Chat bot ??

I need

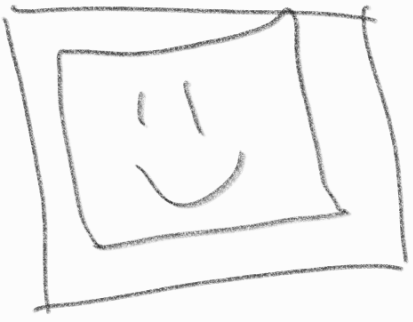
- Python
- Vader (Detect emotion)
- API
- Mascot (Pixel art)
- Emotion Detection (Keyword Mapping)
- Session management ??
- Sound effects ??

Deadline

20 Days

Mascot:

- Computer Screen



5 expressions per emotion.

- idle - 2
- Happy
- sad
- shocked
- glitch
- error / mad
- Good bye !

272 strides

- Robot
  - cute
  - 64 x 64

APIs

- OpenRouter AI
  - to Mistral, Mixtral
  - to open chat
  - to hugging face models

Logic

- gets user input
- Api searches for a reply
  - Associate input with an emotion
- Displays according reaction
- gives answer
- prompt for new input

Python with TKinter

Displays sprites.

Easy GUI.

Python

- Main.py
- Chatbot_logic.py
- mood detector(Vader)
- Assets
  - Sprites
- Requirements.txt

1) install Vader
2) Bot logic
   - get input
   - pass to reply function
   - check mood (pass mood detect)
   - Change sprite.
   - pass answer

Chatbot logic

Connect API Key to handle Answer

get response from (user input)
- if input valid → pass it to mood detector
  - else Display error message
  - Exception to handle error

---

mood Detect

Download Vader for sentiment Detect.
create a Diccionary with emotions
- Happy → good, great...
  - Sad → Bad, tired...
  - Angry → Mad, sucks...

→ Pass user input to mood_Detector function
- loops thru Diccionary to

match words.

Create a "mood" score

- to initialize it to 0
- if score >= 0.5 (happy)
  - elif socore <=-05 (sad)
  - else neutral
- to return top mood

Sprites handling

import random (to get randomized images)

- Connect to images path folder
- Create an empty Diccionary for images
- Give all images same name as emotion: happy1, sad2, ...

- separate numbers and file axtansion

  happy1.png → happy

- Add them to the Diccionary

- Match top mood with image of same name. grab a random one to Display

---

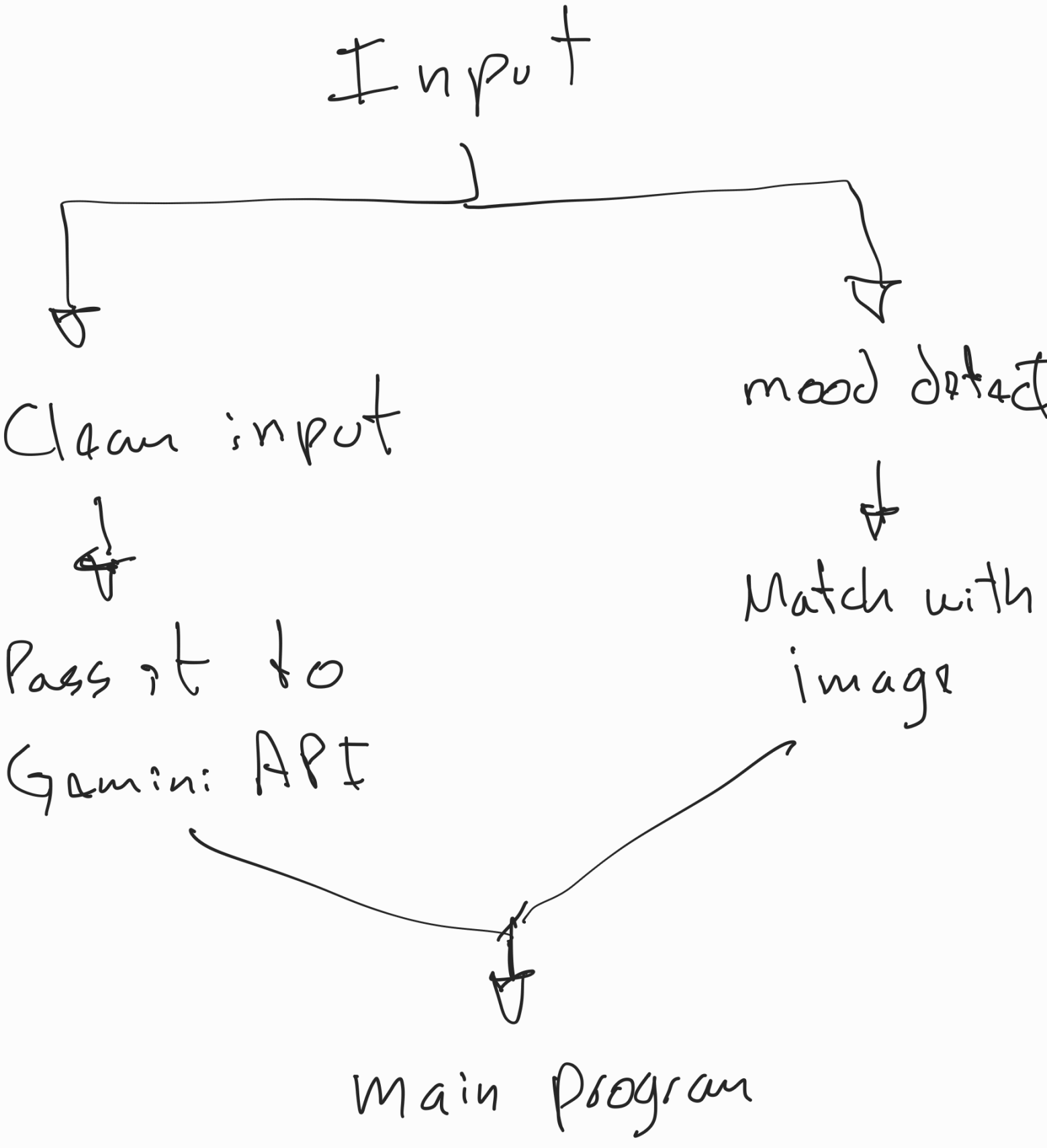
main

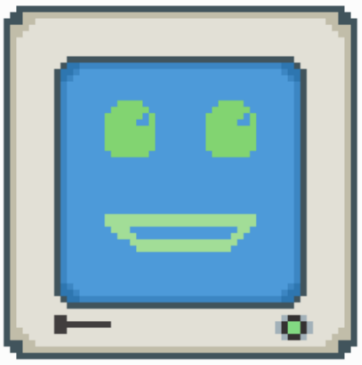
GUI → handled by Tkinter

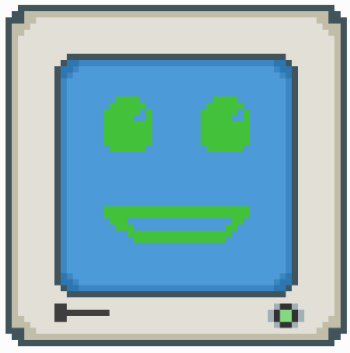
GIF → Pillow

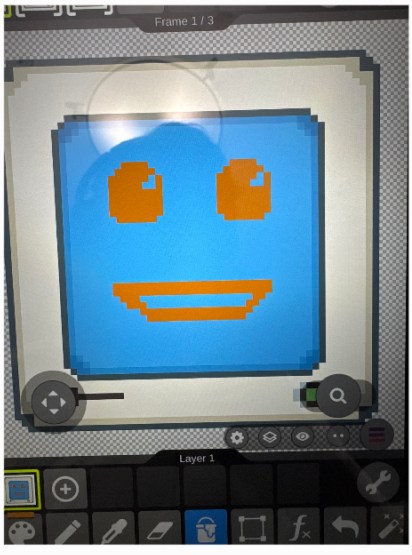
Sound? → Pygame

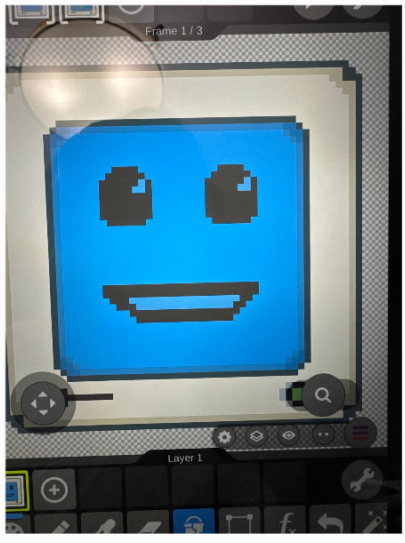
Random → get random images


Input
Clean input
mood detect
Pass it to
Gemini API
Match with
image
Main Program






Frame 1 / 3
Layer 1


Frame 1 / 3
Layer 1


MACNCHEESE


MACNCHEESE


MACNCHEESE


MACNCHEESE


MACNCHEESE


MACNCHEESE