Open Design and Technology

Done by Ruhika Uppalapati and Shreyas More

Project Batman

Summative Assessment 1

Who is Batman going to fight today?

The last villain standing wins.

(int(255*brightness), int(80*brightness), int(80*brightness)),

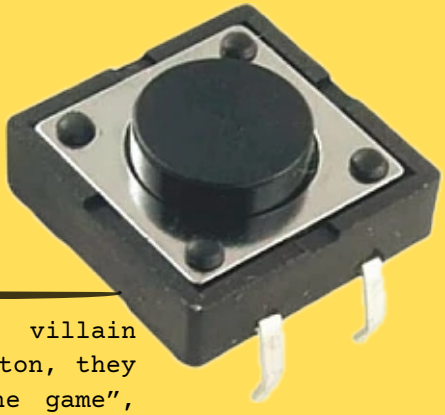
(int(80*brightness), int(160*brightness), int(255*brightness)),

(int(80*brightness), int(255*brightness), int(160*brightness)),

(int(255*brightness), int(200*brightness), int(80*brightness))

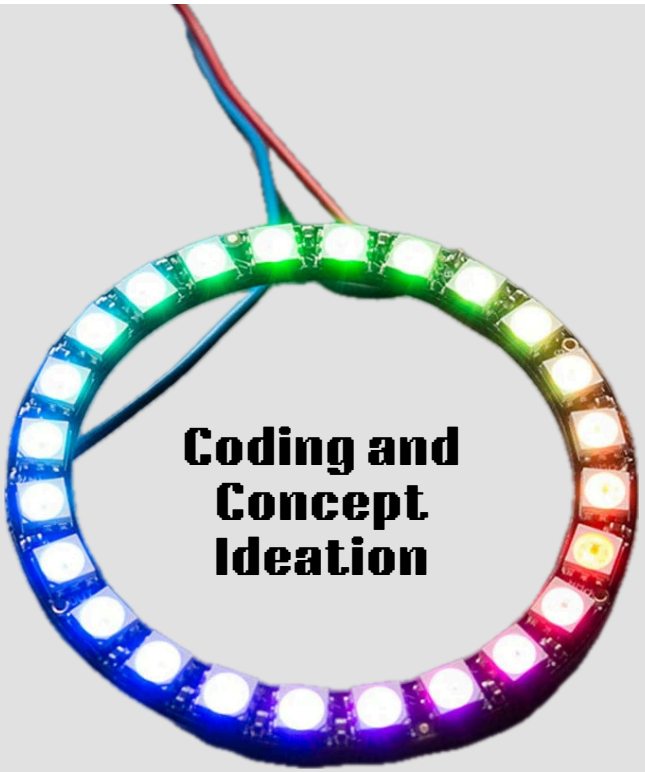
WHAT LED TO THIS IDEA...tracing **what inspired us...**

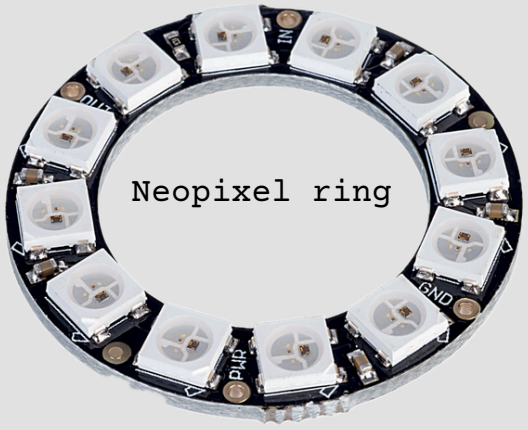
The Russian Roulette. Wanting to quarter up the **neopixel ring** and work with randomisation of the **LEDs** lighting up.



-After the chosen villain presses the push button, they are “executed in the game”, signified by the server motor’s twitch.

Push and Die!





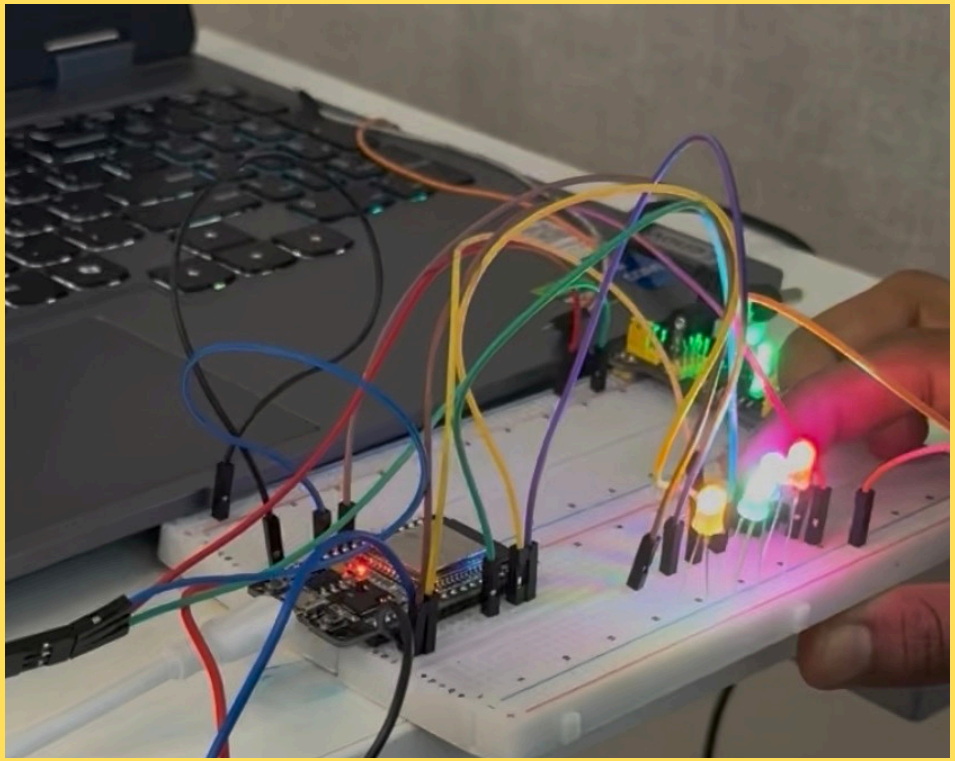

```
from machine import Pin
import neopixel
import random
import time

NUM_PIXELS = 16
np = neopixel.NeoPixel(Pin(4), NUM_PIXELS)
alive_players = [0, 1, 2, 3]
brightness=0.05
player_colors = [

    (int(255*brightness), int(80*brightness),
int(80*brightness)),

    (int(80*brightness), int(160*brightness),
int(255*brightness)),

    (int(80*brightness), int(255*brightness),
int(160*brightness)),

    (int(255*brightness), int(200*brightness),
int(80*brightness))

]

servo_pin = Pin(19)
servo = PWM(servo_pin, freq=50)
servo_positions=[30,60,100,130]



segment_size = NUM_PIXELS // len(alive_players)

spin_cycles = random.randint(20, 40)

for i in range(spin_cycles):

    selected_player = alive_players[i %
len(alive_players)]

    for j in range(NUM_PIXELS):
        np[j] = (0, 0, 0)
```

We wanted our interactive game to inolve 4 players.

We chose to include **“Int”** which stands for **Integer.** Since the brightness is set at **“0.05”**, when it gets multiplied and specified for the colour of certain LEDs, we must avoid a decimal value.

spin_cycles=random.randint(20, 40) regulates the **randomisation of the LED’s spin** and where it comes to a halt. We have chosen a range between **20-40.**

“I”: Random number selected from 20-40.



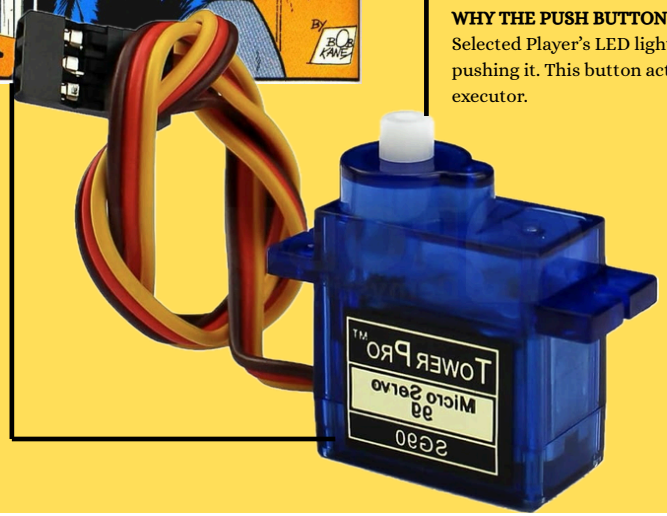
Why we chose to use Specific Components


```
print("Trigger pressed!")
  if button.value() == 0:
     angle = servo_positions[selected_player]
     duty = int((angle / 180)  102 + 26)
     servo.duty(duty)
     print("Servo pointing at Player", selected_player)
     time.sleep(1)
```

GUNPOINT

Pointer.

WHY THE PUSH BUTTON?
Selected Player's LED light goes off upon pushing it. This button acts as the executor.

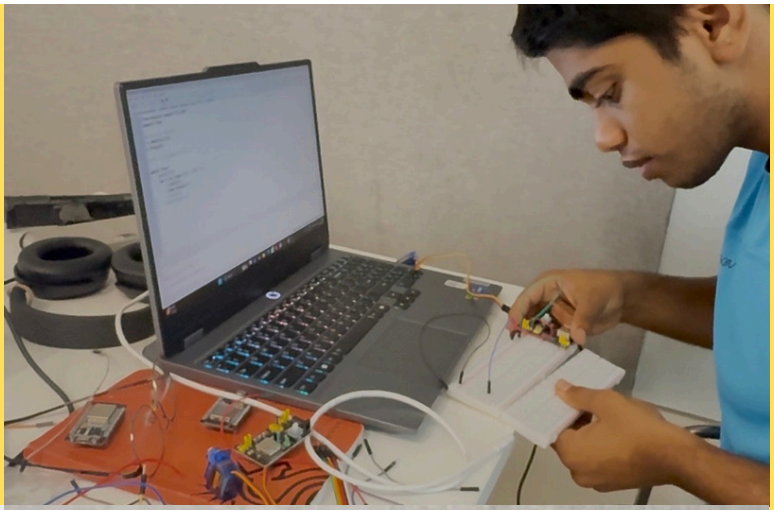


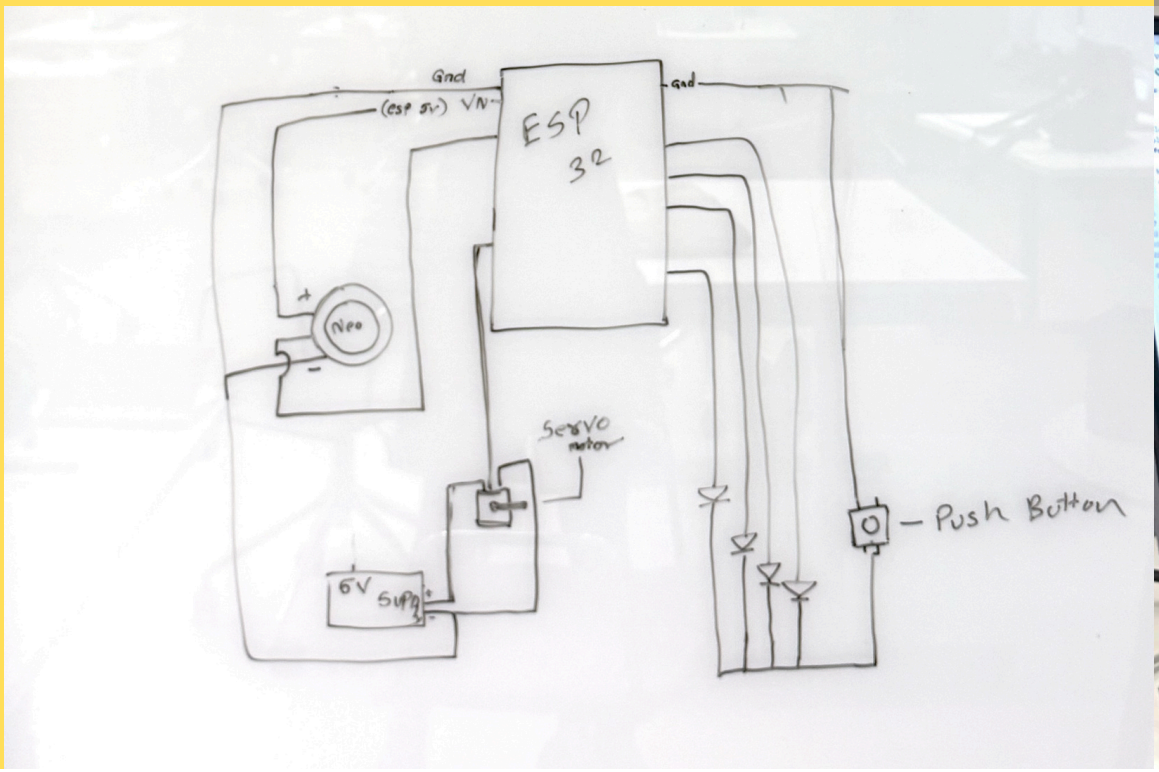
We CHOSE to use the LEDs--blue, red, green and yellow to represent the spark as the **villain's life,** and the **absence** of the LED lighting up as the **villain's death.**

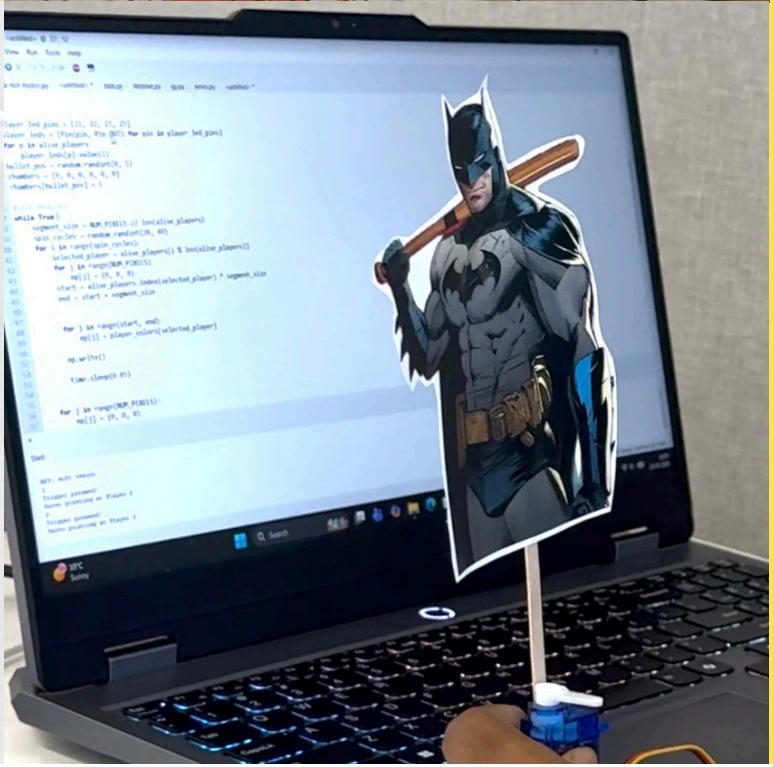
We CHOSE to use the **Servo Motor** to represent **Batman's GUN.** The steering of it and pointing towards the villains after the code has been executed indicates the death of a player in the game.

Purpose of the Neopixel ring--Rnadomisation, acting like the Russian Roulette.

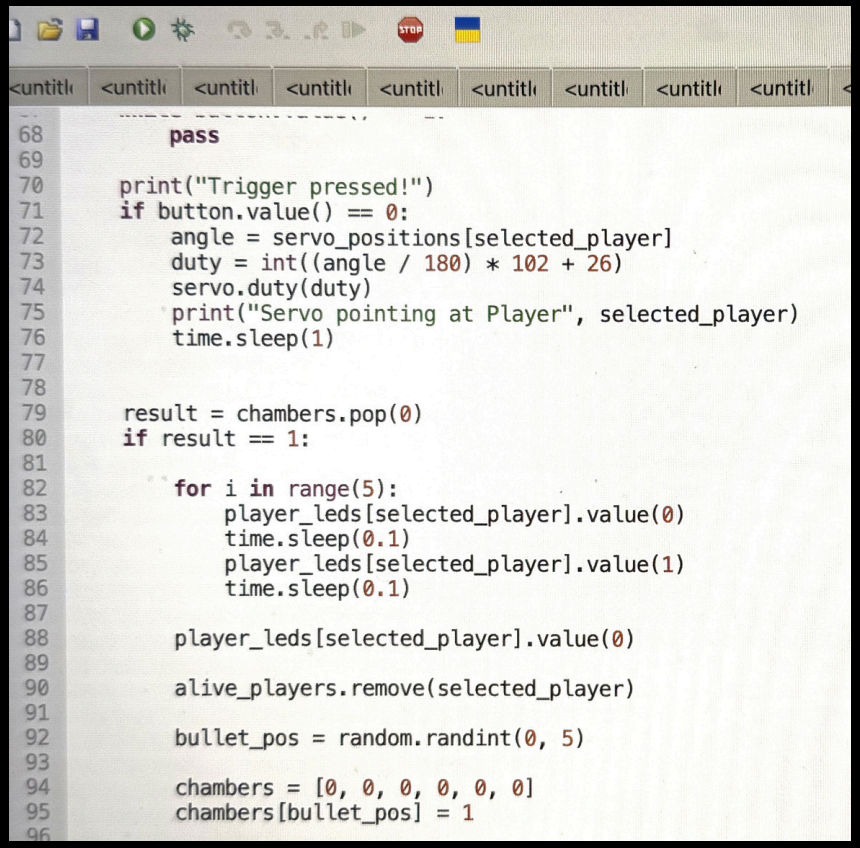
Circuit Diagram , Coding **Process** and figuring out how to incorporate the server motor with the interactive game.







Coding Logic



```
68          pass
69
70      print("Trigger pressed!")
71      if button.value() == 0:
72          angle = servo_positions[selected_player]
73          duty = int((angle / 180) * 102 + 26)
74          servo.duty(duty)
75          print("Servo pointing at Player", selected_player)
76          time.sleep(1)
77
78
79      result = chambers.pop(0)
80      if result == 1:
81
82          for i in range(5):
83              player_leds[selected_player].value(0)
84              time.sleep(0.1)
85              player_leds[selected_player].value(1)
86              time.sleep(0.1)
87
88          player_leds[selected_player].value(0)
89
90          alive_players.remove(selected_player)
91
92          bullet_pos = random.randint(0, 5)
93
94          chambers = [0, 0, 0, 0, 0, 0]
95          chambers[bullet_pos] = 1
```

`alive_players = [0, 1, 2, 3]`

- Important Variable: Decides who is still in the game, who can be selected, who can lose.

When someone dies:

`alive_players.remove(selected_player)`

Spinner skips them. Servo never points to them. They are OUT permanently.

Pain Points

Server Motor wasn't responding to the code initially. (THE CODE SEQUENCE WAS INCORRECT).

-The **Servo Motor** twitched before the **push button interaction.**

-We struggled with button input not responding consistently.

Learnings

We were able to combine **randomisation** with coding in a way that was interactive (through light, motion, and user input).

Brightness scaling using multiplication prevents overpowering LEDs.

random.randint() helped simulate unpredictability in both spin and bullet placement.

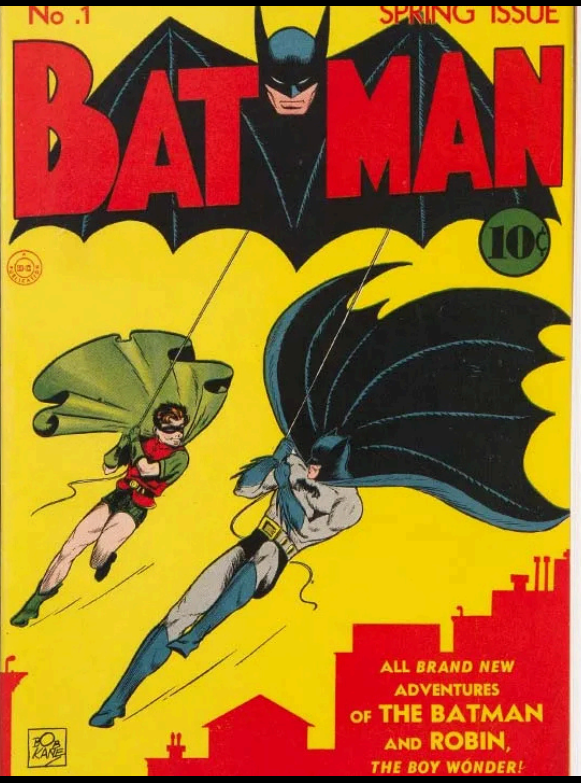
Individual Contribution

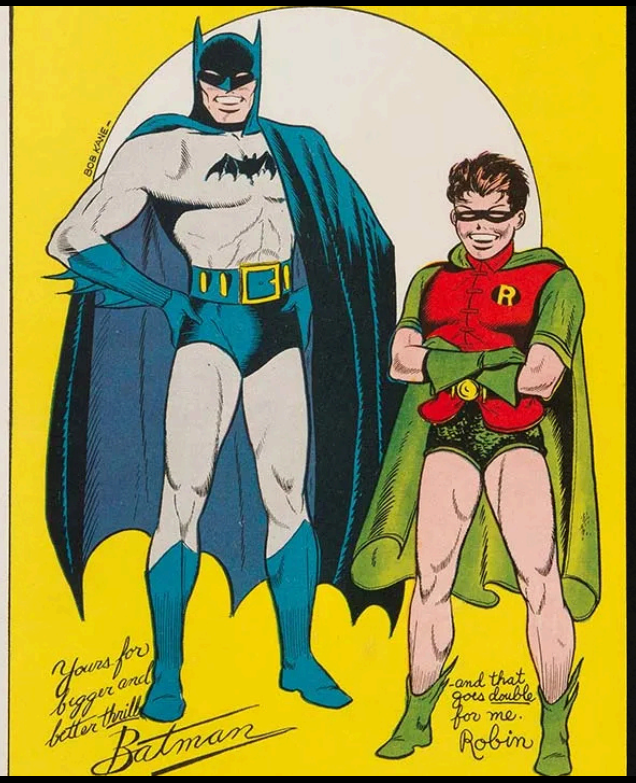
Shreyas More

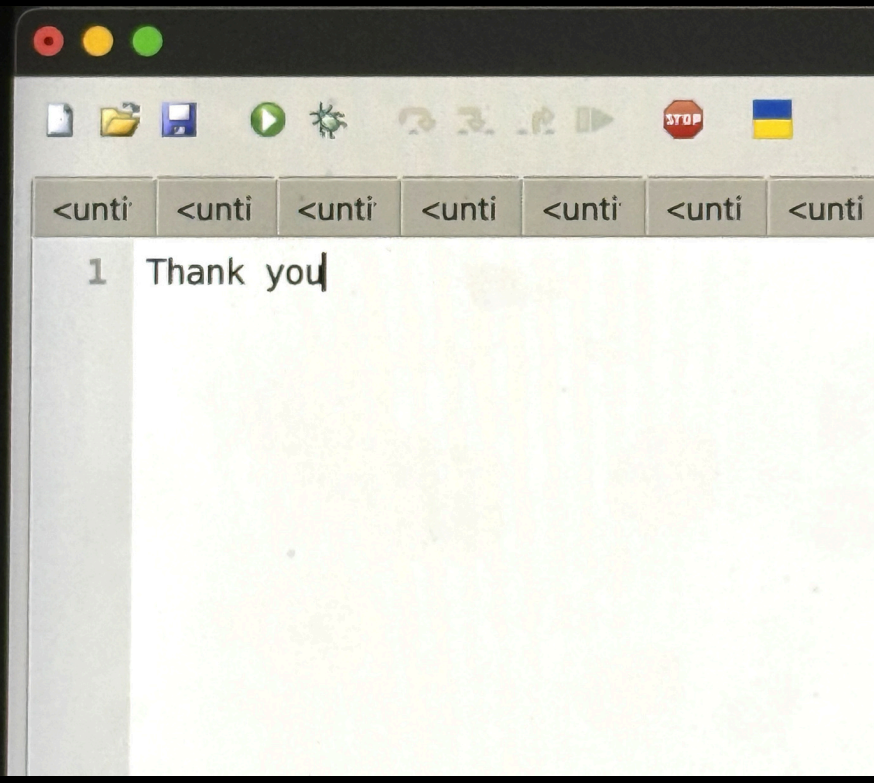
Python Coding (Serveo Motor), conencting the LEDs and Neopixel.
Managing Circuit Layout and hardware interafcing on the breadboard.
Working with the Push button.
Component selection and mechanics behind the game design.

Ruhika Uppalapati

Python coding for the Neopixel (spin cycles, alive players, etc).
Checking errors after executing the codes.
Ideation and Circuit Layout Planning.
Video, aesthetics.


No .1
SPRING ISSUE
BATMAN
10¢
ALL BRAND NEW
ADVENTURES
OF THE BATMAN
AND ROBIN,
THE BOY WONDER!


BOB KANE
Yours for
bigger and
better thrills
Batman
...and that
goes double
for me.
Robin


STOP
<unti
<unti
<unti
<unti
<unti
<unti
<unti
1
Thank you