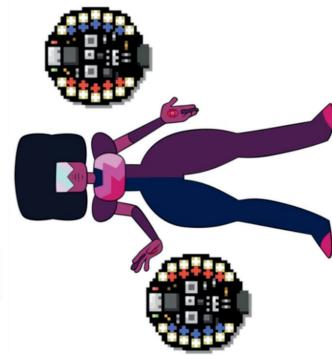


THAT TRANSFORMS COLOR.
PLAYGROUND EXPRESS GEM
FINAL CODE FOR SOLO CIRCUIT



"fused" boolean = False
If "B" button pressed, sparkly
animation starts and turns LEDs
to pink, play bleepy tune.

Fused: PINK LEDS



CN



adafruit
Playground



CN



CN

WHO & WHAT?

Adafruit Circuit
Playground
Express

3 x AAA Battery
Holder with
On/Off Switch

USB cable:
USB A to Micro-B

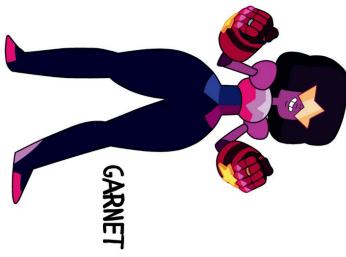


Alkaline AAA
batteries

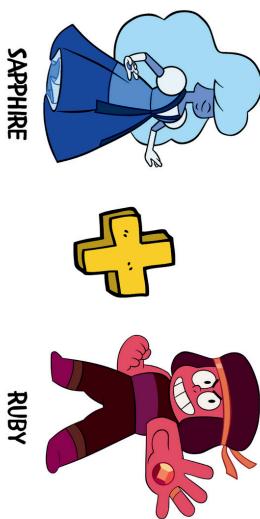
1



MORE INFO



GARNET



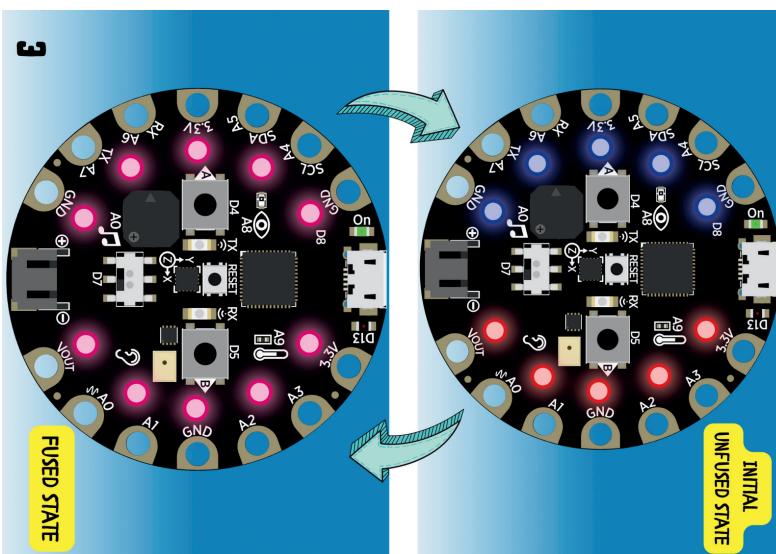
SAPPHIRE



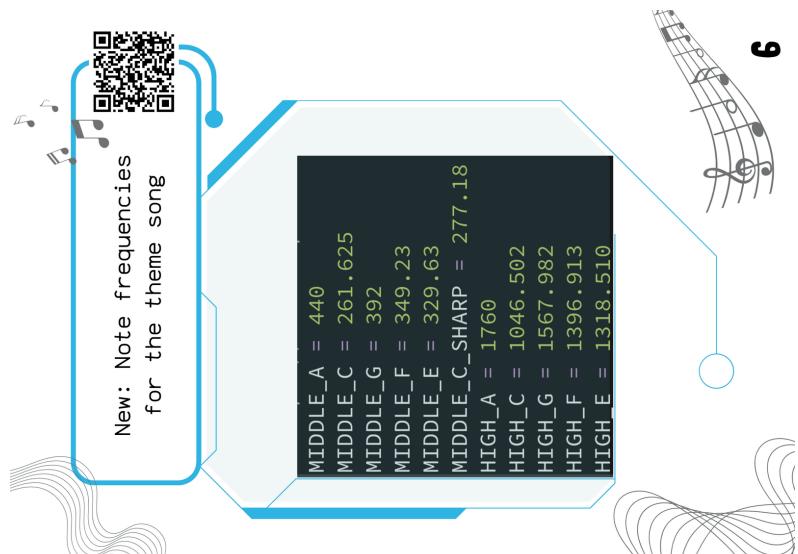
RUBY



2



MAKECODE



Default :
UNFUSED - RED and BLUE LEDs

Play theme song.
"fused" boolean = True

If "A" button pressed, animation
starts and turns LEDs to pink,
play bleepy tune.

"fused" boolean = False



5



INITIALLY TO SEE HOW IT ALL WORKED,
FOLLOWED THE **MAKECODE** INSTRUCTIONS.

NEXT STEP: PORT TO **circuitpython**



CIRCUITYTHON IS A PROGRAMMING
LANGUAGE DESIGNED TO SIMPLIFY

EXPERIMENTING AND LEARNING TO CODE ON
LOW-COST MICROCONTROLLER BOARDS.

4

```

from random import randrange
from adafruit_circuitplayground.express import cpx
import time

cpx.pixels.brightness = 0.1
fused = False # Gems are not fused, so this is set to False

RED = (255, 0, 0)
BLUE = (0, 0, 255)
SPARKLE_PINK = (241, 178, 220)
PINK = (255, 0, 255)
LEDS_OFF = (0, 0, 0)
WHITE = (255, 255, 255)

def animation_fuse():
    # Animation before it fused to form Garnet
    for an_led in range(9, -1, -1):
        time.sleep(0.01)
        cpx.pixels[an_led] = BLUE
        cpx.pixels[an_led] = SPARKLE_PINK
        time.sleep(0.01)
        cpx.pixels[an_led] = BLUE

def animation_unfuse():
    # Turn all leds off, then have a few white leds on and off
    for an_led in range(9, -1, -1):
        cpx.pixels[an_led] = LEDS_OFF
    time.sleep(0.5)
    for i in range(0, 10):
        sparkle()

def sparkle():
    # Sparkling turning on and off random LEDs
    i = randrange(9)
    cpx.pixels[i] = WHITE
    cpx.pixels[i] = LEDS_OFF
    time.sleep(0.2)

def change_colour(colour):
    for an_led in range(9, -1, -1):
        cpx.pixels[an_led] = colour
    time.sleep(0.01)

# Note frequencies
# https://en.wikipedia.org/wiki/Piano_key_frequencies
MIDDLE_A = 440
MIDDLE_C = 261.625
MIDDLE_G = 392
MIDDLE_F = 349.23
MIDDLE_E = 329.63
MIDDLE_C_SHARP = 277.18
HIGH_A = 1760
HIGH_C = 1046.502
HIGH_G = 1567.982
HIGH_F = 1396.913
HIGH_E = 1318.510

def theme_song():
    # Play Steven Universe Song
    cpx.play_tone(MIDDLE_C, 1)
    cpx.play_tone(MIDDLE_A, 0.25)
    cpx.play_tone(MIDDLE_G, 0.125)
    cpx.play_tone(MIDDLE_F, 0.25)
    cpx.play_tone(MIDDLE_G, 0.25)
    time.sleep(0.125)
    cpx.play_tone(MIDDLE_E, 0.5)
    cpx.play_tone(MIDDLE_E, 0.25)
    cpx.play_tone(MIDDLE_C_SHARP, 0.25)
    cpx.play_tone(MIDDLE_A, 0.25)
    cpx.play_tone(MIDDLE_G, 0.25)
    cpx.play_tone(MIDDLE_F, 0.25)
    cpx.play_tone(MIDDLE_E, 0.125)
    cpx.play_tone(MIDDLE_F, 0.25)
    cpx.play_tone(MIDDLE_F, 0.125)
    time.sleep(0.25)
    cpx.play_tone(MIDDLE_C, 0.25)
    time.sleep(0.125)
    cpx.play_tone(MIDDLE_A, 0.25)
    cpx.play_tone(MIDDLE_G, 0.25)
    cpx.play_tone(MIDDLE_F, 0.125)
    cpx.play_tone(MIDDLE_E, 0.25)
    cpx.play_tone(MIDDLE_F, 0.125)
    time.sleep(0.25)
    cpx.play_tone(MIDDLE_C, 0.25)
    time.sleep(0.125)
    cpx.play_tone(MIDDLE_A, 0.25)
    cpx.play_tone(MIDDLE_G, 0.25)
    cpx.play_tone(MIDDLE_F, 0.25)
    cpx.play_tone(MIDDLE_E, 0.125)
    cpx.play_tone(MIDDLE_F, 0.25)
    cpx.play_tone(MIDDLE_F, 0.125)
    time.sleep(0.25)
    cpx.play_tone(MIDDLE_C, 0.25)
    cpx.play_tone(HIGH_G, 0.0625)
    cpx.play_tone(HIGH_G, 0.0625)
    cpx.play_tone(HIGH_A, 0.0625)
    cpx.play_tone(HIGH_C, 0.0625)
    cpx.play_tone(HIGH_G, 0.0625)
    cpx.play_tone(HIGH_A, 0.0625)
    cpx.play_tone(HIGH_C, 0.0625)
    cpx.play_tone(HIGH_G, 0.0625)
    cpx.play_tone(HIGH_A, 0.0625)
    cpx.stop_tone()
    return

def unfuse_bleep_tune():
    # Play when unfusing gems
    cpx.play_tone(HIGH_C, 0.0625)
    cpx.play_tone(HIGH_G, 0.0625)
    cpx.play_tone(HIGH_A, 0.0625)
    cpx.stop_tone()
    return

```

```

while True:
    if fused:
        cpx.pixels[9] = PINK

    # Unfuse ruby and sapphire when B button is pressed
    if cpx.button_b and fused == True:
        animation_unfuse()
        unfuse_bleep_tune()

    fused = False
else:
    # Fuse to form Garnet when A button is pressed
    if cpx.button_a:
        cpx.pixels.fill((0, 0, 0))
        animation_fuse()
        animation_fuse()
        animation_fuse()

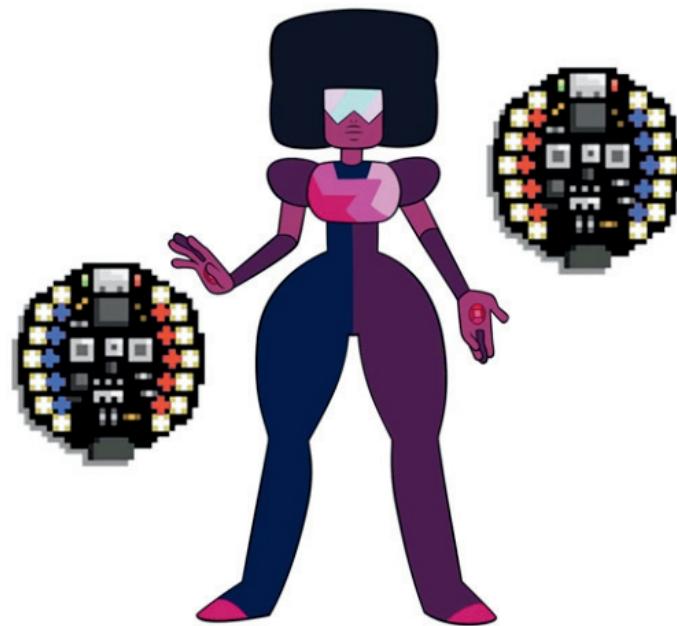
    # Setting the colour sequence in the animation
    # when fusing
    colour_seq = [PINK,
                  SPARKLE_PINK,
                  PINK,
                  SPARKLE_PINK,
                  PINK]

    for c in colour_seq:
        change_colour(c)
    fused = True

    # Now it's fused, play the theme song
    theme_song()

else:
    # Back to Ruby and Sapphire
    for blue_led in range(0, 5):
        cpx.pixels[blue_led] = BLUE
    for red_led in range(5, 10):
        cpx.pixels[red_led] = RED

```



**FINAL CODE FOR SOLO CIRCUIT
PLAYGROUND EXPRESS GEM
THAT TRANSFORMS COLORS.**