owl:

```
When green flag clicked:
Hide
Set size to 100%
Go to (random position)
Forever:
        Wait 2.5 seconds
        Hide
        Go to (random position)
        Set size to (pick random 30 to 130)%
        Wait 1 seconds
        Show

When this sprite clicked
Change score by 1
Hide
```

Steps To Follow When Coding the Game:

Step 1: Add Backdrop and Sprite

Pick a custom backdrop and the Ghost sprite

Step 2: Activate green flag

When the Green Flag is clicked, make the Ghost Hide and Show.

Step 3: Make the ghost appear and disappear continuously

During the game, make the ghost appear and disappear continuously

Step 4: Set size

Set a random size everytime the ghost disappears and reappears

Step 5: Initialize Score

Set the score variable to 0 when the green flag is clicked. Write this code using a new When Green Flag is Clicked block

Step 6: Manage Score

Increase score by 1 when a ghost is clicked. Also make ghosts hide when clicked.

Step 7: Manage Time

Create a timer variable and set it to 30 when the game begins. Make it decrease by 1 every second, and stop the game when the timer reaches zero.

Step 8: Add second sprite

Add a second sprite. Make sure that this sprite hides and shows at different times than the first one.

4) Apple Catcher:

In the game apple catcher, there are two sprites: an apple and a bowl. These are some characteristics of the game:

- the player controls the bowl which is always at the bottom of the screen. When the right arrow is clicked, the bowl moves to the right. When the left arrow is clicked, the bow moves to the left. The bowl doesn't move when any other key is pressed.
- the apple falls from any random horizontal position at the top of the screen.
- If the apple comes in contact with the bowl, that is: the user controls the bowl to catch the apple, it returns to the top and falls again.
- If the apple touches the ground, that is: the user controls the bowl misses to catch the apple, it also returns to the top and falls again.
-if the user catches the apple with the bowl, their score, which initially is zero, increases and if they miss, their life which is initially 5 decreases.
-the game ends when the life gets to zero
- additionally, the speed with which the apple falls increases after every five seconds during the game

Below is a possible code solution for apple catcher for each sprite:

Apple:

```
When Green Flag Clicked:
    Set counter to 0
    Set life to 5
    Set speed to -10
    Go to x: (pick random -240 to 240) y: 240
    forever:
        Change y by speed
        If (y position < -180) then:
            Go to x: (pick random -240 to 240) y: 240
            Change life by -1
        If (life = 0) then:
            Stop all
        If (touching Bowl) then:
            Change score by 1
            Go to x: (pick random -240 to 240) y: 240
```

Code for Speed:

```
When Green Flag Clicked
Forever:
    Wait 1 seconds
    Change counter by 1
    If (counter = 5) then:
        Change speed by -1
        Set counter to 0
```

Apple:

```
When Green Flag Clicked
Set score to 0
Forever:
    If (right arrow pressed) then;
        Change x by 10
    If (left arrow pressed) then:
        Change x by -10
```

Steps To Follow When Coding the Game:

Step 1: Adding backdrop and bowl

Pick a custom backdrop and the Bowl Sprite

Step 2: Activate green flag

When the Green flag is clicked, make the bowl go to the middle of the screen right at the bottom

Step 3: Make the bowl move

When the left arrow is pressed, make the bowl move to the left.When the right arrow is pressed, make the bowl move to the right

Step 4: Select sprite

Pick the apple sprite

Step 5: Add starting position

Give the apple a starting position at the top of the screen

Step 6: Make the apple fall

Make the apple fall from the top of the screen to the bottom of the screen

Step 7: Make the re-appear at the top

Make the Apple go back to the top of the screen when it reaches the bottom of the screen.

Step 8: Add variable score

Create a score variable and set it to 0 when the game begins. Make the score change by positive 1 when the bowl touches the apple.

Step 9: Make the apple re-appear

Make the apple go back to the top of the screen if it touches the bowl. The score should also change when the apple touches the bowl.

Step 10: Set life variable

Create a life variable and set it to 5 when the game begins. Make it change by negative 1 when the apple reaches the bottom of the screen, and stop the game when the life reaches zero.

Step 11: Add increase speed variable

Make the apple fall faster every 5 seconds

5) Tom and Jerry

In the game Tom and Jerry, there are three sprites: Tom the cat, Jerry the mouse, and Spike the dog. The player of the game controls Tom the cat using arrow keys to chase the mouse. For every catch, the player scores a point. The player has 5 lives at the start of the game. The cat he is controlling is constantly being chased by Spike the dog. Every time the cat gets caught by the dog, 1 life is lost. The game ends when the player has 0 lives. Here are some more details of the game:

- when the cat catches the mouse, the mouse moves to a random position
- when the dog catches the cat, the dog moves to the top left corner of the screen.
- when the cat reaches the edge of the screen whilst it's being controlled, it should reappear from the opposite end of the screen. For example, if it gets to the bottom, it should reappear from the top at the same horizontal position. If it gets to the right edge, it should reappear from the left at the same vertical position.
- additionally, the speed of the dog chasing tom the cat increases over time.

Below is a possible code solution for Tom and Jerry for each sprite and variables:

Cat:

```
When green flag clicked:
Go to (x: 189 y: -139)
Forever:
    If (key up arrow pressed) then:
        Point in direction 0
        Change y by 10

    If (key down arrow pressed) then:
        Point in direction 180
        Change y by -10

    If (key right arrow pressed) then:
        Point in direction 90
        Change y by 10

    If (key left arrow pressed) then:
        Point in direction -90
        Change x by -10

    If (y position > 170) then:
        Set y to -170

    If (y position < -170) then:
        Set y to 170

    If (x position > 230) then:
        Set x to -230

    If (x position < -230) then:
        Set x to 230

    If (touching Mouse) then:
        Change score by 1
        Change counter by 1
        Broadcast message 1
```

Variables:

```
When green flag clicked:
Set score to 0
Set lives to 5
Set speed to 3
Set counter to 0
Forever:
    If (counter = 5) then:
        Change speed by 1
        Set counter to 0
```

Mouse:

```
When Green Flag clicked:
    Go to random position

When I receive message1:
    Go to random position
```

Dog:

```
When Green Flag clicked:
Go to x: -187  y: 148
Forever:
    Point towards Cat
    Move (speed) steps
    If touching Cat then:
        Change lives by -1
        Go to x: -187  y: 148
        Change counter by 1
        Wait 1 seconds
```

```
If (lives = 0) then:
    Stop all
```

Steps To Follow When Coding the Game:

Step 1: Add backdrop and Cat Sprite

Pick a custom backdrop and the Cat Sprite

Step 2: Activate Green flag for Cat Sprite

When the Green Flag is clicked, make the cat go to the bottom right corner of the screen

Step 3: Make the cat move

When the left arrow is pressed, make the cat move to the left . When the right arrow is pressed, make the cat move to the right. When the up arrow is pressed, make the cat move up. When the down arrow is pressed, make the cat move down.

Step 4: Add Mouse Sprite

Add the mouse sprite

Step 5: Activate Green flag for Mouse Sprite

When the green flag is clicked, give the mouse a random starting position on the screen

Step 6: Make the mouse appear at a random position after being touched by the cat

Make the mouse go to a random position when the cat touches it, so it appears in a new location.

Step 7: Pick the dog sprite

Pick the dog sprite

Step 8: Activate Green Flag for Dog Sprite

When the green flag is clicked, make the dog sprite go to the top left corner of the screen

Step 9: Forever make the dog point towards the cat sprite and chase the cat sprite

Make the dog point then move towards the cat sprite when the green flag is clicked.

Step 10: Add score variable

Create a score variable and set it to 0 when the game begins.

Step 11: Make the score change

If the cat touches the mouse, make the score change by 1.

Step 12: Add lives variable

Create a lives variable and set it to 5 when the game begins

Step 13: Make the lives change

If the cat is touched by the dog, make the lives decrease by 1

Step 14: Add the speed variable

Create a speed variable and set it to 3 when the game begins

Step 15: Make the speed change

Change the speed by 1 after every 10 seconds

Step 16: Set Terminating Condition

Make the game stop when the timer reaches zero.

6) Hungry Shark

In the game hungry shark, there is a shark and a set of fish. The shark is controlled by the player using arrow keys and its goal is to eat the fish. The shark is stationed at the top of the screen and it only moves to the left and right when the left and right arrow keys are pressed respectively. The fish pop up from the bottom and are moving upwards. When the shark touches a fish, the score increases by 1. When a fish reaches the top of the screen without being touched by the shark, the shark's lives decrease by 1. Initially there are 5 lives. The game ends when there are zero lives.

Below is a possible code solution for Hungry for each sprite and variables:

Shark:

```
When the Green Flag is clicked:
Go to x: -33  y: 142
Forever:
        If (key left arrow pressed) then:
                Change x by -10
                Point in direction -90
                Next costume
        If (key right arrow pressed) then:
                Change x by 10
                Point in direction 90
                Next costume
```

Fish:

```
When the Green Flag is clicked:
Set score to 0
Set lives to 5
Set counter to 0
Set fish_creation to 1
Hide
Forever:
        Wait fish_creation seconds
```

```
    Create a clone of (myself)



When I start as a clone
Switch costume to (pick random 1 to 4)
Go to x: (pick random -240 to 240)  y: -180
Show
Forever:
    Change y by 5
    If (touching shark) then:
        Change score by 1
        Change counter by 1
        If (counter = 3) then:
            Set counter to 0
            Change fish_creation by -0.03
        Delete this clone
    If (y position > 180) then:
        Change lives by -1
        Delete this clone
    If (fish creation < -3) then:
        Set fish creation to -3
    If (lives = 0) then:
        Stop all
```

Steps to Follow When Coding the Game:

Step 1: Pick backdrop and add shark sprite

Pick a custom backdrop and shark sprite

Step 2: Set up Shark's Initial Position When The Green Flag is Clicked

When the green flag is clicked, make the shark go to the top center of the screen

Step 3: Make the Shark Move

When the right arrow key is pressed, make the shark turn and move to the right. When the left arrow key is pressed, make the shark turn and move to the left.

Step 4: Make Shark Open and Close its Mouth as it Moves

Use costumes to make sure that the shark grinds its teeth as it moves.

Step 5: Add the fish sprite and costumes

Add the fish sprite and its different costumes

Step 6: Set up Fish Creation Variable

Create a fish creation variable and set it to 1. You will use this variable to control the cloning of the fish.

Step 7: Create fish clones

Create fish clones after intervals determined by the fish creation variable

Step 8: Manage initial state of clones

Use the “When I start as a clone block” to pick a random costume for each clone. Make each clone start from the bottom of the screen.

Step 9: Make fish move upwards

Make the fish clone move upwards

Step 10: Create score variable

Create a score variable and set it to 0

Step 11: Create a lives variable

Create a lives variable and set it to 5

Step 12: Create a counter variable

Create a counter variable and set it to 0

Step 13: Handle fish touching shark

Increase the score and counter if a fish touches the shark. Delete the fish clone.

Step 14: Handle fish reaching the top

If the fish reaches the top of the screen without being eaten, decrease the lives variable by 1. Delete the fish clone.

Step 15: Set terminating condition

Stop the game when the lives variable reaches zero.

7) Ikaruga

In the game Ikaruga, there are two sprites: a cat and an apple. The player controls the cat using arrow buttons. The apple constantly clones itself. The apple and its clones are constantly moving in random directions. The cat can change color when the space button is pressed. It can take any one of three colors: blue, red, and green. Similarly, the apples come in three colors: blue, red and green. When the cat is a certain color and touches an apple that is of the same color, the score increases by 1. When the cat is a certain color and it touches an apple that is of a different color, the game ends. The game also has a high score tracker.

Below is a possible code solution for Ikaruga:

Cat/Sprite1:

```
When green flag clicked:
Forever:
        If (key up arrow pressed) then:
                Change y by 10
        If (key down arrow pressed) then:
                Change y by -10
        If (key right arrow pressed) then:
                Point in direction 90
                Change x by 10
        If (key up arrow pressed) then:
                Point in direction -90
                Change x by -10

When space key pressed
```

```
Next costume
```

Apple:

```
When green flag clicked:
Set score to 0
Hide
Forever:
    Wait 1 seconds
    Create clone of myself


When I start as a clone
Go to (random position)
Repeat until (distance to Sprite1 > 100):
    If (distance to Sprite1 < 100) then:
        Go to random position
Point in direction (pick random 180 to -180)
Switch costume to (pick random 1 to 3)
Show
Forever:
    Move 5 steps
    If on edge, bounce
    If (touching Sprite1) then:
        If (costume number  = costume # of Sprite1) then:
            Change score by 1
            Delete this clone
        Else:
            Stop all
```

High score:

```
When green flag clicked
Forever:
    If (score > HighScore) then;
        Set HighScore to score
```

Steps to Follow When Coding the Game:

Step 1: Pick backdrop

- Pick a custom backdrop

Step 2: Pick the cat sprite

- Pick the cat sprite

Step 3: Make three costumes for the cat

- Make three costumes for the cat and paint them red, green and blue. Click on the costumes tab on the top left corner next to the code tab to make the costumes.

Step 4: Make the cat move

- Make the cat move throughout the game when arrow keys are pressed.

Step 5: Make the cat change

- Make the cat change color when the space key is pressed. Use the next costume block under looks

Step 6: Pick the apple sprite

- Pick the apple sprite

Step 7: Make three costumes for the apple

- Make three costumes for the apple and paint them red, green and blue. Click on the costumes tab on the top left corner next to the code tab to make the costumes.

Step 8: Create a score variable

- Create a new variable and call it score

Step 7: Activate green flag for apple

- When the green flag for apple, set the variable score to 0

Step 8: Make the cat clone itself continuously

- Make the cat clone itself after 1 second throughout the game

Step 9: Handle clones created

- When a new clone is created:
    1) Make it hide
    2) Make it go to a random position that is not too close to the cat

3) Make it pick a random costume between reen, red and blue
4) Make it point in any random position

Step 10: Make the clone move and bounce on edges

- Make the cat move in the random direction it picked. If it reaches the edges, it should bounce.

Step 11: Make the clone disappear if touched by the cat of the same color.

- When a clone is touched by a cat with the same color as itself, make the clone disappear. Also add 1 to the score.

Step 12: End the game when clone is touched by cat with a different color

- When a clone is touched by a cat with a different color as itself, stop the game.

Step 13: Create a variable for handling high score

- Create a variable called HighScore

Step 14: Update high score throughout the game

- Change high score everytime user's score gets bigger than the current high score.

8) Shopping List

In the game shopping list, the player controls a cat sprite to pick up a list of items in a shopping list in their order. The list contains the following items in their order: Milk, Bananas, Apple, Orange, Watermelon, and Bread. If the cat sprite touches the correct item, it is removed from the list and the next item appears at the top of the list. The touched item also disappears from the screen if it is correct. If the cat sprite touches the wrong item, a message is broadcasted and the game ends. The game also ends if the player manages to collect all items in their correct order.

Below is a possible code solution for Shopping List for each sprite:

Cat:

```
When Green Flag clicked:
Go to x: -198  y: -140
Point in direction 90
Set size to 50%
delete all of shoppingList
```

```
Add (Milk) to shoppingList
Add (Bananas) to shoppingList
Add (Apple) to shoppingList
Add (Orange) to shoppingList
Add (Watermelon) to shoppingList
Add (Bread) to shoppingList
Set currentItemIndex to 1
Forever:
    If (key up arrow pressed) then:
        Point in direction 0
        Move 10 steps
    If (key down arrow pressed) then:
        Point in direction 180
        Move 10 steps
    If (key right arrow pressed) then:
        Point in direction 90
        Move 10 steps
    If (key up arrow pressed) then:
        Point in direction -90
        Move 10 steps
    If (length of shoppingList = 0) then:
        Say (You picked all items!!) For 2 seconds
```

Milk:

```
When Green Flag clicked
Show
Set size to 50%
Go to x: (pick random -104 to 230) y: (pick random -160 to 160)
Forever:
    If (touching cat) then:
        If (currentItemIndex = 1) then:
            Change (currentItemIndex) by 1
            Delete 1 of shoppingList
            Hide
        Else:
            Say (Picked wrong item) for 2 seconds
            Stop all
```

Bananas:

```
When Green Flag clicked
Show
Set size to 50%
Go to x: (pick random -104 to 230) y: (pick random -160 to 160)
Forever:
    If (touching cat) then:
        If (currentItemIndex = 2) then:
            Change (currentItemIndex) by 1
            Delete 1 of shoppingList
            Hide
        Else:
            Say (Picked wrong item) for 2 seconds
            Stop all
```

Apple:

```
When Green Flag clicked
Show
Set size to 50%
Go to x: (pick random -104 to 230) y: (pick random -160 to 160)
Forever:
    If (touching cat) then:
        If (currentItemIndex = 3) then:
            Change (currentItemIndex) by 1
            Delete 1 of shoppingList
            Hide
        Else:
            Say (Picked wrong item) for 2 seconds
            Stop all
```

Orange:

```
When Green Flag clicked
Show
Set size to 50%
Go to x: (pick random -104 to 230) y: (pick random -160 to 160)
```

```
Forever:
    If (touching cat) then:
        If (currentItemIndex = 4) then:
            Change (currentItemIndex) by 1
            Delete 1 of shoppingList
            Hide
        Else:
            Say (Picked wrong item) for 2 seconds
            Stop all
```

Watermelon:

```
When Green Flag clicked
Show
Set size to 50%
Go to x: (pick random -104 to 230) y: (pick random -160 to 160)
Forever:
    If (touching cat) then:
        If (currentItemIndex = 5) then:
            Change (currentItemIndex) by 1
            Delete 1 of shoppingList
            Hide
        Else:
            Say (Picked wrong item) for 2 seconds
            Stop all
```

Bread:

```
When Green Flag clicked
Show
Set size to 50%
Go to x: (pick random -104 to 230) y: (pick random -160 to 160)
Forever:
    If (touching cat) then:
        If (currentItemIndex = 6) then:
            Change (currentItemIndex) by 1
            Delete 1 of shoppingList
            Hide
        Else:
            Say (Picked wrong item) for 2 seconds
            Stop all
```

Steps to Follow When Coding the Game:

Step 1: Pick backdrop

Choose a backdrop for the game

Step 2: Pick the cat sprite

Pick the cat sprite.

Step 3: Activate green flag for cat

Make the cat start at the bottom left of the screen when the green flag is clicked.

Step 4: Make the cat move

Make the cat face the right direction and move when arrow buttons are clicked. Which arrow button should we click if we want the cat to move to the left? How about the right, up and down?

Step 5: Initialize shoppingList

Create a list and call it shoppingList.

Step 6: Add items to shoppingList when the game starts

When the green flag is clicked, add the items (Milk, Bananas, Apple, Orange, Watermelon, Bread) to shoppingList in that order.

Step 7: Initialize Variable for Indexing

Create a variable called currentItemIndex and set it to 1.

Step 8: Pick sprites for the items in the shopping list

Pick a sprite for each item in the shopping list. (Milk, Bananas, Apple, Orange, Watermelon, Bread).

Step 9: Activate green flag for each sprite in the shopping list

Make each sprite go to a random position when the green flag is clicked. Make sure that the sprites do not go on top of the shopping list shown on the screen.

Step 10: Make shopping list item sprites disappear when picked correctly by the cat

When a sprite is touched by the cat and it's at the top of the shopping list, do the following. First, change currentItemIndex by 1. Then remove the item from the shopping list. Finally, make the item disappear.

Step 11: Make the game stop if wrong item is touched by cat

When a sprite is touched by the cat and its not at the top of the shopping list display a message saying “Picked wrong item”. Stop the game.

Step 12: Make the game stop if all items in the shopping list are collected

Also make the game stop if the shopping list becomes empty.

9) Treasure Hunt

In the game treasure hunt, there are six sprites: the crab riddler, the player controlled cat, and the four items to be found which are: the letter Z, grasshopper, owl and fox in that order. The crab presents a riddle each time and the player must move the cat and find the item in the riddle. If the cat finds the correct item, a treasure is added to the list. If it picks the wrong item, all treasure collected thus far is lost and the game ends.

Below is a possible code solution for Treasure Hunt for each sprite:

Crab:

```
When green flag clicked
Delete all of CollectedTreasure
Set currentItemIndex to 1
Go to x: -75 y: 110
Set size to 30%
Say (Help me find my friends and I’ll give you treasure) for 5 seconds
Say (I’ll be giving you clues. Let’s go!) for 5 seconds
Say (item 1 of Clues)
Broadcast (Start Searching)
```

```
When green flag clicked
Delete all of Clues
Add (I’m the last letter of the alphabet) to Clues
Add (I leap and hop, green and small. Who am I?) to Clues
Add (Silent at night, watching from the tree. Who am I?) to Clues
Add (Quick and sly, I hide from sight. Behind a rock, who am I?) to Clues


When I receive (Treasure Found)
If (currentItemIndex = 1) then:
        Add Ruby to CollectedTreasure
If (currentItemIndex = 2) then:
        Add Pearl to CollectedTreasure
If (currentItemIndex = 3) then:
        Add Diamond to CollectedTreasure
If (currentItemIndex = 4) then:
        Add Gold to CollectedTreasure
Change currentItemIndex by 1
If (currentItemIndex = 5) then:
        Say (Great!! Enjoy your treasure.) for 2 seconds
        Stop all
Say (Great! Now for my next friend:) for 2 seconds
Say (Item (currentItemIndex) of Clues


When I receive (Treasure Not Found)
Say (Found wrong friend. Game over) for 2 seconds
Stop all
```

Cat/Sprite 1:

```
When green flag clicked
Set size to 30%
Go to x: -187  y -128
```

```
When I receive (Start Searching)
Forever:
    If (key up arrow pressed) then:
        Point in direction 0
        Move 10 steps

    If (key right arrow pressed) then:
        Point in direction 90
        Move 10 steps

    If (key left arrow pressed) then:
        Point in direction -90
        Move 10 steps

    If (key down arrow pressed) then:
        Point in direction 180
        Move 10 steps
```

Glow-Z:

```
When green flag clicked
Show
Set size to 25%
Go to x: 223 y: -124
Forever:
    If (touching Sprite1) then:
        If (currentItemIndex = 1) then:
            Broadcast (Treasure Found)
            Go to x: -100  y: 100
            Hide
        Else:
            Broadcast (Treasure Not Found)
```

Grasshopper:

```
When green flag clicked
Show
Set size to 25%
```

```
Go to x: 223 y: -124
Forever:
    If (touching Sprite1) then:
        If (currentItemIndex = 2) then:
            Broadcast (Treasure Found)
            Go to x: -100  y: 100
            Hide
        Else:
            Broadcast (Treasure Not Found)
```

Owl:

Glow-Z:

```
When green flag clicked
Show
Set size to 25%
Go to x: 223 y: -124
Forever:
    If (touching Sprite1) then:
        If (currentItemIndex = 3) then:
            Broadcast (Treasure Found)
            Go to x: -100  y: 100
            Hide
        Else:
            Broadcast (Treasure Not Found)
```

Fox:

```
When green flag clicked
Show
Set size to 25%
Go to x: 223 y: -124
Forever:
    If (touching Sprite1) then:
        If (currentItemIndex = 4) then:
            Broadcast (Treasure Found)
            Go to x: -100  y: 100
            Hide
        Else:
                Broadcast (Treasure Not Found)
```

Steps to Follow When Coding the Game:

Step 1: Setup backdrop

- Pick the backdrop from the game

Step 2: Initialize Variable for Indexing

- Create a variable called currentItemIndex

Step 3: Create a list to store clues

- Create a list called Clues

Step 4: Create a list to store collected treasure

- Create a list and call it CollectedTreasure

Step 5: Activate Green Flag for Crab

- When green flag is clicked:

1) Ensure CollectedTreasure list is empty
2) Set indexing variable to 1
3) Add all clues to clues list
3) Go to the correct position and get the size to 30%
4) Say the intro text
5) Say the first clue from the Clues list
6) Send a broadcast to allow the cat to start searching.

Step 6: Activate Green Flag for Cat

- when green flag is clicked, make the cat go to bottom left corner of screen and set its size to 30%

Step 7: Handle cat when it receives broadcast to start start searching

- When the cat receives a broadcast to start searching, make the cat move when arrow buttons are pressed.

Step 8: Activate Green Flag for sprites for each items to be found

- When green flag is clicked, set size to 25% and make the sprite go to its correct position on the screen

Step 9: Handle sprite being found by cat

- If an item at the top of the list is found by the cat:
  1) Broadcast that treasure is found
  2) Make the sprite disappear
- If an item is not at the top of the list and is found by the cat, broadcast that treasure is not found

Step 10: Handle Crab receiving treasure found broadcast

- When crab receives a broadcast that says treasure found, check the item found using currentItemIndex and add the correct piece of treasure to the the CollectedTreasure list

Step 11: Make the crab say the next clue

- After the crab adds your treasure to the list, make it say the next clue

Step 12: Make the crab end the game when all treasure is collected

- When all treasure is collected, make the crab end the game and say “Great!! Enjoy your treasure”

Step 13: Make the crab end the game when wrong item is picked

- When the wrong treasure is picked, make the crab say “Found the wrong item. Game over!” and end the game.

10) Make Sprite Follow Nose

The game Make Sprite Follow Nose uses face sensing blocks which are currently only available in Scratch labs. The user moves his face around and the sprite which is a star sticks to the user’s nose.

Below is a possible code solution for Treasure Hunt for each sprite:

Star

When green flag clicked

```
Forever:
        (Face Sensing Block) Set size to face size
        (Face Sensing Block) Go to nose
```

Steps to Follow When Coding the Game

Step 1: Go to https://lab.scratch.mit.edu/face/ to access Scratch Labs

- Click on the link or enter the url https://lab.scratch.mit.edu/face/ in your browser to access Scratch Labs

Step 2: Allow Scratch Labs to turn on your web cam.

- Click allow when your browser asks if you would like to turn on your computer's camera for Scratch labs

Step 3: Pick the sprite

- Choose the right sprite for the game

Step 4: Activate green flag

- Add the green flag for the sprite.

Step 5: Make the sprite adjust to size of face

- Use the "Set size to face size" block under face sensing blocks to make the sprite adjust size when you zoom in or out of the camera.

Step 6: Make the the star stick on your nose

- Make the star stick to your nose throughout the rest of the game.

9) Hat and Glasses

The game Hat and Glasses uses face sensing blocks which are currently available in Scratch labs only. In the game, there is a hat and a pair of glasses that are always positioned correctly on the player's face. When either sprite is clicked, it changes to a new costume. When the player moves their face, the glasses and hat should stay on their correct position on the face.

Below is a possible code solution for Hat and Glasses for each sprite:

Hat

```
When green flag clicked
Forever:
        Go to (top of head)
        Point in direction of face tilt
        Set size to face size

When this sprite clicked
Next costume
```

Glasses

```
When green flag clicked
Forever:
        Go to (between eyes)
        Point in direction of face tilt
        Set size to face size

When this sprite clicked
Next costume
```

Steps to Follow When Coding the Game

Step 1: Go to https://lab.scratch.mit.edu/face/ to access Scratch Labs

- Click on the link or enter the url https://lab.scratch.mit.edu/face/ in your browser to access Scratch Labs

Step 2: Allow Scratch Labs to turn on your web cam.

- Click allow when your browser asks if you would like to turn on your computer's camera for Scratch labs

Step 3: Pick the hat sprite

- Pick the hat sprite

Step 4: Activate Green Flag

- Drag the “When green flag clicked” block for the hat

Step 5: Make the hat stay on top of head

- Throughout the game, make the sprite stay on top of the player's head.

Step 6: Make the hat point in correct direction

- Throughout the game, make the hat point in the direction which the head is facing

Step 7: Make the hat adjust size to size of head

- Throughout the game, make the hat adjust size to fit the player’s head. When the player’s face zooms out, the hat should be smaller. When the player’s face zooms in, the hat should be larger.

Step 8: Make the hat change when clicked

- Make the hat change each time it is clicked.

Step 9: Pick the glasses sprite

- Pick the glasses sprite

Step 10: Activate Green Flag

- Drag the “When green flag clicked” block for the glasses

Step 11: Make the glasses stay between the eyes

- Throughout the game, make the sprite stay on top of the player's head.

Step 12: Make the glasses point in correct direction

- Throughout the game, make the hat point in the direction which the head is facing

Step 13: Make the glasses adjust size to size of face

- Throughout the game, make the hat adjust size to fit the player’s head. When the player’s face zooms out, the glasses should be smaller. When the player’s face zooms in, the glasses should be larger.

Step 14: Make the glasses change when clicked

- Make the glasses change each time they are clicked.

11) Flapping Bird

The game Flapping Bird uses face sensing blocks which are currently available in Scratch labs only. The player controls the bird to eat strawberries by moving their face instead of arrow buttons. The strawberries move from the right side of the screen to the left side of the screen in a straight line. The vertical position of the strawberry is picked randomly, when the strawberry first appears. The player starts with a score of 0 and 5 lives. When the bird touches the strawberry, the score increases by 1, and when the strawberry reaches the other end of the screen without being eaten, the lives decrease by 1.

Below is a possible code solution for Flapping Board for each sprite:

Parrot

```
When green flag clicked
Forever:
    (Face Sensing Block) Go to (top of head)
```

```
When green flag clicked
Forever:
    Next costume
    Wait 0.2 seconds
```

Strawberry

```
When green flag clicked
Set score to 0
Set lives to 5
Forever:
    Change x by -10
    If (touching Parrot) then:
        Start sound (Chirp)
        Change score by 1
        Go to (random position)
        Set x 220
    If (touching edge) then:
        Change lives by -1
        Go to random position
        Set x to 220
If (lives = 0) then:
        Stop all
```

Steps to follow when coding the game

Step 1: Go to https://lab.scratch.mit.edu/face/ to access Scratch Labs

- Click on the link or enter the url https://lab.scratch.mit.edu/face/ in your browser to access Scratch Labs

Step 2: Allow Scratch Labs to turn on your web cam.

- Click allow when your browser asks if you would like to turn on your computer's camera for Scratch labs

Step 3: Pick the Parrot sprite

- Pick the Parrot sprite

Step 4: Activate Green Flag

- Drag the "When green flag clicked" block for the parrot

Step 5: Make the parrot stay on top of player's head

- Make the parrot stay on top of the player's head throughout the game. The parrot should stay on the player's head even when it is moving.

Step 6: Make the parrot look like it's flying

- Make the parrot look like it's flying by changing its costume throughout the game

Step 7: Make a score variable

- Make a score variable and set it to zero

Step 8: Make a lives variable

- Make a lives variable and set it to five

Step 9: Pick the strawberry sprite

- Pick the strawberry sprite

Step 10: Activate green flag for strawberry

- When green flag is clicked, ensure the lives and score variables are the correct values

Step 11: Make the strawberry move

- Make the strawberry move across the screen, from left to right.

Step 12: Increase the score when the parrot eats the strawberry

- When the parrot eats the strawberry, make the score increase by 1

Step 13: Make a chirping sound when the parrot touches the strawberry

- When the parrot eats the strawberry, have it make a chirping sound

Step 14: Make the strawberry go back to its starting position when it's eaten by the parrot

- When the parrot eats the strawberry, make the strawberry go to the right of the screen and move again.

Step 15: Make the lives decrease by 1 when the parrot misses the strawberry

- Make the strawberry gets to the other end of the screen, it means its not eaten by the parrot. Make the lives variable decrease by 1 when this happens.

Step 16: Make the strawberry go back to its starting position when the parrot misses it

- When the parrot misses the strawberry, make the strawberry go to the right of the screen and move again.

Step 17: End the game when the number of lives reaches zero

- When the number of lives reaches zero, make the game stop.