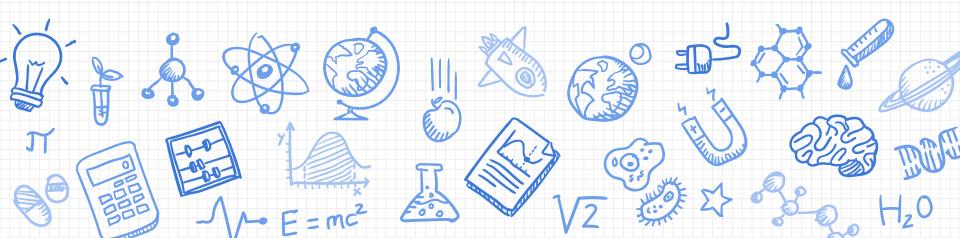
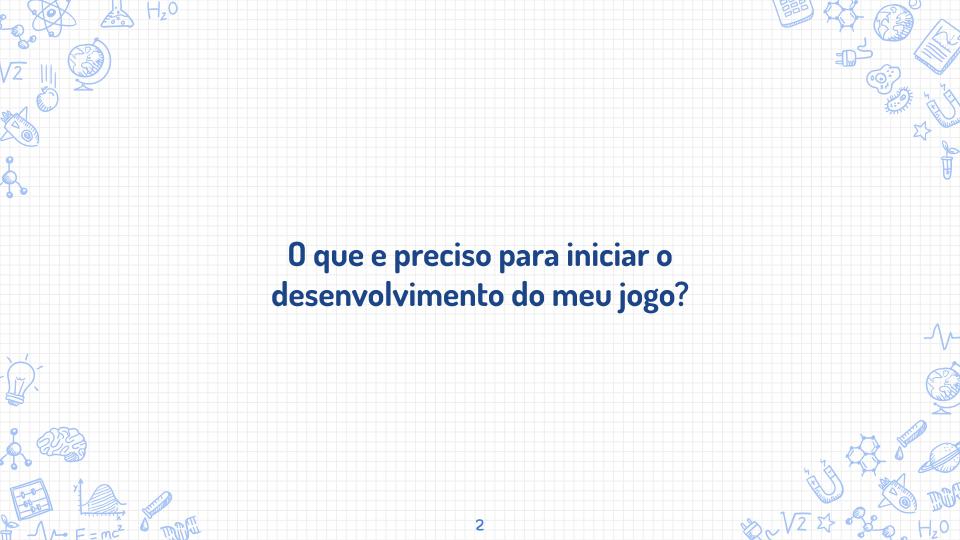
# Desenvolvendo um jogo Python e Graphics.py





# O que e preciso para iniciar o desenvolvimento do meu jogo?

- Decidir se irá criar um jogo, ou fazer um clone;
- ✗ Analisar as estruturas de dados;
- Definir um escopo; Iniciar o desenvolvimento;



#### Criar ou Clonar um Jogo?

- Criar um jogo implica em pensar em todas as mecânicas e desenvolvê-las do zero.
  - X 0 que o jogador irá fazer?
  - Como ira fazer?
- Fazer um clone implica em estudar o jogo alvo e clonar suas mecânicas (debriefing).
  - X 0 que o jogador faz?
  - Como o jogador faz?



- Que estrutura de dados (tipo de variável) melhor se adequa ao seu jogo?
  - X Tipos primitivos (int, str, bool, float)
  - X Lista?
  - X Matriz??
  - X Modularização????

Escolher a estrutura de dados certa pode facilitar o desenvolvimento.





Em um jogo de cartas a estrutura de dados lista pode ajudar bastante, já que fica muito mais fácil trabalhar com índices do que com 108 variáveis!





Em um jogo de tabuleiro o que vale mais? 64 variáveis, 8 listas ou uma matriz?





Mas e agora? O que usar?

Modularizar os dados pode ajudar!



#### Modularização



0	1	2	3
Nome	Tipo	Nível	Habilidades
"Turtwig"	"Grama"	20	
			Ţ
0		1	
0	1	0	1
Nome	Atk	Nome	Atk

"Folha Navalha"

12

O objetivo com isto e manter as variáveis agrupadas!

Pokemon[3][1][1] = Valor de atk da segunda habilidade do pokemon

"Atacar"

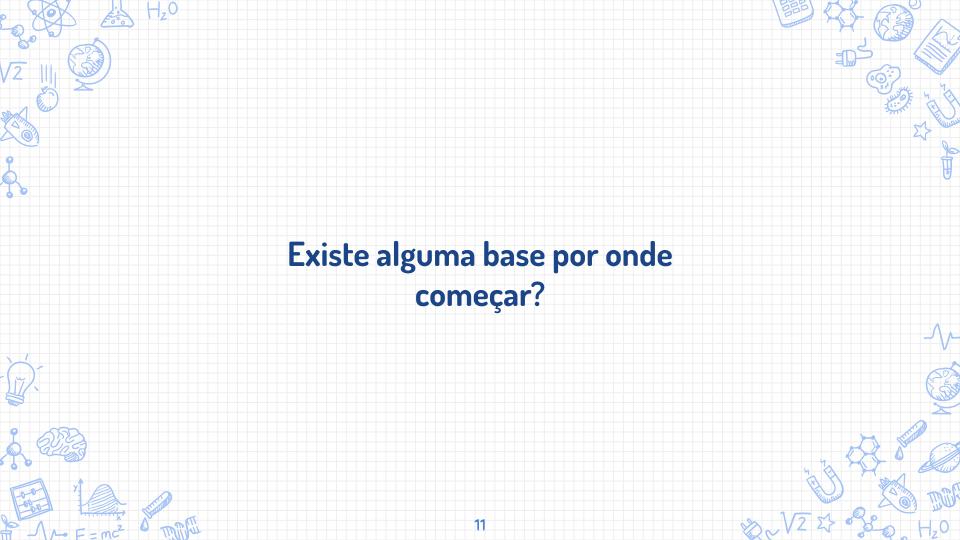
#### Escopo e Desenvolvimento.

- Será um desenvolvimento em equipe?Separar tarefas!
- **X** Definir cronogramas

Atenção a forma como cada um programa, dividir as tarefas e não conseguir juntar os códigos depois pode ser um problema!

- ✗ Escolha desenvolver funções
  - Só precisará saber o que a função faz, e não como faz!





#### O Codigo base

```
import time
from graphics import *
def Main (Titulo: str, W: int, H:int):
   win = GraphWin(Titulo, W, H, autoflush=False)
   framerate = 1/60.0
   print ("Este espaço é executado apenas uma vez ao iniciar o jogo")
   print("Use ele para iniciar as variáveis")
   #Equanto a janela não for fechada
   while win.closed == False:
       print("O While True faz isto se tornar um loop infinito")
      print ("Então agora tudo que está aqui e executado a cada frame")
       print ("Use este espaço para programar as interações com o jogador")
       print ("E também as interações entre os componentes do jogo")
       #Este comando faz a janela ser atualizada a cada frame
       win.update()
       #Este comando faz com que o framerate fique sempre abaixo de 60fps
       time.sleep(framerate)
```



## Tips and Tricks

Algoritmos para solucionar problemas pontuais



#### Movimentação

#Framerate limita a quantidade de frames por segundo

framerate = 1/60.0

circulo = Circle(Point(0, 0), 50)

#velocidade e dada em pixels/segundo
velocidade = 5

#Equanto a janela não for fechada

while win.closed == False:

#Movemos o círculo no eixo X

circulo.move(velocidade \* framerate, 0)

win.update()

time.sleep(framerate)

velocidade = 5

 $f ramerate = \frac{1}{60}$ 

 $dx = velocidade \cdot f ramerate$ 

$$dx = 5 \cdot \frac{1}{60} = \frac{5}{60}$$

Em um segundo esse comando e executado 60 vezes

Com isto temos

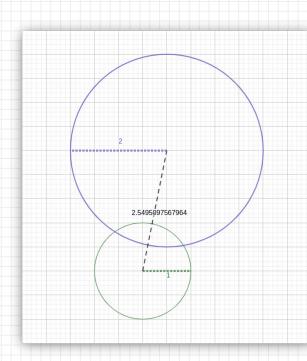
$$dx \cdot 60 = \frac{5 \cdot 60}{60}$$

$$dx \cdot 60 = 5$$



#### Intersecção e detectando colisão

```
#Equanto a janela não for fechada
while win.closed == False:
   mouse = win.checkMouse()
   if (mouse != None):
        circuloB.move(
          mouse.x - circuloB.getCenter().x,
          mouse.y - circuloB.getCenter().y)
        distanciaDosCirculos = Distancia(
          circuloA.getCenter(),
          circuloB.getCenter())
        somaDosRaios = circuloA.radius +
          circuloB.radius
        if (distanciaDosCirculos < somaDosRaios):</pre>
            label.setText("Intersecção")
        else:
            label.setText("Nada de novo sob o sol")
```

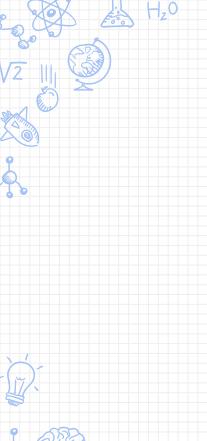




#### Animação

```
charizard = [Image(Point(600, 300), "charizard/"+str(i)+".gif") for i in range(53)]
charizardFrame = 0
charizard[charizardFrame].draw(win)
grotle = [Image(Point(150, 300), "grotle/"+str(i)+".gif") for i in range(113)]
grotleFrame = 0
grotle[grotleFrame].draw(win)
#Equanto a janela não for fechada
while win.closed == False:
    charizard[charizardFrame].undraw()
    charizardFrame = (charizardFrame + 1) % len(charizard)
    charizard[charizardFrame].draw(win)
    grotle[grotleFrame].undraw()
    grotleFrame = (grotleFrame + 1) % len(grotle)
    grotle[grotleFrame].draw(win)
```





### Qualquer dúvida é só perguntar!

wilgnne.kba@gmail.com
Atendimentos
Terças - 9hs às 12hs
Quintas - 15hs às 17hs

