Scratch Experience Class

Yeu-Perng Nieh 2024/03/21

Introduction

- Understand the Scratch game design process
  - Game storyline
  - Main characters
  - Stage settings
  - Game flow design
  - Programming logic development
  - Optimization, debugging, testing, and publishing

Story Synopsis

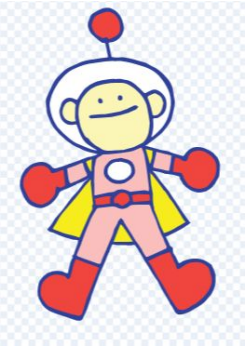
- Title: "Lost in Space: Ripley's Odyssey"
- Story Synopsis:

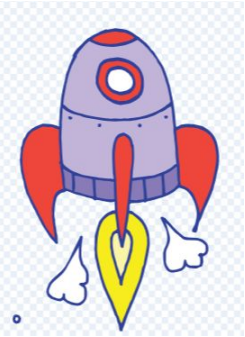
In the vast expanse of the future universe, astronaut Ripley is tasked with a mission to explore a distant planet. However, during the mission, Ripley's spacecraft loses contact with the main ship, leaving Ripley stranded in unknown space and facing daunting challenges.

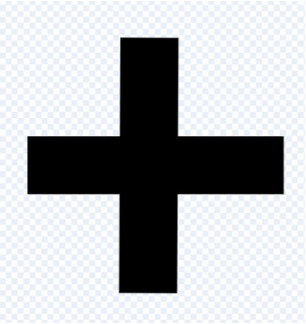
Throughout this adventurous journey, **Ripley must overcome various obstacles to find a way back to the spacecraft.** Utilizing wit and skills, Ripley navigates through unforeseen dangers, solves intricate problems, and seeks pathways that lead back to safety.

Main Characters(Sprites)

- Ripley, the Astronaut
- Rocketship
- Obstacles of various shapes

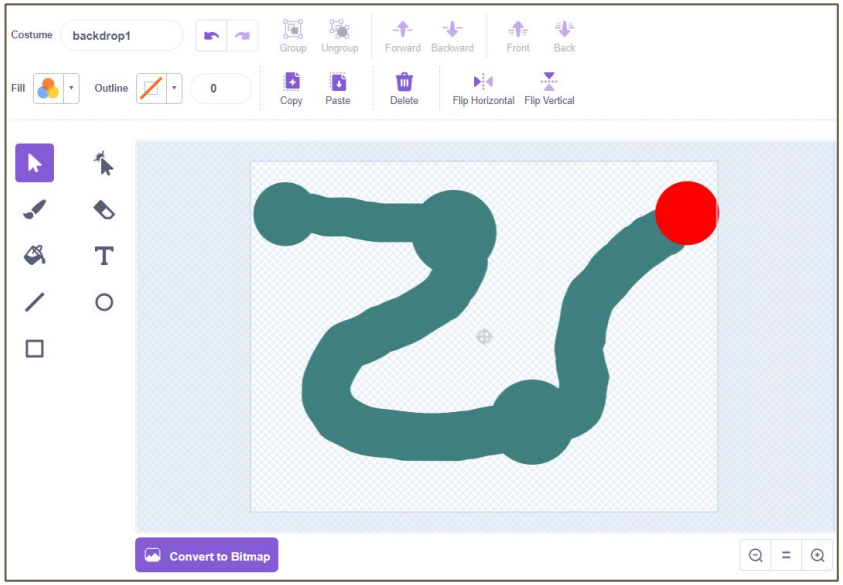






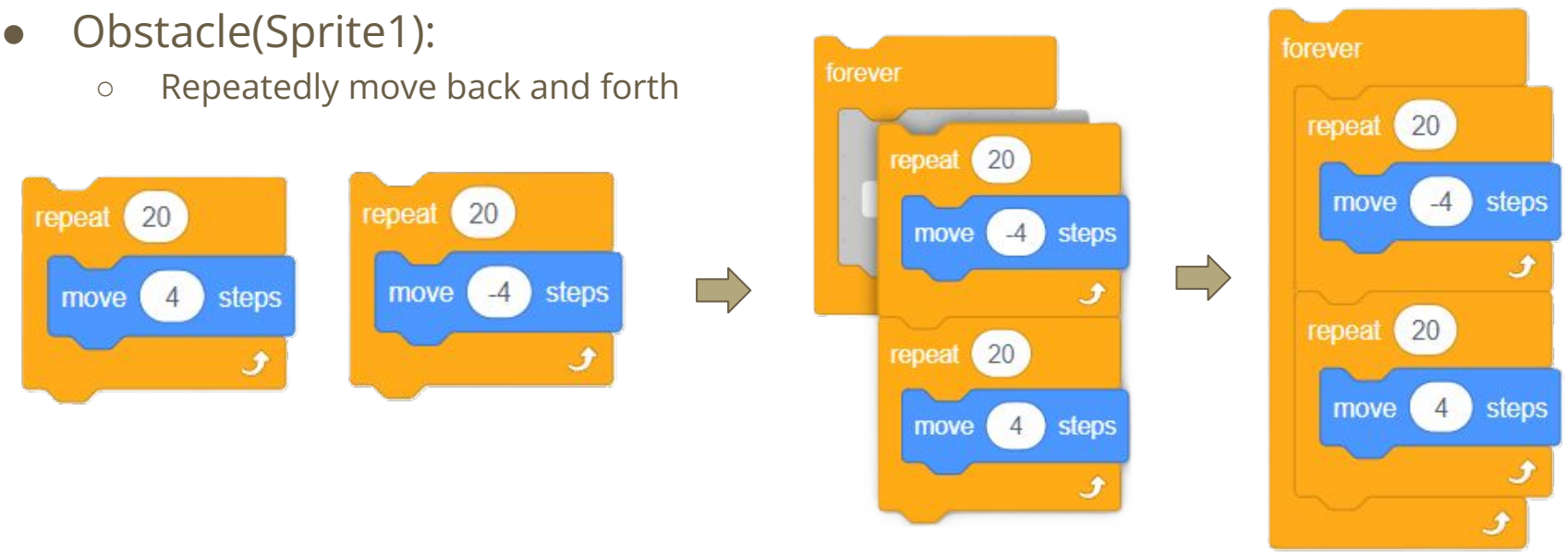
Stage design

- Design a path, for example starting from the top left corner of the stage, winding through various obstacles, and finally reaching the top right corner of the stage.
    - Pay attention to the color codes for both the path and the endpoint
- Ripley will start from the starting point (top left corner of the stage), move along the path, dodge various obstacles, and finally reach the location of the rocket spaceship (top right corner of the stage).



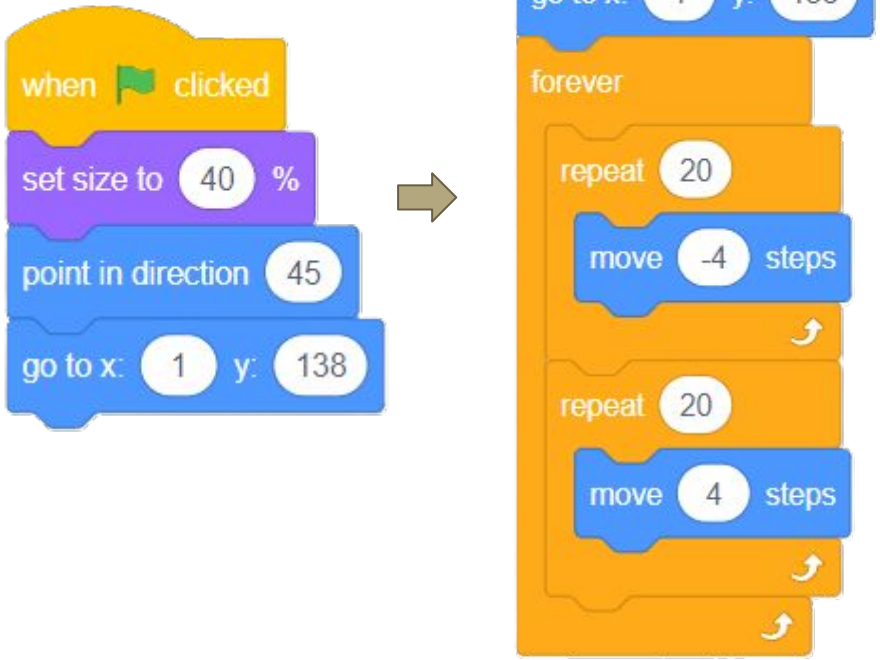
Step 2 : Flow design and logic development

- Obstacle(Sprite1):
  - Repeatedly move back and forth



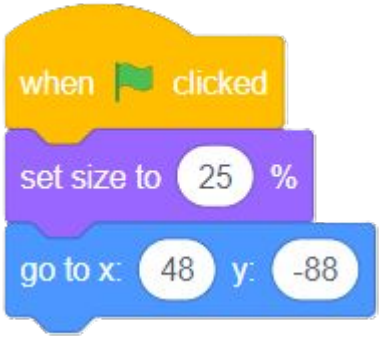
Step 2 : Flow design and logic development

- Obstacle(Sprite1):
  - Adjust size appropriately
  - Adjust orientation
  - Place in the correct position



Step 2 : Flow design and logic development

- Obstacle(Sprite2):
  - Repeatedly rotate
  - Adjust size appropriately
  - Place in the correct position

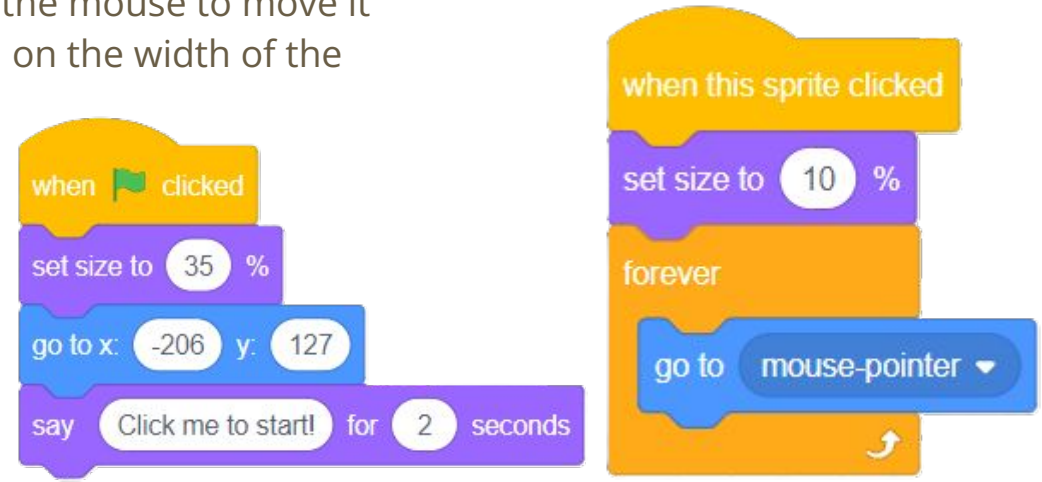


Step 2 : Flow design and logic development

- Astronaut Ripley:
  - How to move Ripley with mouse?

Step 2 : Flow design and logic development

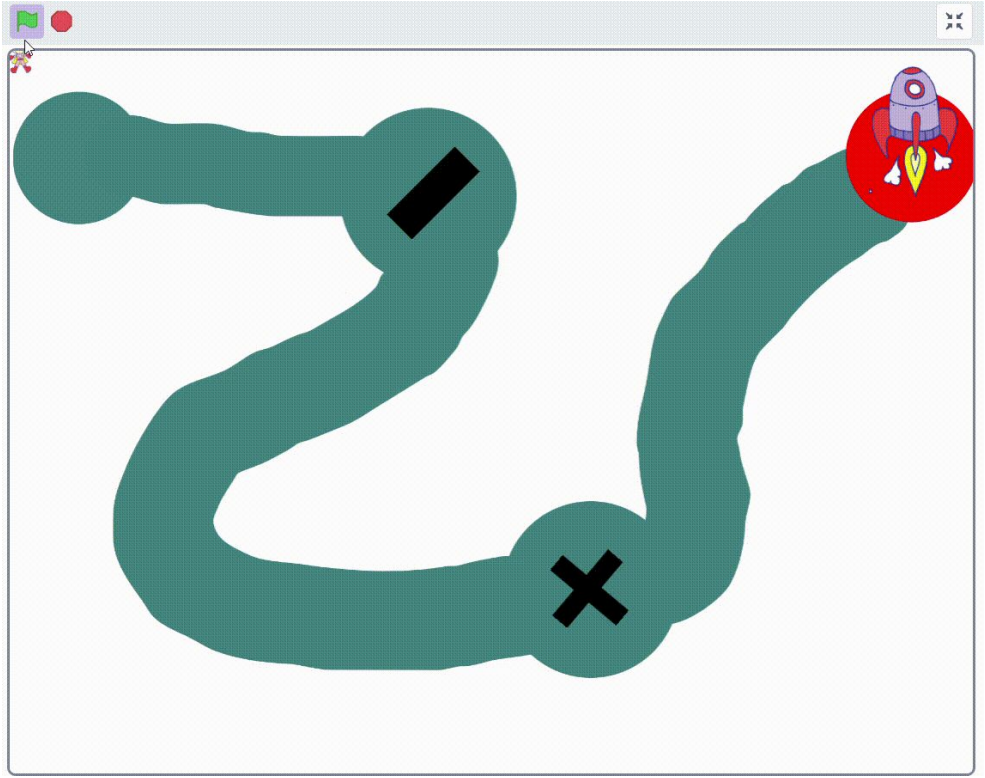
- Astronaut Ripley:
  - Increase size slightly for easier mouse clicking
  - Starting position is at the top left corner of the path
  - Display dialogue message 'Click me to start!'
  - After clicking the character, use the mouse to move it
  - Adjust size appropriately (based on the width of the path and the size of obstacles)



Step 2 : Flow design and logic development

- Rocketship:
  - This part is omitted. Please refer to the shared project code.

What the game looks like so far

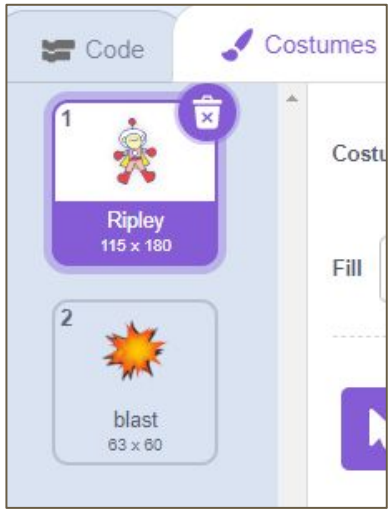


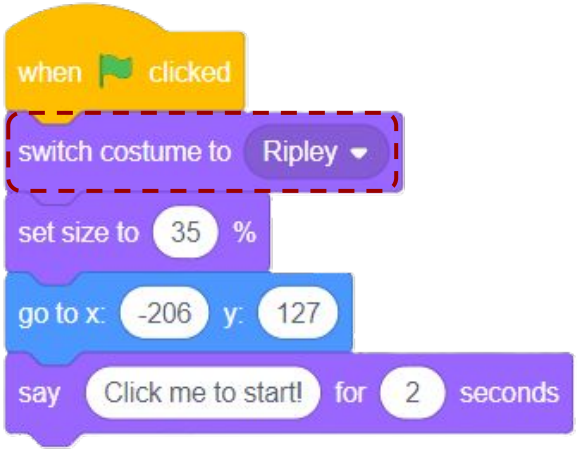
Step 3 : Game flow design

- Astronaut Ripley:
  - While moving Astronaut Ripley, if it collides with an obstacle or the edge of the path, an explosion will occur, and Ripley must return to the starting point to restart
    - Add a blast costume
    - When an explosion occurs:
      - ➢ Switch to the blast costume
      - ➢ Play the "Crunch" sound
      - ➢ Play the explosion animation (changing its size)
      - ➢ Return to the starting point to restart
  - If successfully reaching the endpoint:
    - Play the "Cheer" sound
    - Terminate the game

Step 3: Programming logic development

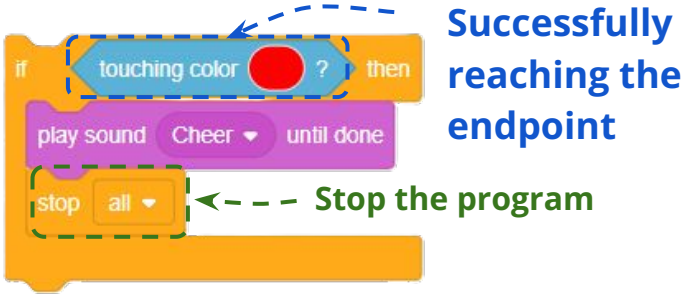
- Astronaut Ripley:
  - Add the second costume (blast), so we have two costumes ready(Ripley, blast)
  - When game starts, set the costume to Ripley

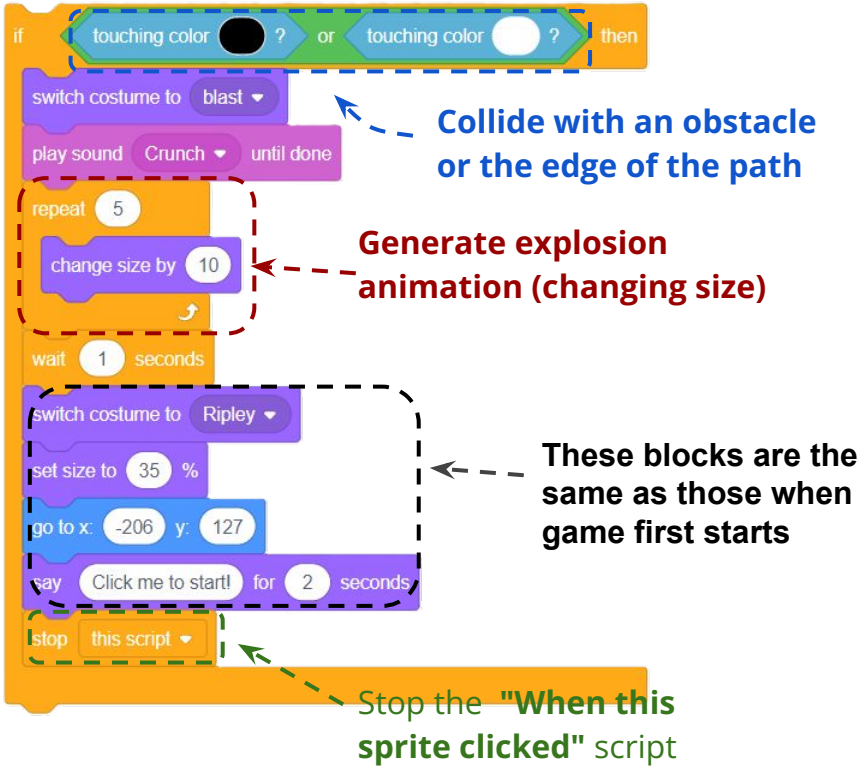




Step 3: Programming logic development

- If it collides with an obstacle or the edge of the path, an explosion will occur.
  - Switch the sprite to the blast costume
  - Play the "Crunch" sound
  - Generate explosion animation (changing size)
  - Return to the starting point to restart
  - The script for "When this sprite clicked" must be terminated
- If successfully reaching the endpoint:
  - Play the "Cheer" sound
  - Terminate the game.

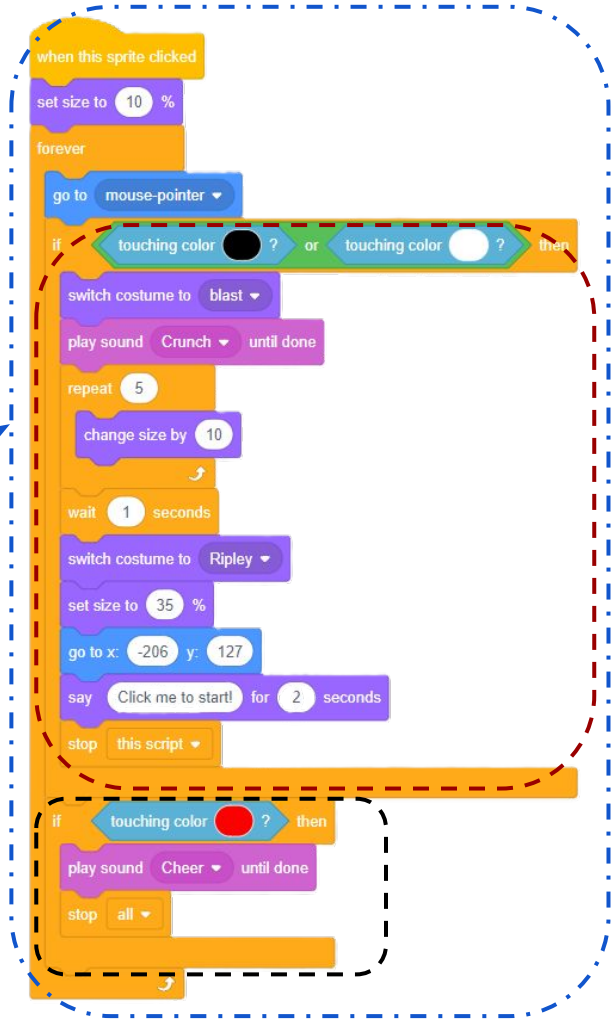




Step 3: Programming logic development

- The logic for colliding with obstacles, the edge of the path or reaching the endpoint successfully must be placed within the "when this sprite clicked" block.
    - Ensure it is placed within the "forever" block.

After the sprite is clicked, it can start moving, and then detect whether it collides with obstacles, the edge of the path, or successfully reaches the endpoint.



Let's further optimize our game design!

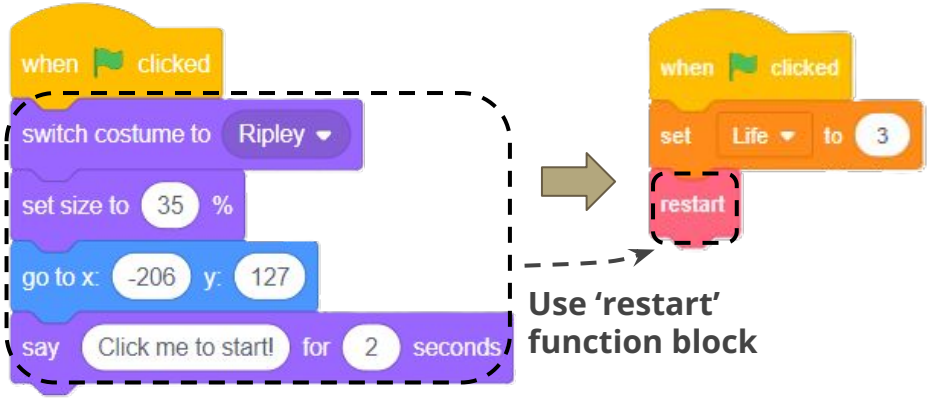
Variable

- Think of a variable as **a box for storing information**, with a **label** to remember what's inside. When you create a variable, give it a sensible name, such as "High Score" or "Player Name." You can put all sorts of data into variables, including numbers and words, and the data can change while the program is running

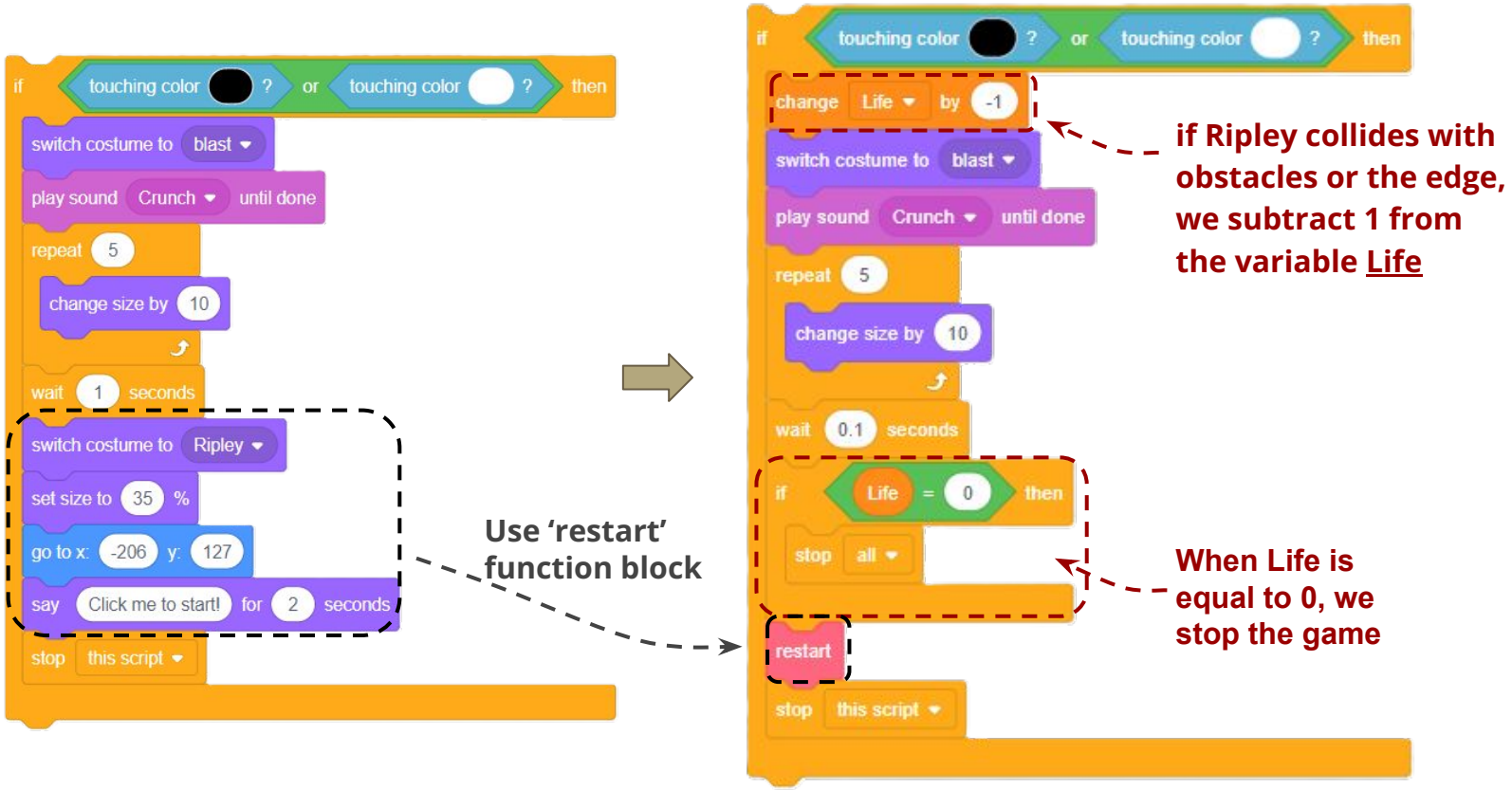
Optimization 1

- Define the initial actions for Astronaut Ripley at the start of the game using a **function block (restart)**. This allows for better management of recurring program logic.

- To make the game more challenging, stop the game if Ripley collides with obstacles or the edge of the path three times
  - Add a **variable** called "**Life**" to represent the number of lives, with an initial value of 3.

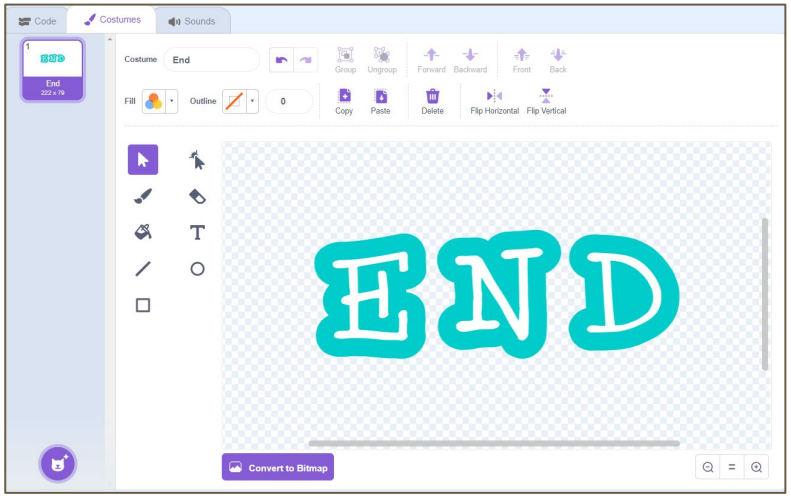


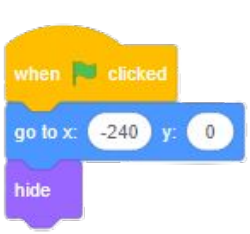
Optimization 1

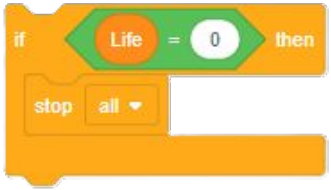


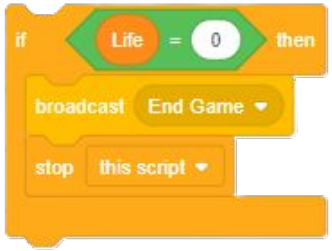
Optimization 2

- Add a game fail animation
  - Let the text "END" appear on the stage and play the "Lose" sound
  - Use the **broadcast message** **"End Game"**









Optimization 2

- Add a game success animation
  - When Ripley successfully reaches the endpoint, display the text "Winner" on the stage and play the "Cheer" sound
  - Use the broadcast message "Winner"