MONOCLICKER PROPOSAL

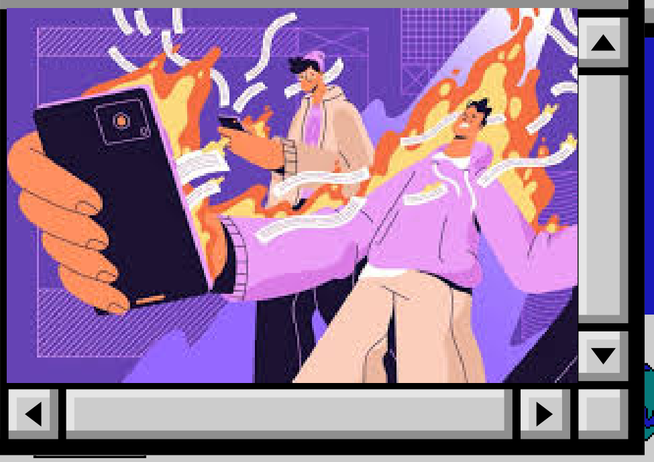
Incremental game

11:11PM

PROBLEM STATEMENT

- Most people are bored or get bored easily, but often turn to solutions that leave a negative impact on them (like doomscrolling). (Mental Health Foundation, 2025)

No



OBJECTIVES

- Create a clean simulation experience for players to enjoy without too many unnecessary features.
- Destroy boredom and be a time-killer that isn't too addictive.

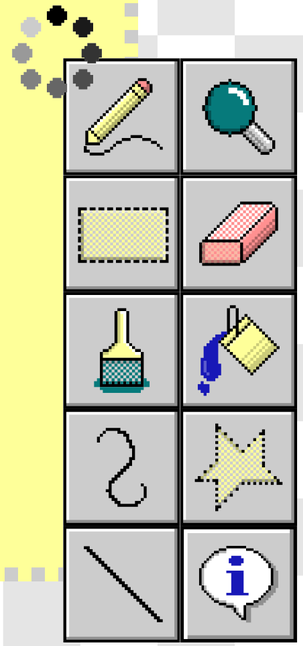
PLANNED FEATURES

– Clickable button to increase a number (the user's G)

– Generators that the user can purchase with G to increase G output and can be accessed via the "Shop" button.

– Slight chance for generators to break down, and requires a "minigame" using a consistent stream of user input in a limited amount of time to "fix" the generator

– Tips at the bottom of the main menu to push along progress and give the user hints as to what to do

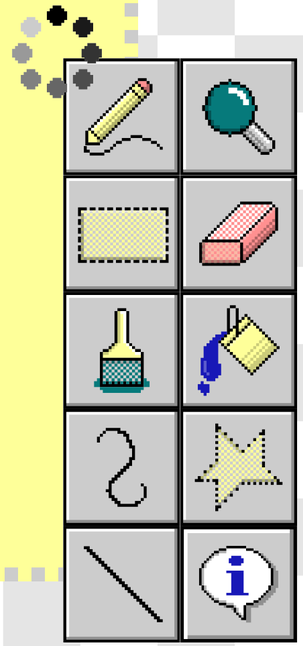


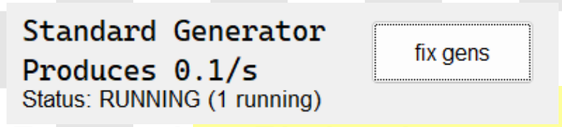
PLANNED FEATURES

– A timer that is put at the top of the screen that takes roughly 25% of the user's current G to slow down progress.

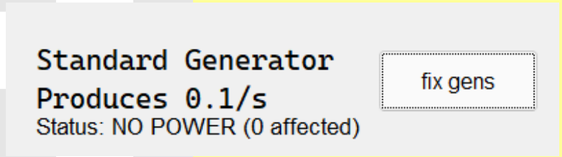
– An automatic save system that uses JSON files to store data, such as generator amounts, inflated prices, and the user's G.

– A menu used to upgrade and fix generators, accessed via the "Generator Status" button.

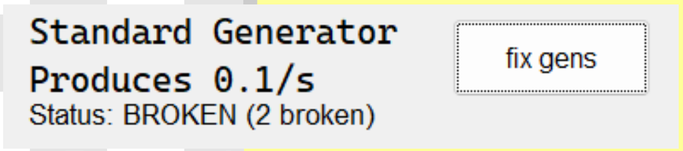




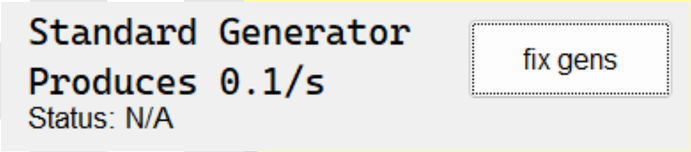
Generator Status in menu
(Running)



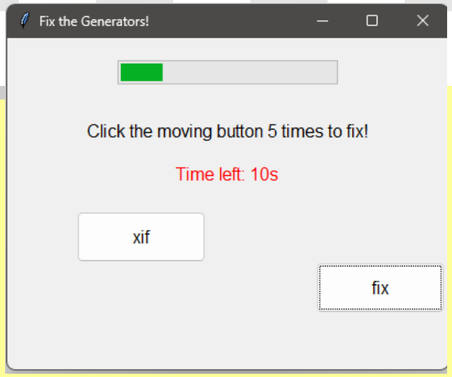
Generator Status in menu
(No electricity)



Generator Status in menu
(Broken)

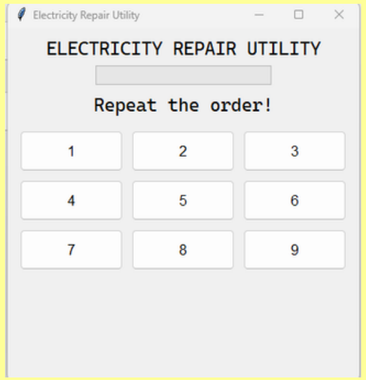


Generator Status in menu
(Not bought yet)

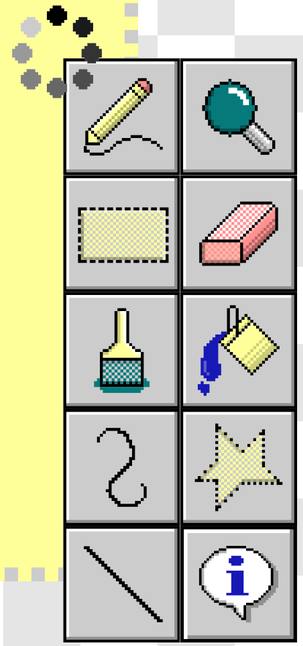


Example Generator Fix Minigame

Electricity status in menu
(No electricity)



Electricity Fix minigame
(Memory Game)



Close Shop

SHOP

0.00G

Standard G Generator

Standard G generator (1g/s) amount: 1 +1 -1
Current amount: 0 Price: 10G Stock: 15
BUY

Massive G Generator

Massive G generator (10g/s) amount: 1 +1 -1
Current amount: 0 Price: 100G Stock: 3
BUY

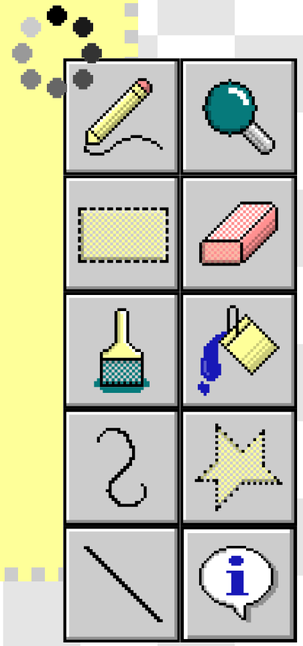
Industrial G Generator

Industrial G generator (100g/s) amount: 1 +1 -1
Current amount: 0 Price: 250G Stock: 0
BUY

Randomized G Generator

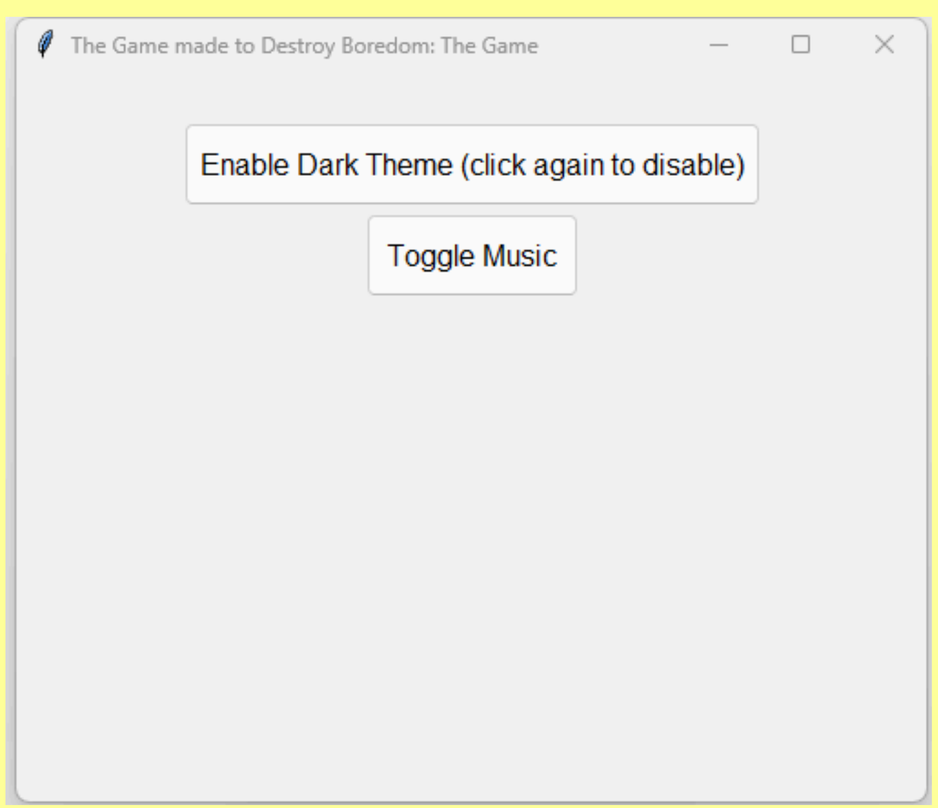
Randomized G Generator (1-100G/s) amount: 1 +1 -1
Current amount: 0 Price: 77.7G Stock: 4
BUY

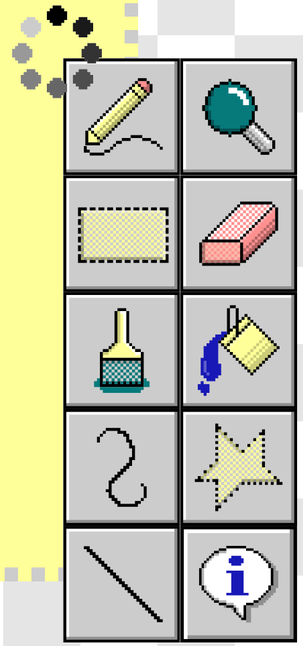
Restock (100G)



Shop Menu

Upgrade Menu





Settings Menu

TURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATU

View Log

The Game Made to Destroy Boredom: The Game

TIME LEFT BEFORE TAXES

04:48

1.00G

0.00G/s

Increase G (1)

Shop | Gen Status & Upgrades | Settings

Quit Game

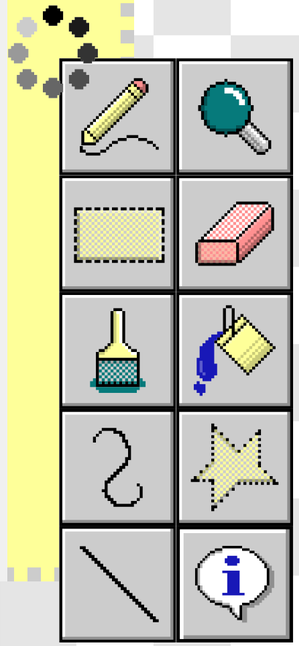
Level: 1

1/100

Pro tip: This game takes a while to complete.

Main Menu

Save | Load



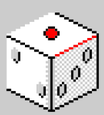
RES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES • FEATURES

PLANNED INPUTS AND OUTPUTS

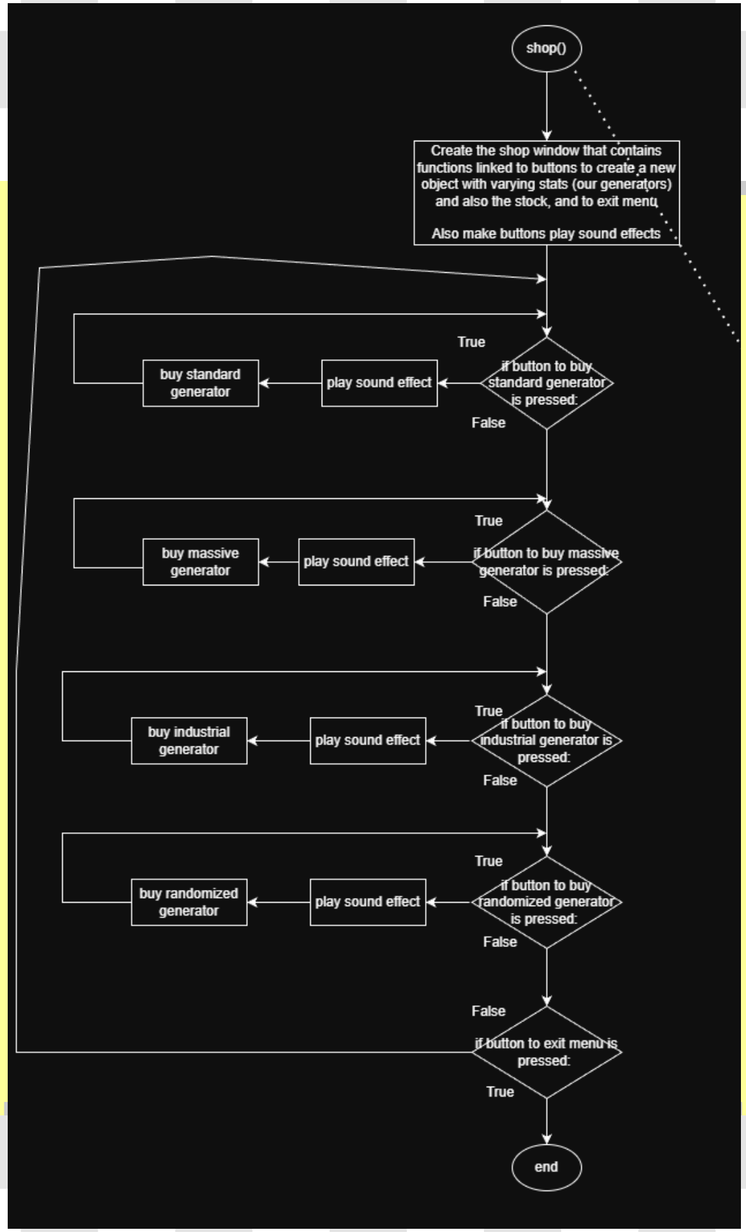
THIS IS A BRIEF GENERALIZATION OF ALL THE INPUTS AND OUTPUTS.

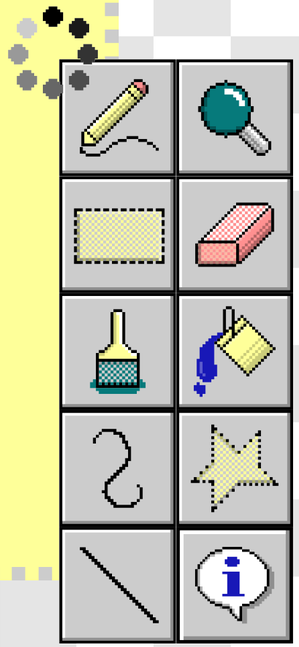
- User opens program and loads into menu
- Menu contains buttons that include a button to earn in-game currency in "g", a button to go to the shop in order to buy generators, and a status button that opens to check on the status of their generators and electricity and upgrade generators, as well as a timer at the top to indicate the time left before a percentage of their g is taken away forcefully.
- Generators are essentially objects that increase the user's "g" by an amount.
- Sometimes, generators will receive a tag that makes it stop producing g, forcing the user to "fix" generators by playing a minigame in the status menu.
- Electricity is a mechanic that disables ALL generators when it receives a tag to make it False.
- When reactivating electricity, the user has to a play another minigame, similar to fixing generators, but they have to play a memory game to restore it.
- This all happens simultaneously every "tick", 1/20th of a second that the game takes to perform every single action in that time.



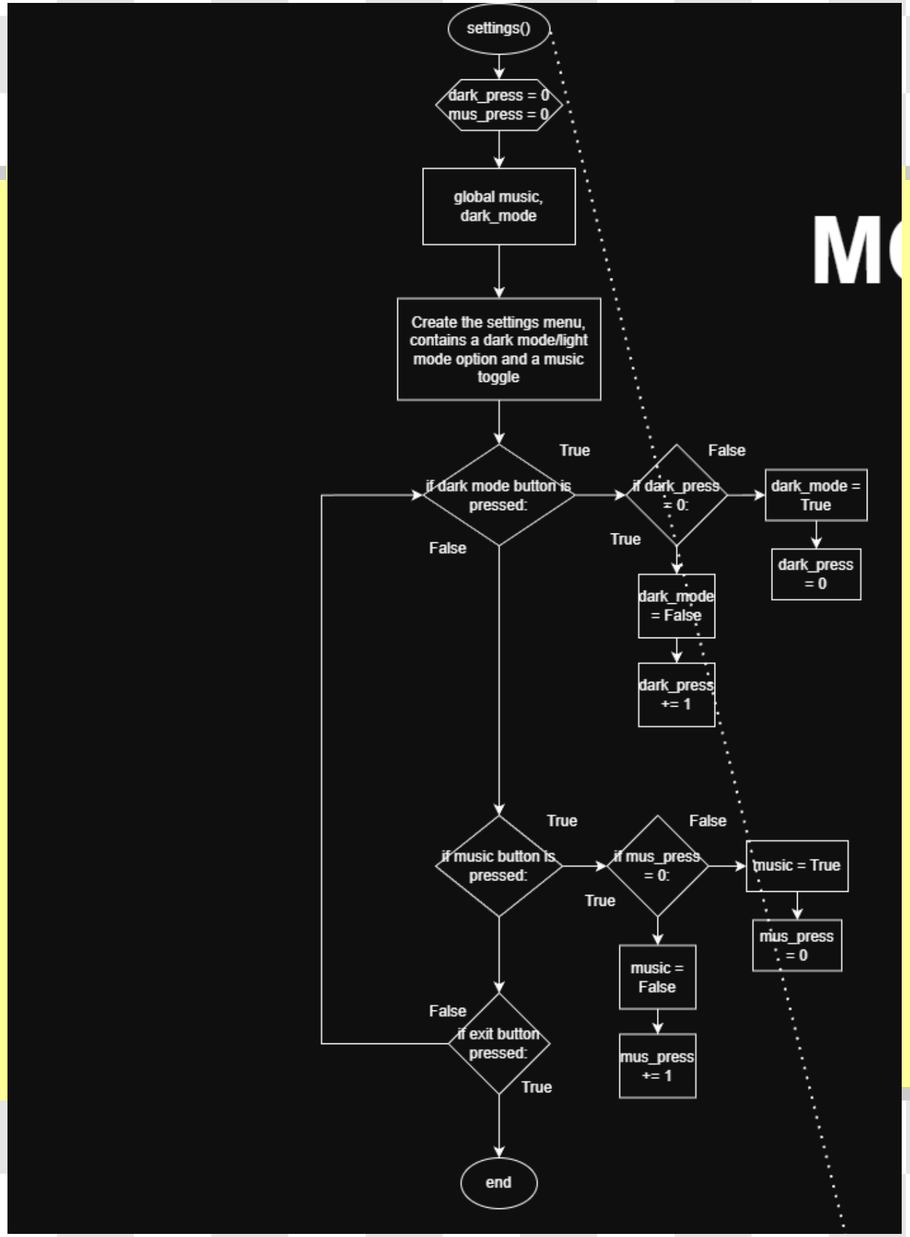



Back to Agenda Page

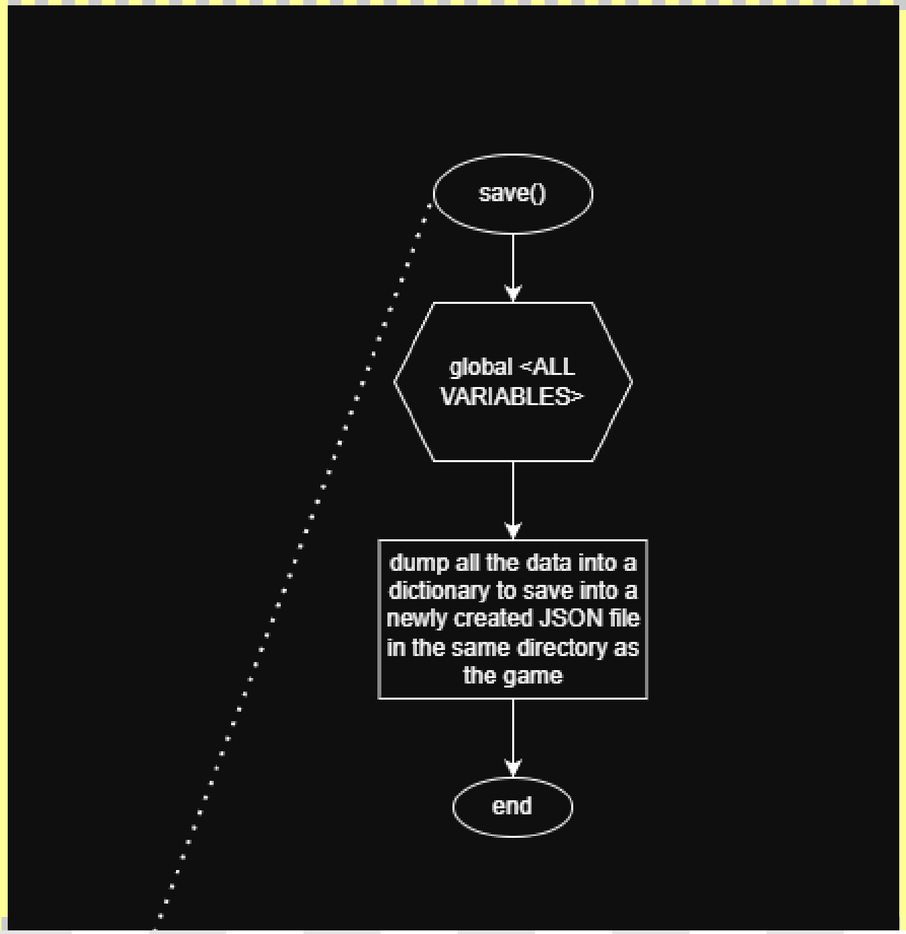

shop()
Create the shop window that contains functions linked to buttons to create a new object with varying stats (our generators) and also the stock, and to exit menu
Also make buttons play sound effects
if button to buy standard generator is pressed:
True
play sound effect
buy standard generator
False
if button to buy massive generator is pressed:
True
play sound effect
buy massive generator
False
if button to buy industrial generator is pressed:
True
play sound effect
buy industrial generator
False
if button to buy randomized generator is pressed:
True
play sound effect
buy randomized generator
False
if button to exit menu is pressed:
False
True
end



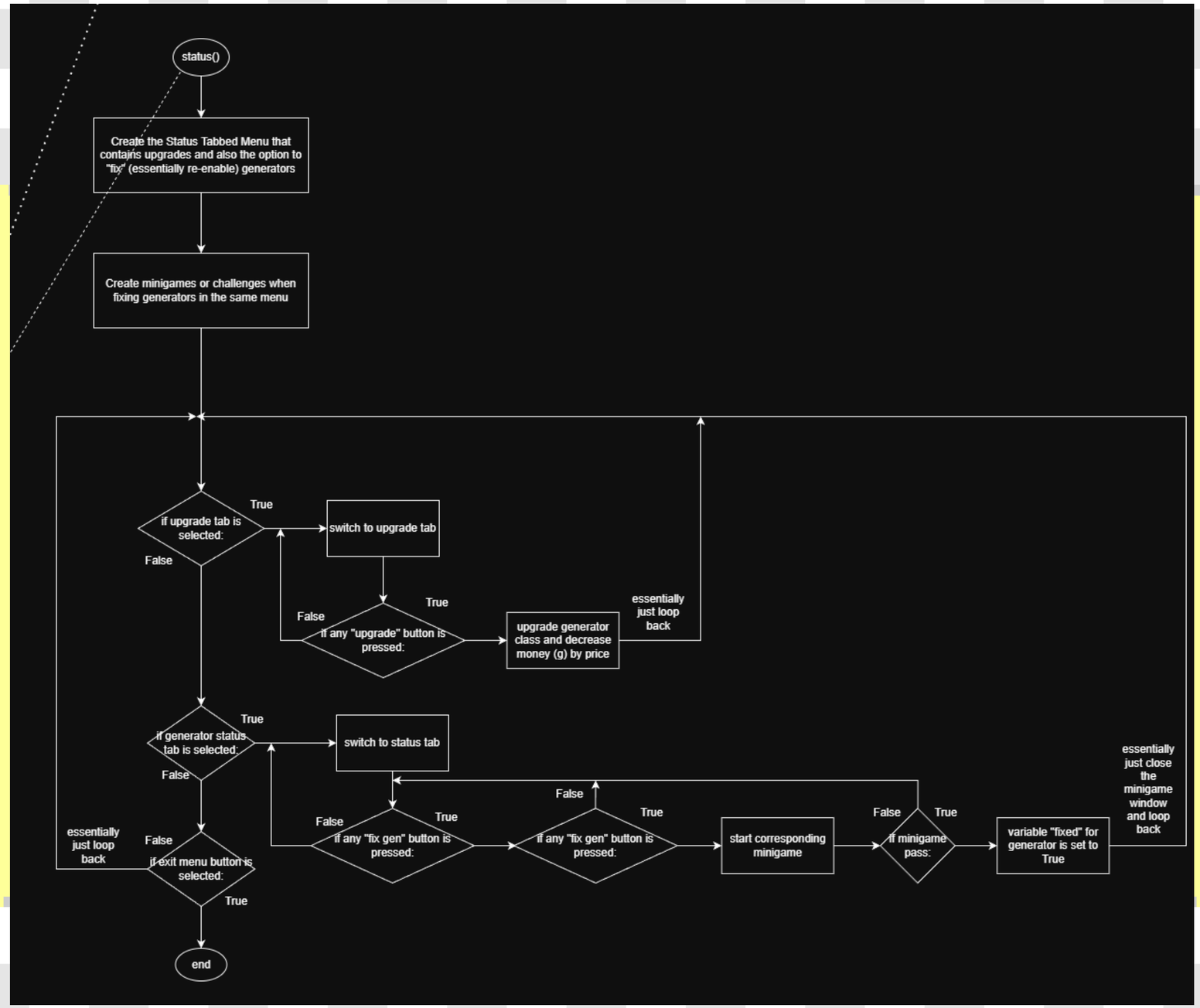
Shop menu



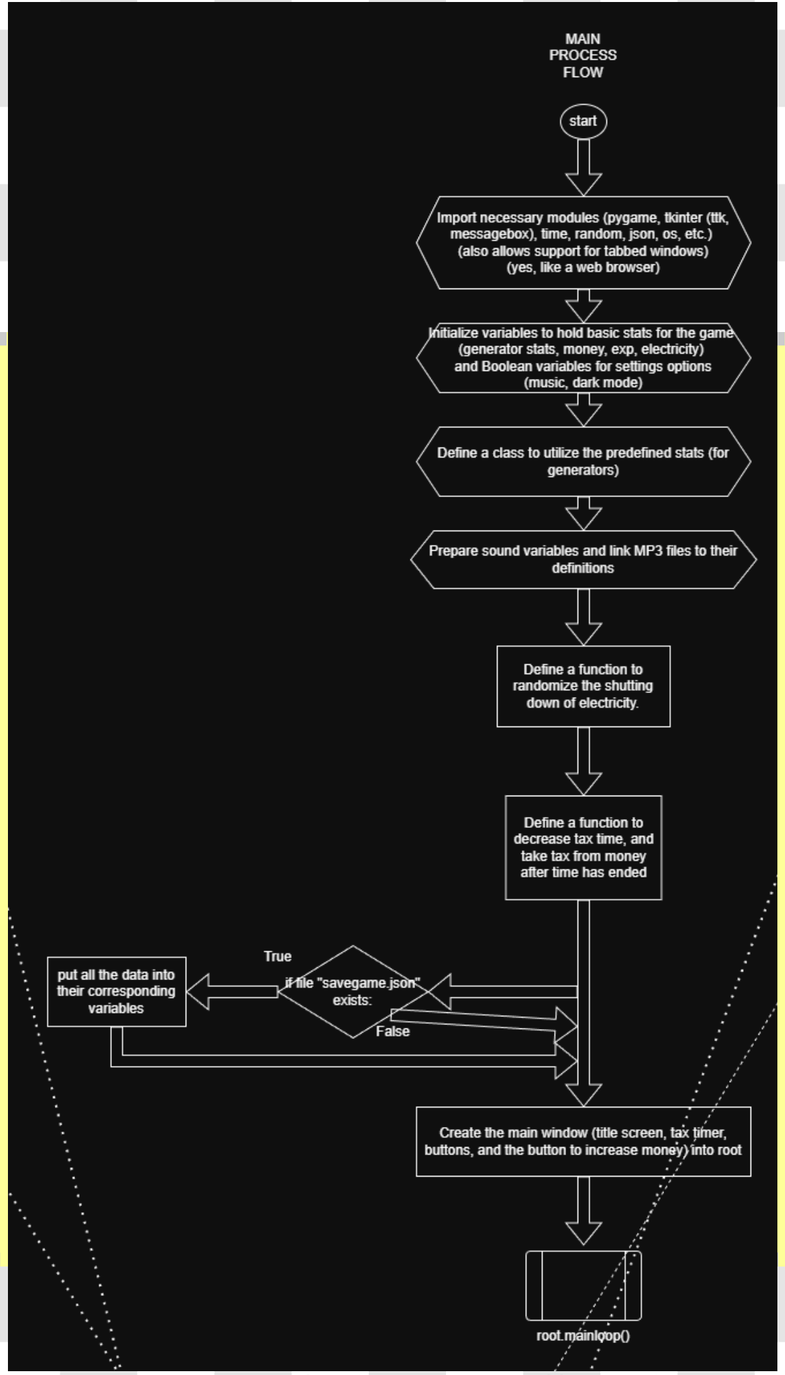
settings menu


save()
global <ALL VARIABLES>
dump all the data into a dictionary to save into a newly created JSON file in the same directory as the game
end

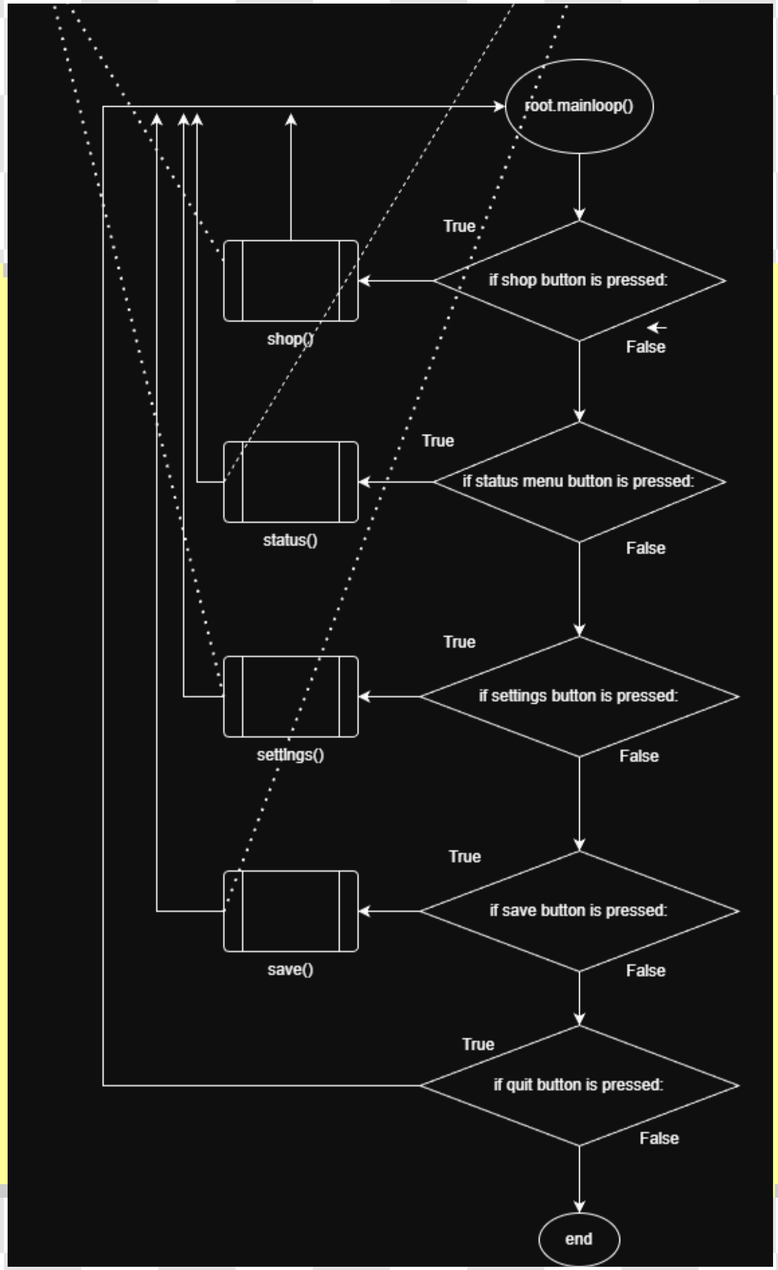
Save button



Status menu and Generator Fix/Upgrade buttons

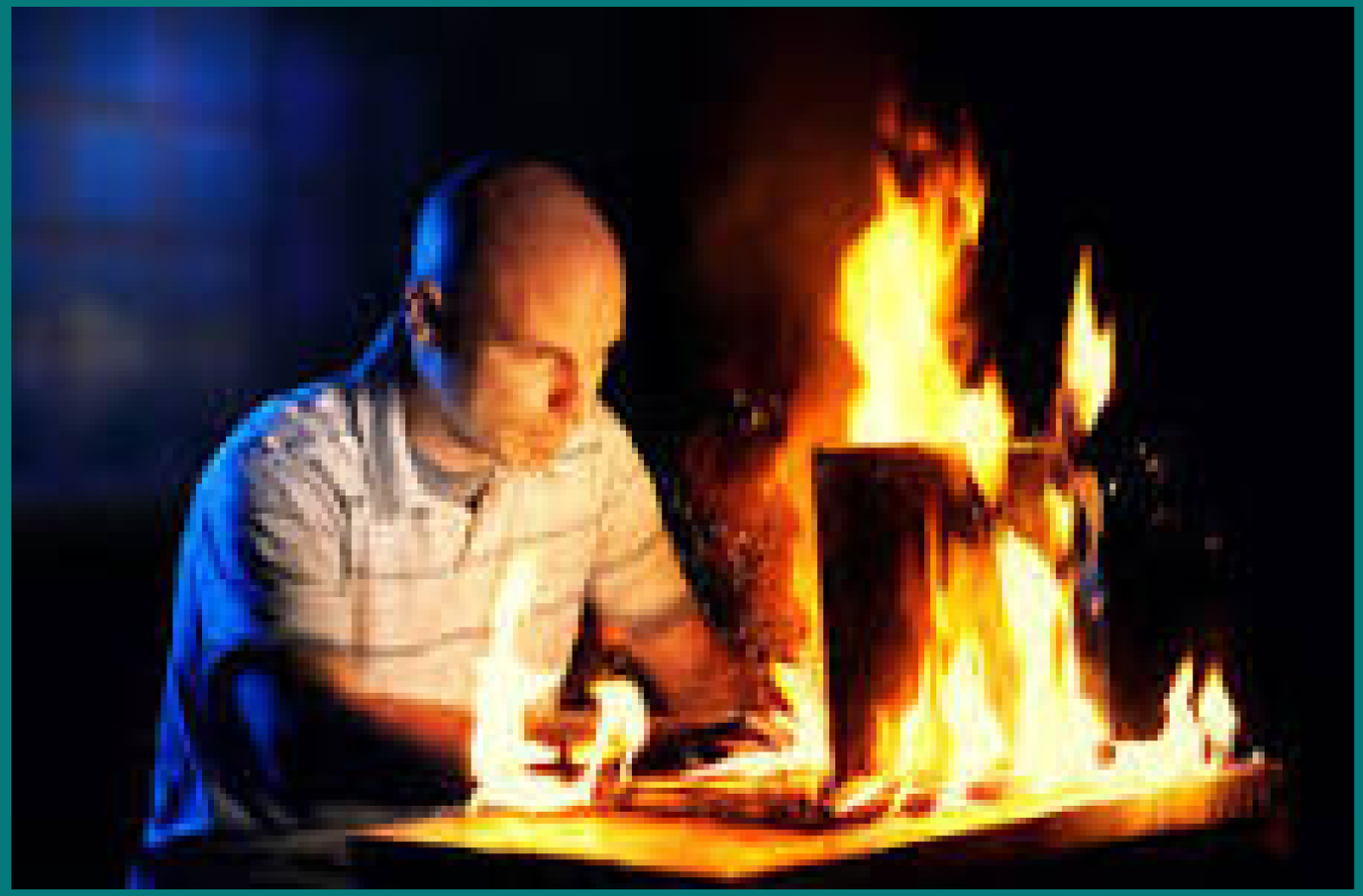


Main process flow (Main Menu and buttons)

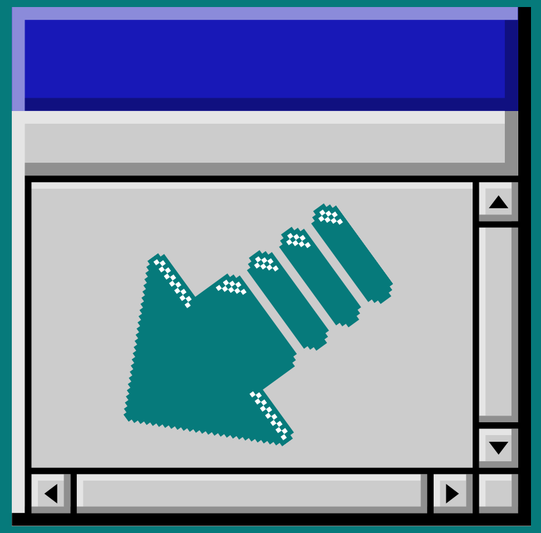


Root.mainloop() (something tkinter has to keep everything running but GUI tools have the same function in a different name

PSEUDOCODE



```
MAIN MENU LOOP
FUNCTION main_menu():
  WHILE True:
    CLEAR screen
    DISPLAY "Current G: ", user_G
    DISPLAY "Timer: ", timer.remaining_time
    DISPLAY "Tips: ", random_tip()

    DISPLAY BUTTONS:
      [1] Earn G
      [2] Shop
      [3] Generator Status
      [4] Exit Game

    EVERY 1/20 second (TICK):
      CALL tick_update()

    IF button [1] clicked:
      CALL earn_G()

    IF button [2] clicked:
      CALL open_shop()

    IF button [3] clicked:
      CALL generator_status_menu()

    IF button [4] clicked:
      CALL save_game()
      EXIT program
```



```
MAIN PROGRAM:
  LOAD saved_data.json
  INITIALIZE user_G, generators, electricity_status, timer, upgrades
  CALL main_menu()
```

1. Loads save data
2. Initializes user stats from data
3. Calls main menu function

1. Displays buttons
2. Updates game stats every tick (1/20th of a second)
3. Opens window corresponding to display button

```
GAME LOOP ACTIONS
FUNCTION tick_update():
  IF electricity_status == True:
    FOR each generator in generators:
      IF generator.broken == False:
        user_G += generator.output_rate / 20
      ELSE:
        CONTINUE  # Skip broken generators
  ELSE:
    DISPLAY "Electricity Down! Fix to resume power!"

  # Timer countdown
  timer.remaining_time -= 1/20 second
  IF timer.remaining_time <= 0:
    user_G -= user_G * 0.25
    RESET timer.remaining_time to default value

  # Random chance for generator breakdown
  FOR each generator in generators:
    IF random_chance(0.001):  # 0.1% chance per tick
      generator.broken = True
EARNING G MANUALLY
FUNCTION earn_G():
  user_G += 1
  DISPLAY "+1 G"
```

Electricity manager

Time before taxes and timer

Generator break chance

Clicking button to gain G manually

```
SHOP MENU
FUNCTION open_shop():
  WHILE True:
    CLEAR screen
    DISPLAY all available generators:
      FOR each generator_type:
        DISPLAY name, cost, output_rate, count_owned

    DISPLAY "[B] Buy Generator | [E] Exit"

    INPUT choice

    IF choice == "B":
      ASK which generator to buy
      IF user_G >= generator.cost:
        user_G -= generator.cost
        generator.count_owned += 1
        INCREASE generator.cost by inflation_rate
      ELSE:
        DISPLAY "Not enough G!"

    IF choice == "E":
      BREAK
```

```
GENERATOR STATUS MENU
FUNCTION generator_status_menu():
  WHILE True:
    CLEAR screen
    DISPLAY each generator:
      name, output_rate, broken_status, upgrade_level

    DISPLAY BUTTONS:
      [1] Fix Generator
      [2] Upgrade Generator
      [3] Fix Electricity
      [4] Exit
```

```
      INPUT choice

      IF choice == 1:
        ASK which generator to fix
        IF generator.broken == True:
          CALL fix_minigame(generator)
        ELSE:
          DISPLAY "That generator is already working!"

      IF choice == 2:
        ASK which generator to upgrade
        IF user_G >= generator.upgrade_cost:
          user_G -= generator.upgrade_cost
          generator.output_rate *= 1.2
          generator.upgrade_level += 1
          generator.upgrade_cost *= 1.5

      IF choice == 3:
        IF electricity_status == False:
          CALL electricity_memory_game()
        ELSE:
          DISPLAY "Electricity is already active!"

      IF choice == 4:
        BREAK
```

```
MINIGAMES
FUNCTION fix_minigame(generator):
  DISPLAY "Fixing generator... Type the
shown letters quickly!"

  SET timer = 5 seconds
  WHILE timer > 0:
    DISPLAY random letter
    WAIT for user input
    IF user input matches:
      CONTINUE
    ELSE:
      DISPLAY "Failed! Try again."
      RETURN

  DISPLAY "Generator fixed!"
  generator.broken = False
FUNCTION electricity_memory_game():
  DISPLAY "Memory Restore Game!"

  GENERATE sequence of 5 random letters
  DISPLAY them for 3 seconds
  CLEAR screen
  ASK user to type the sequence

  IF correct:
    DISPLAY "Electricity Restored!"
    electricity_status = True
  ELSE:
    DISPLAY "Failed! Try again later."
```

```
SAVE AND LOAD SYSTEM
FUNCTION save_game():
  data = {
    "user_G": user_G,
    "generators": list of generator states,
    "electricity_status": electricity_status,
    "timer": timer.remaining_time
  }
  WRITE data TO "saved_data.json"

FUNCTION load_game():
  IF "saved_data.json" exists:
    READ data
    user_G = data["user_G"]
    generators = data["generators"]
    electricity_status = data["electricity_status"]
    timer.remaining_time = data["timer"]
  ELSE:
    INITIALIZE default values
```

```
TIPS SYSTEM
FUNCTION
random_tip():
    tips = [
 "Click the G button
to earn G!",
 "Buy generators to
automate G
generation.",
 "Keep an eye on your
generators — they
can break!",
 "Don’t forget to
restore power when
it goes out."
 ]
    RETURN random
choice from tips
```

Thank you!

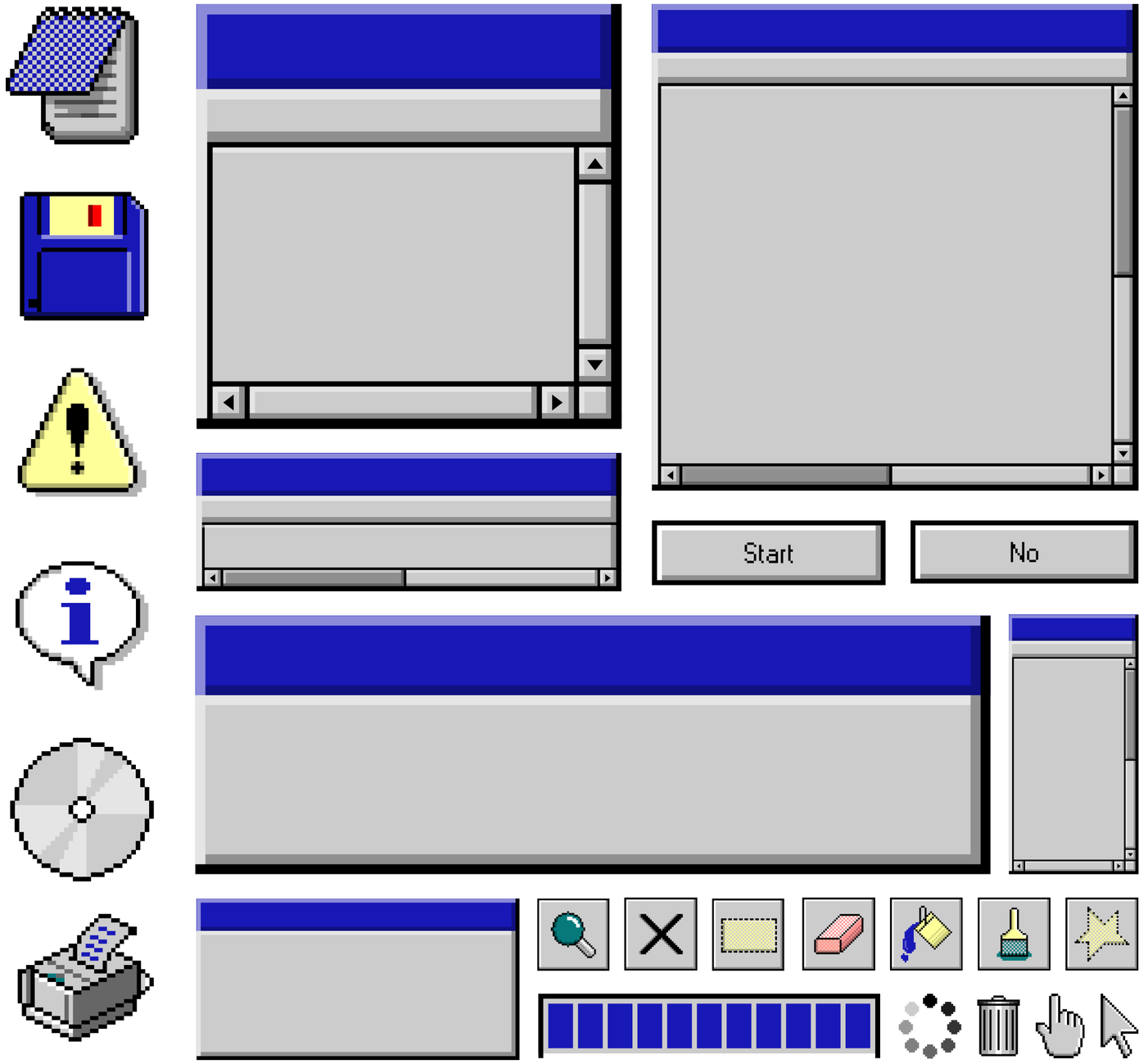
Github Link:

https://github.com/xbbatingkay-hub/camia-BatingkayOlympia

Resource Page

Use these design resources in your Canva Presentation. Happy designing!

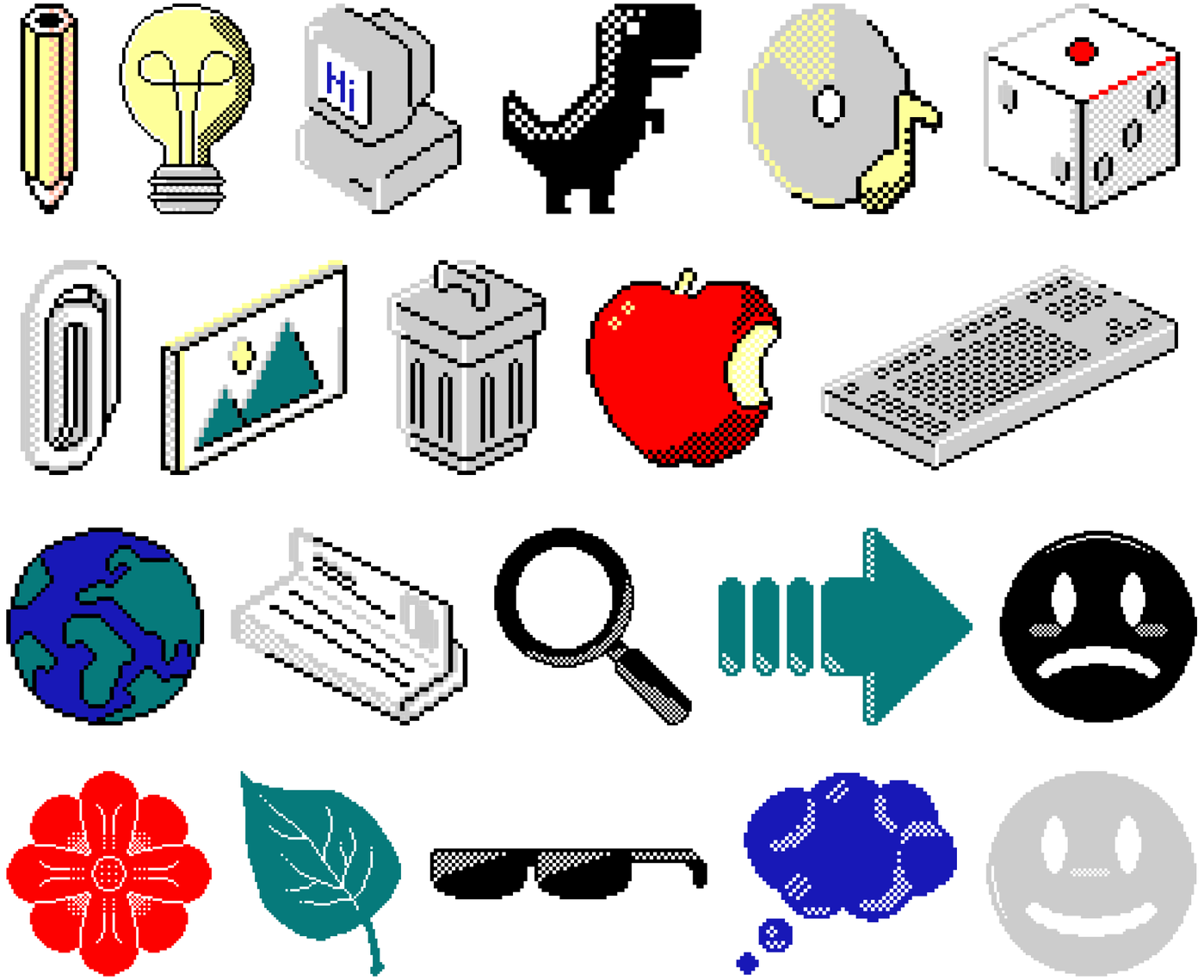
Don't forget to delete this page before presenting.



Resource Page

Use these design resources
in your Canva Presentation.
Happy designing!

Don't forget to delete
this page before presenting.



Resource Page

Find the magic and fun in presenting with Canva Presentations. Press the following keys while on Present mode!

Delete this page before presenting.

B for blur	C for confetti
D for a drumroll	M for mic drop
O for bubbles	Q for quiet
U for unveil	Any number from 0-9 for a timer

Resource Page

Want more flexibility as a presenter? Record your presentation to engage in a way that works for you — and let your audience watch in a way that works for them.

Delete or hide this page before sharing.

Go to Uploads, then Record Yourself.

All recordings will be saved on Canva in your Uploads folder.

Edit your recording. Feel free to adjust the background, volume, and duration.

Need multiple presenters?
Select Share then update Collaboration link to Can edit.

Share editable link. Assign team members to record their sections.

You can also mix live presentation and recorded videos to make your session more engaging.