



CMU Graphics Tips

Contents

- Speed of CMU Graphics
- Things guaranteed to cause lags
- Speeding up image drawing
- Timer event pattern to reduce lag
- Sound
- Other tips
- Arcade

CMU Graphics speed

CMU Graphics



What causes lags

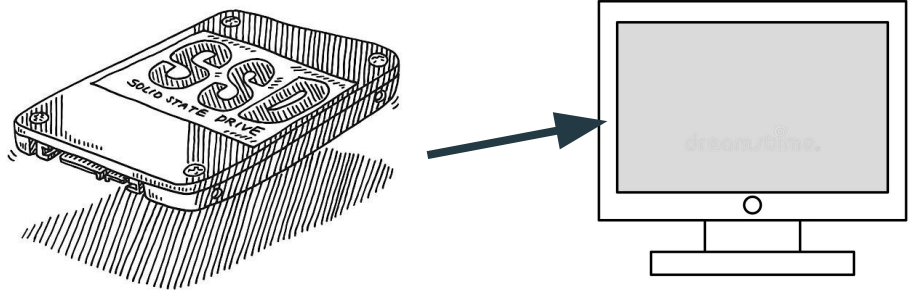
- IMAGES
- A LOT of stuff happening in `redrawAll()`
- Extremely inefficient algorithms
- Messing with `app.stepsPerSecond`

Biggest cause of lag : Drawing Images!

`drawImage(url, left, top, width=int, height=int)`

Problem 1:

- Loads the image every single time from the url, every time it is called
- Very slow!

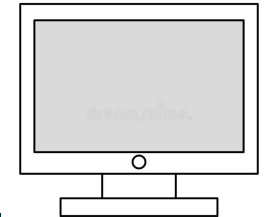
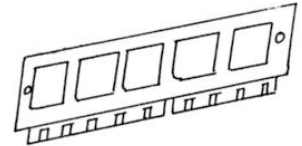
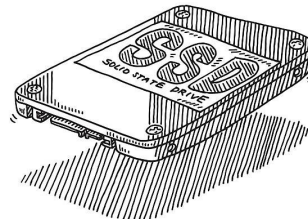


Solution: Image Caching

```
from cmu_graphics.shape_logic import  
loadImageFromStringReference
```

```
image =  
loadImageFromStringReference(  
    'images/eduardo.png')
```

```
drawImage(image, x=0, y=0, width = self.width, height  
= self.height)
```



Another issue: large image size

`drawImage(image, left, top, width=int,
height=int)`

Problem 2:

- If you have a very large image and you change the width and height parameters, it will still load the large image into memory
- Change actual image dimensions instead



Demo

onStep() - messing with app.stepsPerSecond

- Changing this will cause unpredictable behaviour, sometimes causing issues
- Also difficult to use if you want to control the speed of different events
- Keep app.stepsPerSecond constant and change something else...



#30 times per second by default?

```
def onStep(app):  
    app.timer += 1  
    # If you want to run it 5 times per second  
    desired = 5  
    interval = app.stepsPerSecond // desired  
    if app.timer % interval == 0:  
        runCode()
```

Sound

Part of CMU Graphics

```
# creates a sound object
track = Sound("url")
# plays a sound object
track.play(loop=False, restart = False)
# pauses sound
track.pause()
```

(NOT THAT SIMPLE)



Loading sound locally

```
import pathlib
cntPath = pathlib.Path(__file__).parent.resolve()
cntPath = str(cntPath)
cntPath = cntPath.replace("\\", "/")
menu = Sound(f'file:/// {cntPath}/menu.mp3')
upgrade = Sound(f'file:/// {cntPath}/upgrade.mp3')
gameOver = Sound(f'file:/// {cntPath}/gameOver.mp3')
story = Sound(f'file:/// {cntPath}/story.mp3')
won = Sound(f'file:/// {cntPath}/won.mp3')
# Jacob, E (2023)
```

Sound Bug!

Once the sound loops
more than once, you can
no longer pause it

**Solution: Play the sound
right before pausing it**

```
won.play()  
won.pause()  
upgrade.play()  
upgrade.pause()  
story.play()  
story.pause()  
gameOver.play()  
gameOver.pause()
```

Other tips

Use the CMU Graphics built-in functions!



▼ Built-In Functions

distance

angleTo

getPointInDir

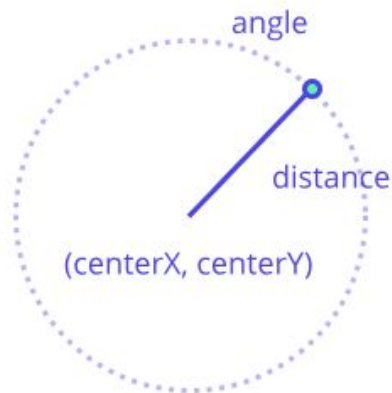
rounded

makeList

sleep

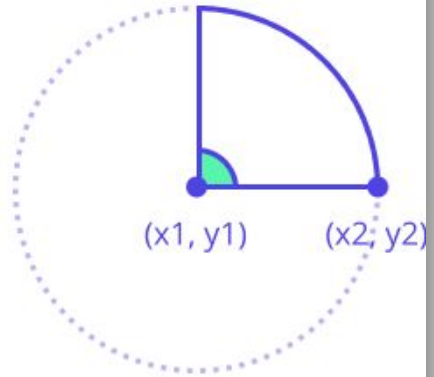
getPointInDir(centerX, centerY, angle, distance)

Returns a point that is the given number of pixels from the center point, in the given direction.



angleTo(x1, y1, x2, y2)

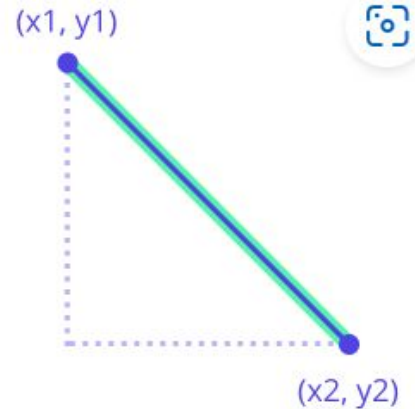
Returns an integer representing the angle from the point (x1, y1) to (x2, y2) with straight up being 0 degrees.



distance(x1, y1, x2, y2)

Returns a number representing how far apart the point (x1, y1) and (x2, y2) are.

This is equivalent to the square root of $((x1-x2)^2 + (y1-y2)^2)$.



Arcade 🤑😎🔥

Beautiful, state of the art
arcade machine

Perfect for your post-MVP
addition

Easy to transport (kind of)



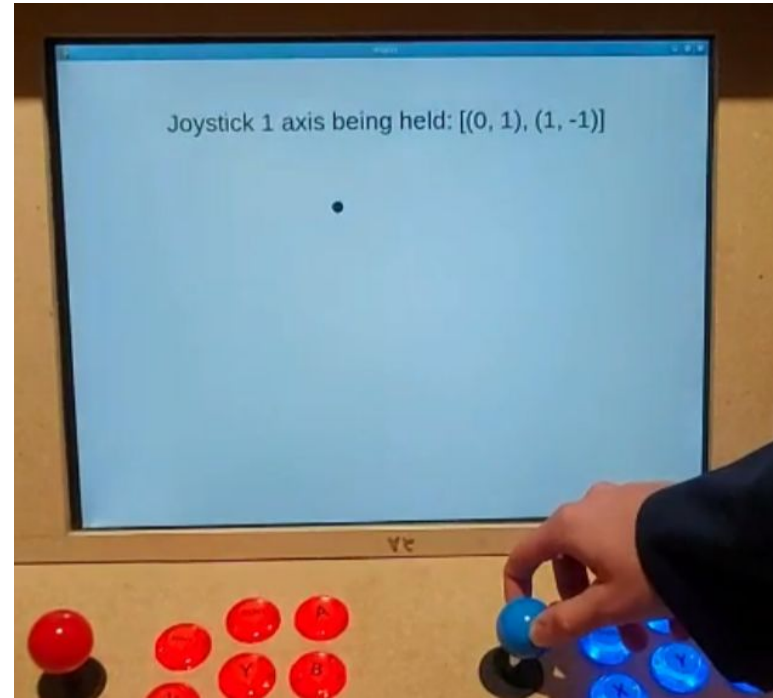
```
def onJoyPress(app, button, joystick):
```

```
def onDigitalJoyAxis  
(app, results, joystick):
```

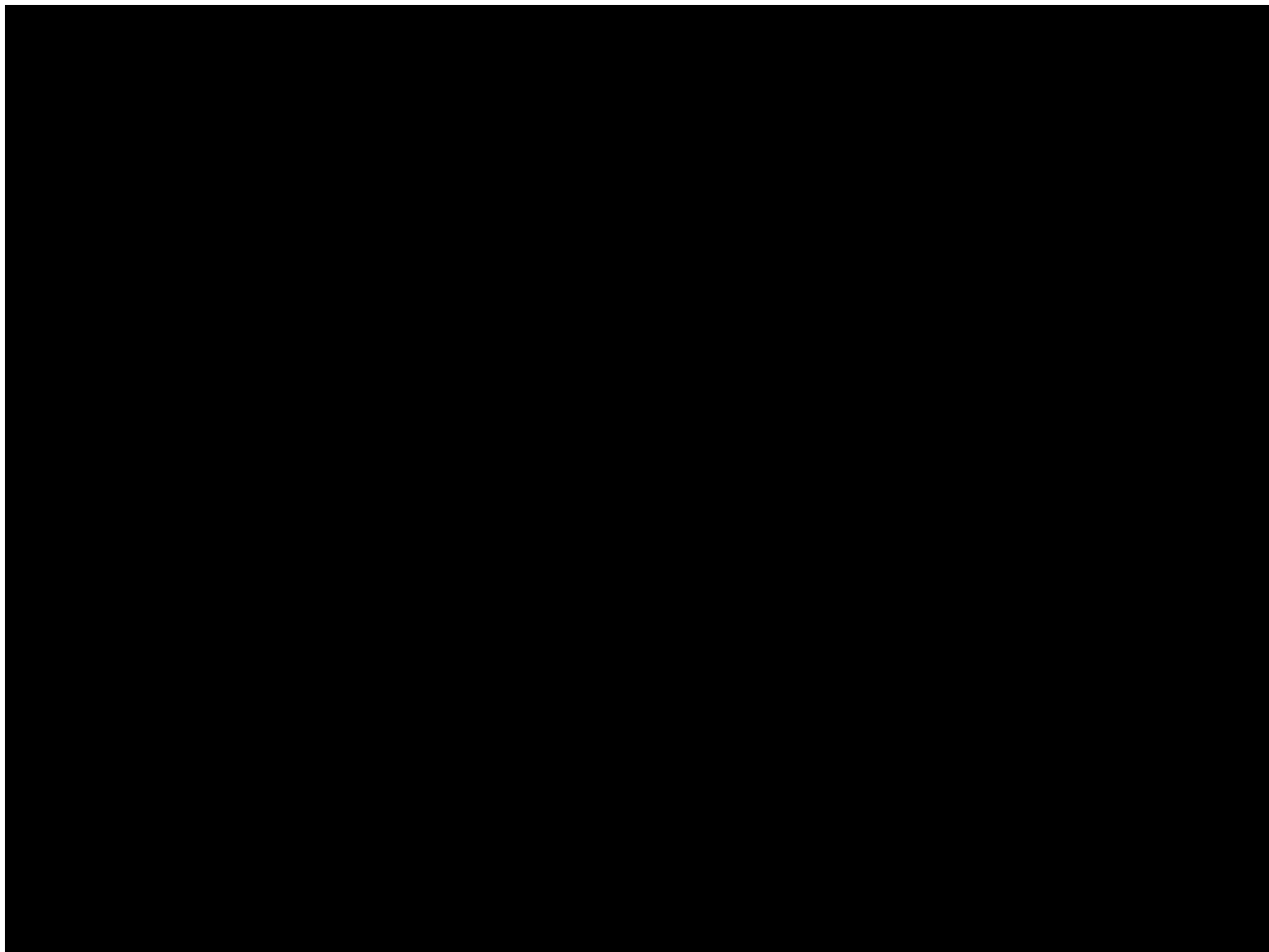


What inputs do the functions take?

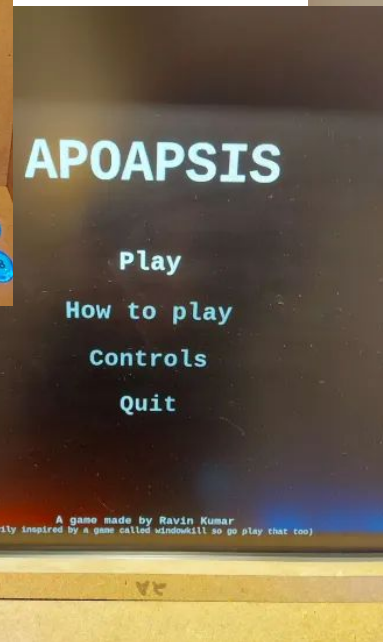
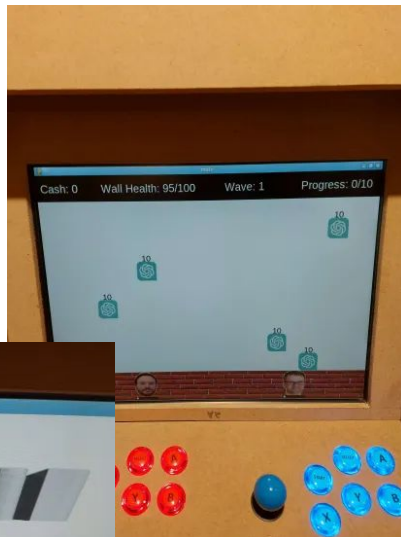
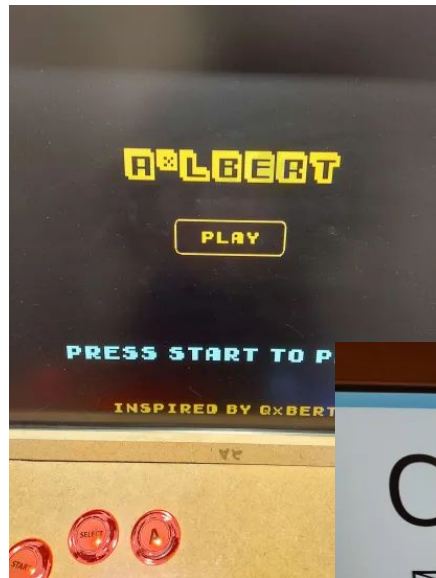
```
def onJoyPress(app, button, joystick):  
    """Pressing Triangle and X triggers a size change in the circle"""  
    app.text = f"Joystick {joystick} button pressed: {button}"  
    if button == "3":  
        app.player.getBigger()  
    elif button == "0":  
        app.player.getSmaller()  
    # Make sure we can exit by pressing the "P1" button  
    elif button == '5':  
        sys.exit(0)
```



<https://github.com/CMU15-112/arcade-box-startercode>



There's many games there already



Q/A