

how to make ur project not
look like this slide (aka kind
of shitty)

An introductory talk on User Interface and PIL

Jenny Yu

No, seriously, UI makes a difference,
even with tkinter



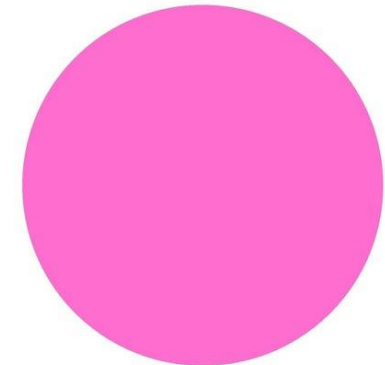
graphic design is my passion

Do's

- PLEASE USE IMAGES/PICTURES/CLIPART
- Put some thought into color scheme
- Font (doesn't this look so much better without comic sans???)
- Potentially use PIL (what's this??)

Don'ts

- `canvas.create_rectangle()` and `canvas.create_oval()` everywhere
- Use default tkinter colors ("hotpink", "green2", etc)
- Default font, disproportional font, illegible font



A dark, irregular ink blot with white text "Let's talk about pictures" centered on it. The blot is surrounded by a light, textured background with scattered dark specks.

Let's talk about
pictures

A COUPLE OF OPTIONS FOR IMAGES/ANIMATION:



USE TKINTER AS A
DEFAULT (WHICH CAN
STILL LOOK VERY NICE
AND FUNCTION WELL!)



FOR THOSE
ARTISTICALLY-INCLINED,
CREATE/DRAW YOUR
OWN ART



USE PIL/PYGAME,
WHICH CAN TAKE
SOME EFFORT TO LEARN



NOT USING IMAGES AT
ALL AND STICKING TO
SHAPES



SOMETHING ELSE
COMPLETELY. SHOW
YOUR CREATIVITY!



STICKING WITH OUR OLD FRIEND (TKINTER)

TKINTER UI GUIDELINES

- Use color palettes (for background, text, images, art!)
 - <https://coolors.co/>
 - <https://htmlcolorcodes.com/color-picker/> (color picker)
 - Format colors using hexadecimal code
 - fill = “#RRGGBB” (but with the color you actually want instead after the “#”)
- When in doubt: stick with high contrast background and text (e.g. white on black, navy blue on white, pastels on black, etc)
- Images with transparent backgrounds

THING THAT EXIST THAT WE DID NOT TEACH YOU IN ANIMATION WEEK ABOUT TKINTER

- You can modify the `run()` function to add more features:
 - Registering multiple keyboard inputs at the same time
 - Register mouse movement and mouse being released
 - Delta graphics (to make animation run faster when you have excessive pictures)
- To learn more: <http://www.kosbie.net/cmu/fall-16/15-112/notes/notes-tkinter-demos.html>
- Animating images



Brief Intro to PIL (Pillow)

What the heck is PIL? Why use it?

- ▶ Python Imaging Library
- ▶ Documentation: <http://effbot.org/imagingbook/>
- ▶ Gitbook for brief overview: <https://abhgog.gitbooks.io/pil/content/>
- ▶ Tkinter can be too limited:
 - ▶ Can't use .jpg or .png (for Macs)
 - ▶ Hard to resize/rotate images
 - ▶ Hard to do anything with images except display them ☹️

Some cool/powerful functionalities of PIL

- ▶ Resize/rotate images easily
- ▶ Filter images easily
- ▶ Can change individual pixels of your choosing (an image can be represented by a 2D list of pixel values in RGB)
 - ▶ Especially useful for image processing/manipulation projects
 - ▶ Edge detection
- ▶ Can be easily integrated into your animation framework by using ImageTk provided by PIL
- ▶ Give you satisfaction about learning something new

How to install on pyzo?

```
Shells
Python
>>> pip install pillow
Collecting pillow
  Downloading https://files.pythonhosted.org/packages/bd/39/c76eaf781343162bdb1cf4854cb3bd5947a87ee44363e5acd6c48d69c4a1/Pillow-5.3.0-cp36-cp36m-win_amd64.whl (1.6MB)
Installing collected packages: pillow
Successfully installed pillow-5.3.0
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

>>> import PIL    #no error here if successfully installed

>>> import sockets
Traceback (most recent call last):
  File "<console>", line 1, in <module>
ModuleNotFoundError: No module named 'sockets'

>>> |
```

#if not installed correctly, error message will look like this

After installation...some basics

```
1 from PIL import Image
2
3 #loads the image
4 im1 = Image.open("img1.png")
5
6 #rotates image CCW by 45 degrees
7 im2 = im1.rotate(45)
8
9 #save the image to your disk
10 im2.save("rotated.png")
11
```

Pixel manipulations example

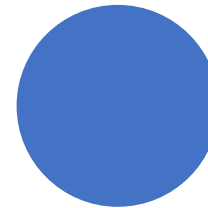
```
11
12 # create a new black image
13 img = Image.new( 'RGB', (255,255), "black")
14 pixels = img.load() # create the pixel map
15
16 for i in range(img.size[0]):
17     for j in range(img.size[1]):
18         newPixel = (255, 128, 100) # whatever you want here
19         pixels[i,j] = newPixel
```

Examples of PIL projects

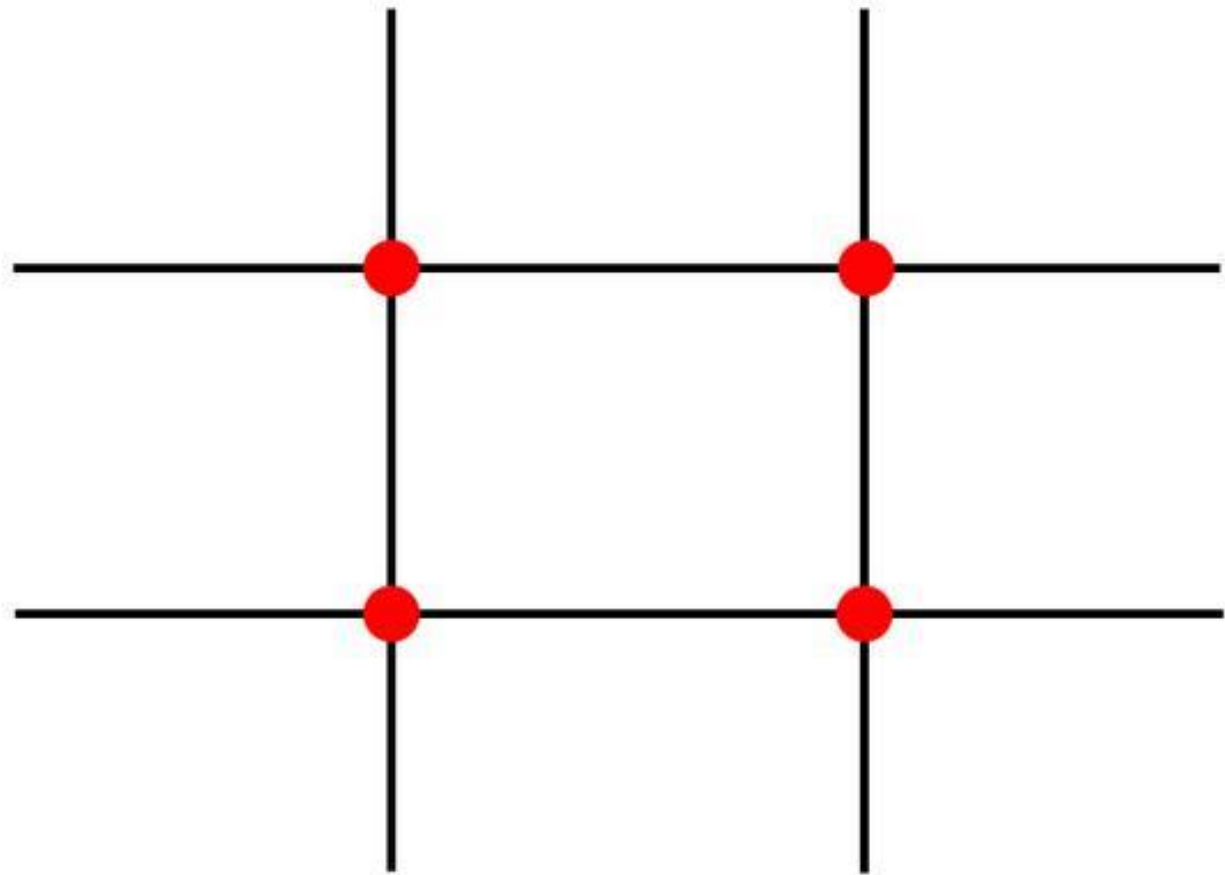
- ▶ Artify: <https://www.youtube.com/watch?v=ePl7bsKpd2k&feature=youtu.be>
 - ▶ Turn an imported image into a different style of art
- ▶ Color Deficiency Test: <https://www.youtube.com/watch?v=YGtb-YMbtf8&feature=youtu.be>
 - ▶ Test your color vision with generated images
- ▶ Snapchat: <https://www.youtube.com/watch?v=5YjqNkB-3Q0>
 - ▶ Add filters/emojis/text to pictures



Layout Design

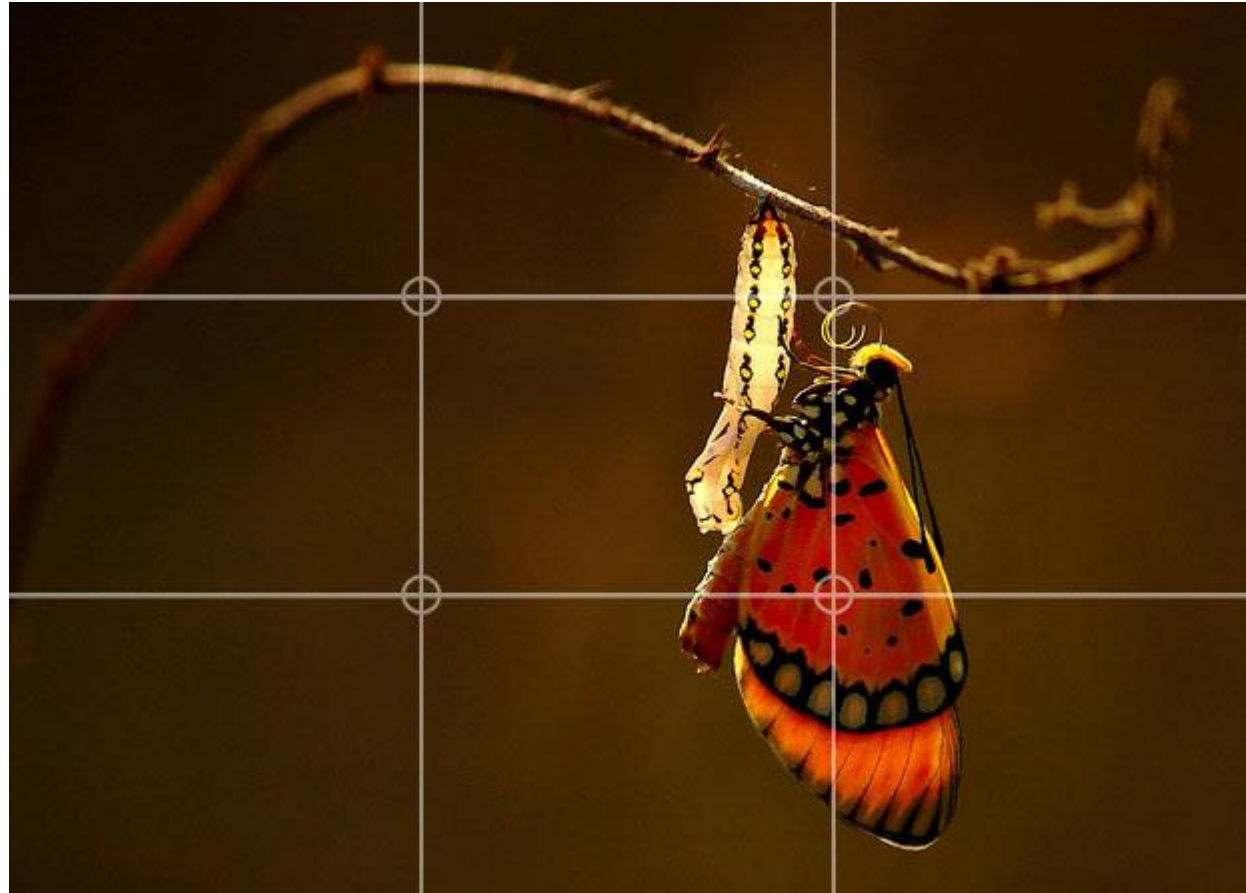


Rule of thirds



Important elements are along these lines or points of intersection

- Section off parts of your canvas
- Decide how to break down your design

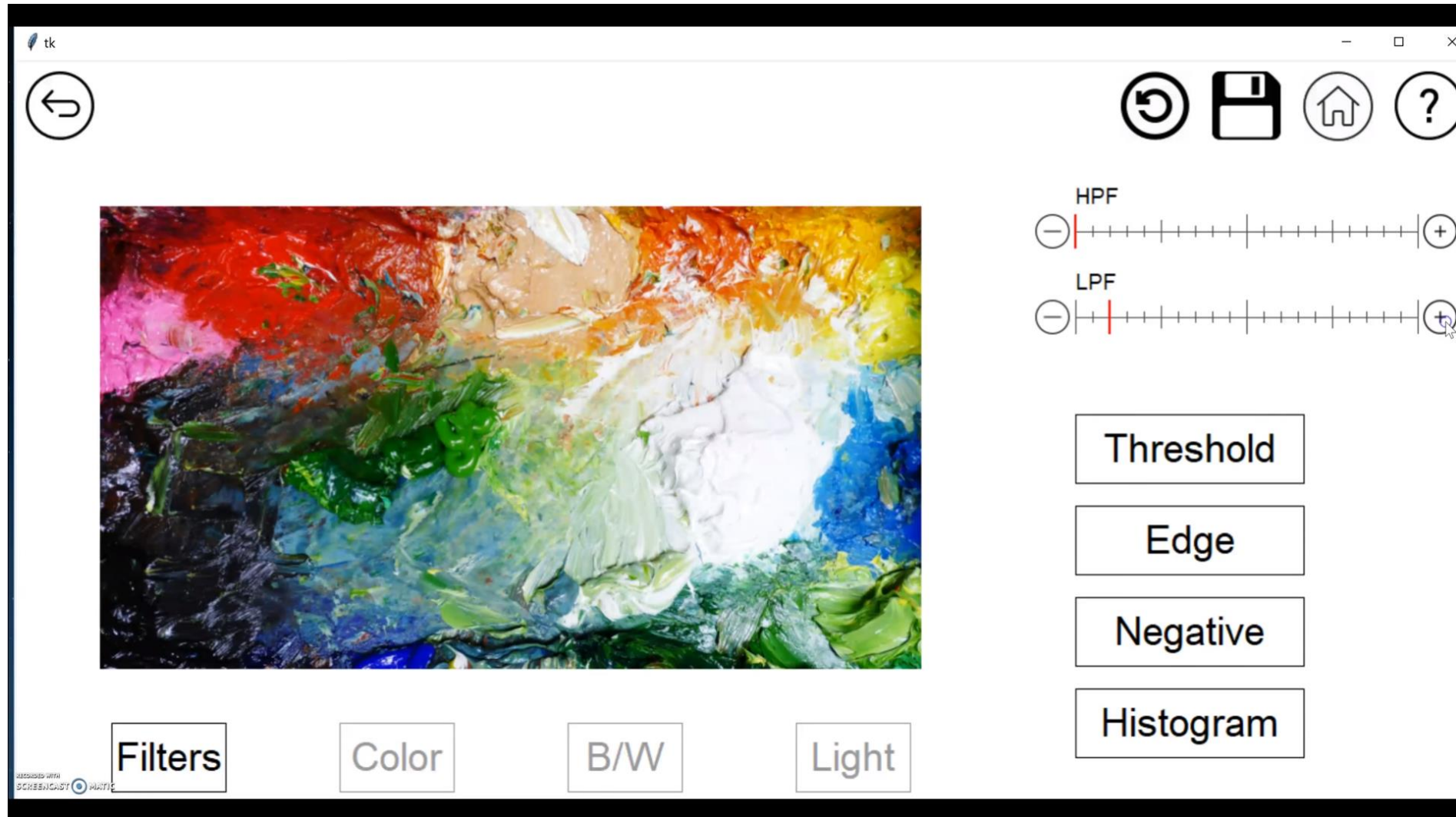




General tips

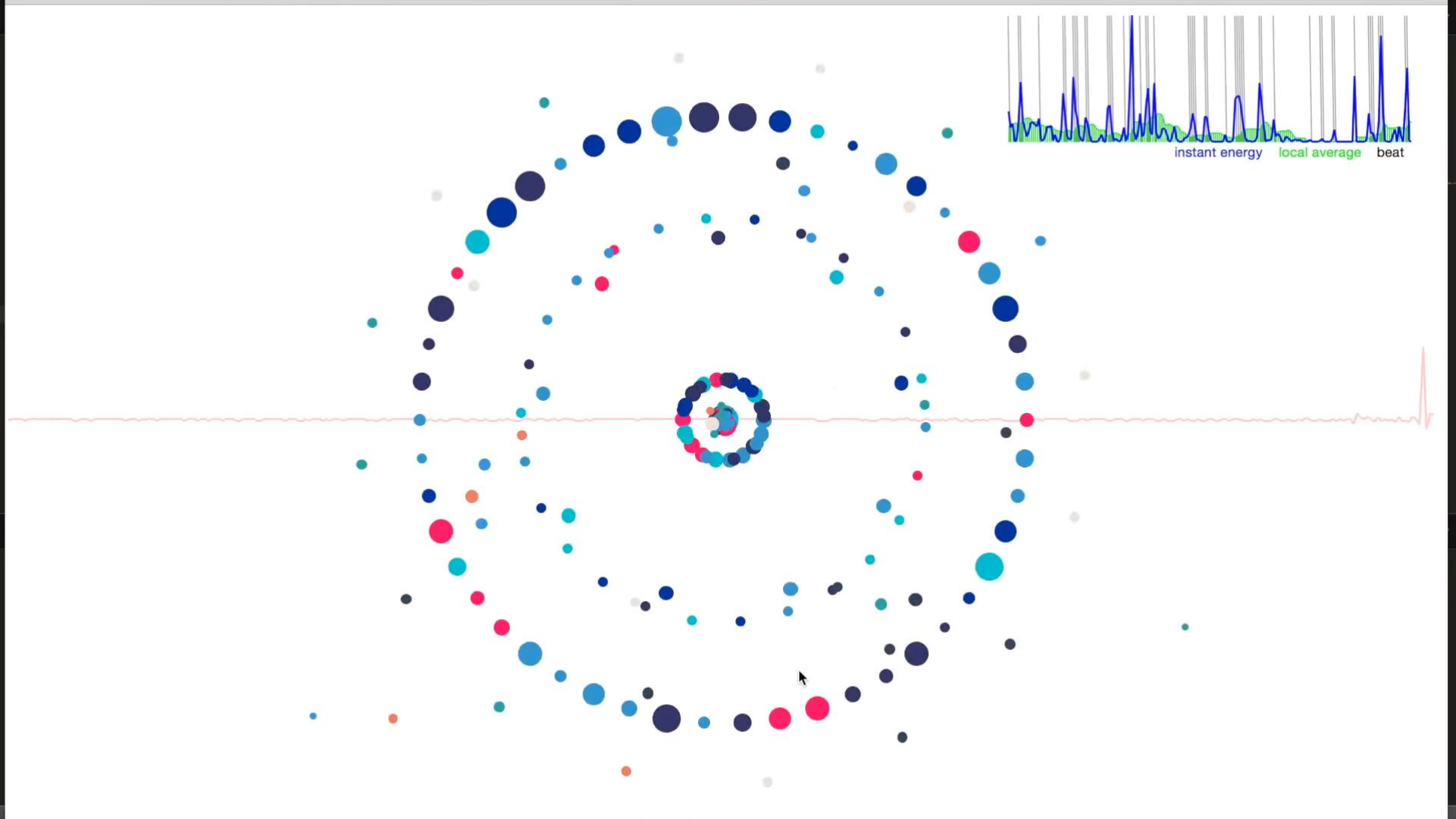
- Make sure eyes are drawn to parts of the canvas in logical order
- Everything is intuitive
- Simpler is better. Do not overload the eyes
- More whitespace > less whitespace

Examples of good UI



FPS: 33.0







Just remember:
bad UI \neq bad project

But good UI is like a wrapper
function: makes everything
a lot nicer 😊