

UNIVERSIDADE FEDERAL DO PARANÁ

ALEXANDRE THIAGO FERRARI TUPINAMBÁ

RELATÓRIO FINAL

PROGRAMA DE IC:

☒ **PIBIC**

☐ **PIBIC Af**

☐ **PIBIC EM**

☐ **PIBITI**

MODALIDADE:

☐ **CNPq**

☒ **UFPR TN**

☐ **Fundação Araucária**

☐ **Voluntária**

ENSINO DE PROGRAMAÇÃO COM DESENVOLVIMENTO DE JOGOS

Relatório apresentado à Coordenação de Iniciação Científica e Tecnológica da Universidade Federal do Paraná como requisito parcial da conclusão das atividades de Iniciação Científica ou Iniciação em desenvolvimento tecnológico e Inovação - Edital 20__

Orientador(a): ALEXANDRE PRUSCH ZUGE

Título do Projeto: Ensino de Programação com Desenvolvimento de Jogos

JANDAIA DO SUL

2023

RESUMO

A pesquisa tem como objetivo desenvolver um jogo digital de fins educativos, com a finalidade de utilizá-lo como base para a criação de um curso online de programação. O jogo foi criado com base na metodologia ativa utilizando a Aprendizagem Baseada em Projetos, onde os alunos assumem um papel ativo na construção do conhecimento. Durante o desenvolvimento do jogo, foram utilizadas estruturas básicas de programação aplicadas de forma simples, utilizando a linguagem Python. A escolha dessa linguagem foi feita visando facilitar a compreensão dos conceitos, e a biblioteca Pygame, uma biblioteca para programas de mídia interativa, foi utilizada para a criação de elementos no jogo. A ideia do jogo baseia-se no gênero de tower defense, onde o jogador utiliza peças de xadrez de uma cor para se defender das tropas inimigas representadas por peças de xadrez de outra cor que seguem um percurso pré-estabelecido, o jogador deve impedir que o inimigo chegue ao final. O curso será feito através de vídeos que ficarão disponíveis em uma plataforma de compartilhamento de vídeos. Através do curso, os estudantes aprenderiam os conceitos de programação enquanto produzem esse jogo, tornando o processo de aprendizagem mais dinâmico e envolvente. A proposta do curso é ensinar programação de forma prática, permitindo que os alunos apliquem os conceitos aprendidos na criação de um projeto concreto. A combinação da metodologia ativa com a utilização de um jogo educativo busca promover a participação ativa dos alunos, o desenvolvimento do pensamento crítico, a resolução de problemas e a aplicação prática do conhecimento adquirido. Essa abordagem visa tornar o aprendizado de programação mais significativo e engajador, proporcionando uma experiência de aprendizagem mais envolvente e estimulante para os estudantes.

Palavras-chave: Jogos Digitais, Desenvolvimento de Jogos, Aprendizagem Baseada em Projetos,

1. INTRODUÇÃO

O ensino de programação tem se mostrado uma habilidade cada vez mais útil no mundo atual, a sociedade depende mais de aplicativos e sites, seja para entretenimento, trabalho, organização. Apesar disso, o relatório da Associação Brasscom (2017) afirma que as empresas de tecnologia precisarão de cerca de 797 mil formados na área tecnológica entre 2021 e 2025. No entanto, o número de graduados não é suficiente para atender a essa demanda, o que resultará em um déficit de aproximadamente 106 mil talentos por ano. É possível observar que existe uma escassez de programadores no Brasil, essa demanda vem crescendo cada vez mais e por isso é importante que as pessoas aprendam a programar.

Podemos definir ciência de dados como o estudo de grandes conjuntos de dados brutos a fim de identificar padrões, e a partir destes extrair conclusões significativas que ajudem no processo decisório e na resolução dos problemas do negócio, respondendo perguntas como: o que, porque, quando e o que acontecerá. Matemática, inteligência artificial e estatística são as subáreas de estudo da ciência de dados.

O aprendizado pode ser difícil para algumas pessoas, isso pode ser justificado por diversos fatores, como região ou oportunidade, mas também porque envolve vários conceitos abstratos e algumas habilidades específicas que nem todas as pessoas possuem facilidade de adquirir.

Levando em conta essas dificuldades no aprendizado, que causam consequências diretas no déficit de programadores, e a crescente popularidade dos jogos digitais surge a síntese desta iniciação científica. O desenvolvimento de simples jogos digitais como forma de ensinar programação, levando em conta que o estudante já possui conhecimento básico em uma linguagem de programação.

O objetivo é criar um jogo que poderá ser utilizado como base de um futuro curso online. Nesse curso seria ensinado a criação desse jogo abordando a linguagem de programação Python, ensinado e reforçando conceitos de programação como programação orientada a objetos, alguns algoritmos básicos como de busca e ordenação, e o uso da biblioteca do Pygame.

2. REVISÃO DA LITERATURA

Diversas bibliografias foram utilizadas como forma de inspiração e auxílio na construção desta iniciação científica, desde o conhecimento da linguagem Python até a criação do jogo, através da biblioteca Pygame.

Uma das literaturas que contribuíram para a utilização do Python, que foi a linguagem utilizada para criar o jogo inteiro, foi a de Paiva et al (2019) realizou um livro introdutório para contribuir no ensino de Python a novos programadores. Além desta, outra bibliografia que auxiliou nesse ponto foi o curso de Python online, através do youtube, criado por Guanabara(2017).

Além do Python, outra ferramenta fundamental para o desenvolvimento da iniciação científica, mais especificamente para o jogo, foi a biblioteca Pygame a qual possui diversas funções e ferramentas para a programação de softwares de mídia interativa. A utilização desta biblioteca foi através da obra de Sweigart(2010) que em

seu livro ensina a criar jogos utilizando o Pygame, além de ensinar o básico de Python. Além deste autor o ClearCode(2022) também auxiliou na utilização da biblioteca através de seu vídeo de 3 horas publicado no youtube que tem foco somente na criação de jogos utilizando o Pygame.

Além destas bibliografias que foram fundamentais para este trabalho também houveram pesquisas relacionadas que inspiraram a criação desta.

Como exposto por Costa(2019) na revista Educitec, o desenvolvimento de jogos como forma de auxiliar no processo de ensino em programação. Após a utilização desse método, Costa revela que os estudantes conseguiram aprender o conteúdo e houve uma queda no índice de reprovação e desistência em 47% em relação ao ano anterior.

Também um artigo publicado por Carvalho, Rodriguez e Rocha (2022) avaliou o uso da aprendizagem baseada em projetos, que é uma das metodologias consideradas ativas, com foco no desenvolvimento de jogos com base em diversos artigos.

3. MATERIAIS E MÉTODOS

No projeto, para a construção do jogo, foi utilizada a linguagem Python e a biblioteca Pygame - a mais importante para o projeto, mas não a única. A biblioteca Pygame é popular na área de desenvolvimento de jogos, sendo a mesma criada com esse intuito, mas também é aplicada em animações e em outras formas de mídias interativas. O Python foi selecionado por ser uma linguagem de alto nível e de fácil aprendizado já que o objetivo do curso não é aprender a linguagem em si, mas sim estruturas gerais que constroem um *software*.

A ideia do jogo surgiu a partir de um gênero chamado *tower defense* em que o jogador deve proteger um território, ou um objetivo, de inimigos que se movem em direção a ele através de uma rota predefinida. Para fazer isso, o jogador deve construir torres defensivas que vão atirar quando o inimigo passar próximo delas. Se uma tropa inimiga alcançar o objetivo, o jogador perde uma quantidade de pontos. Assim o jogador deve colocar as torres em pontos estratégicos ao longo do caminho para atacar e destruir os inimigos antes que eles alcancem o objetivo.

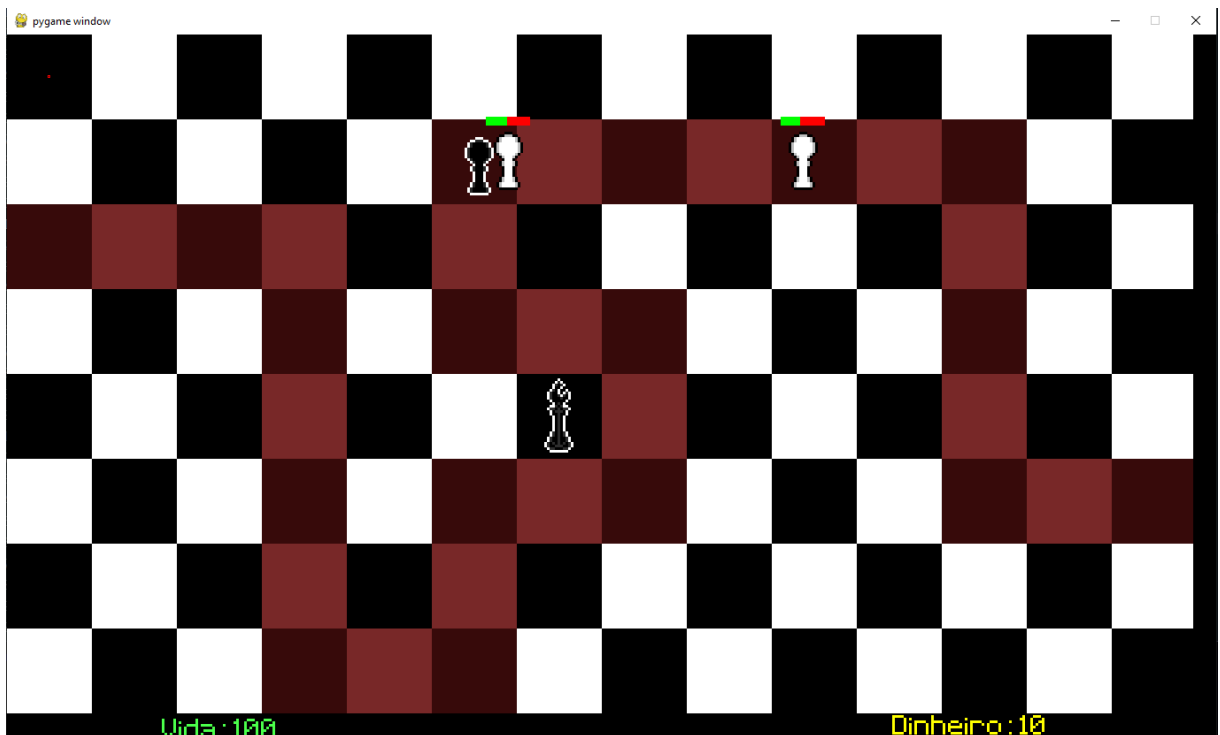
O jogo criado para o curso tem alterações contudo ainda possui as bases do gênero *tower defense*, o jogador possui peças de xadrez de uma cor e o inimigo que seria o computador possui peças de outra cor, o jogador vai posicionar as peças em

pontos específicos e o inimigo vai tentar chegar ao fim do caminho, se uma peça do jogador perceber uma peça inimiga por perto ela vai comer assim como no xadrez, por exemplo um peão comeria na diagonal e então voltaria para sua posição.

Na criação do jogo foram utilizadas diversas estruturas básicas da programação que poderão ser apresentadas de forma didática para que os estudantes possam apreender ou até reforçar seus conhecimentos ao mesmo tempo que criam um jogo

4. RESULTADOS e DISCUSSÃO

O jogo já foi finalizado se trata de uma mesa de xadrez de dimensões diferentes do usual, para se encaixar com o gênero do jogo, que é um *tower defense*. Existem diferentes inimigos e aliados, que são as peças de cor branca e preta respectivamente. É possível criar diferentes grupos de inimigos como que seguem um caminho, pré-determinado, que é vermelho e ao chegarem ao fim diminuem a vida do jogador. Quando uma peça aliada “come” a inimiga o jogador ganha dinheiro para comprar mais peças de diferentes tipos e preços



Um cronograma foi projetado para que as aulas de desenvolvimento do jogo sejam abordadas apresentando os conceitos gradualmente a partir de materiais já feitos para que os alunos avancem nas aulas sem se perder.

A Primeira aula é a apresentação do projeto e preparação para o ambiente ,a qual será abordado o objetivo do projeto para os alunos, e depois a preparação para o ambiente python, onde os alunos terão liberdade de escolher entre algumas IDEs e serão instruídos a como baixar e instalar o python, o pip e a importação e apresentação da biblioteca pygame

A Segunda aula é teórica e tem como objetivo apresentar os conceitos básicos para a criação de qualquer videogame, a utilização dos loops e como os frames dão a sensação movimento, para isso os alunos serão instigados a refletir sobre a diferença entre um filme que possui linearidade e um jogo o qual é completamente dinâmico.

Na Terceira aula a biblioteca pygame será apresentada e começará a ser abordada, apresentando aos alunos os conceitos iniciais de manipulação de imagem que a biblioteca oferece, desde a configuração inicial até a manipulação de dimensões e posições, explorando métodos eficazes para gerar um ambiente visual propício à criação. Uma vez familiarizados com a criação de janelas, os alunos serão introduzidos ao poder das variáveis do tipo "*surface*". Compreender como essas estruturas armazenam e manipulam gráficos é crucial para o desenvolvimento de aplicações gráficas mais complexas. Os alunos serão desafiados a aplicar seus novos conhecimentos, criando figuras geométricas básicas através do código¹

¹ Nota de rodapé
Os códigos
<https://github.com>

```
1  import pygame
2
3  pygame.init()
4  #inicia os metodos importados do pygame
5
6  clock = pygame.time.Clock()
7  #retorna um objeto para traçar o tempo
8  #vai ser utilizado para limitar a taxa de imagens por segundo
9
10 Tela = pygame.display.set_mode((800,600))
11
12 # pygame.display.set_mode() Cria uma tela para definir o tamanho dela
13 # coloque uma tupla nos parametros o tamanho levando em consideração o
14 # eixo x e y
15
16 Figura = pygame.Surface([100,200])
17 #esse metodo vai criar nossa primeira imagem, ele basicamente devolve uma
18 #variavel tipo surface de tamanho x = 100 e y = 200 que a gente definiu
19 #nos parametros
20
21 cobra = pygame.image.load('D:\VscodeTests\Pygame_cobra.png').convert_alpha()
22 #esse metodo cria uma variavel surface atravez de uma imagem do proprio computador,
23 #para isso o parametro deve conter o caminho até o arquivo da imagem
24
25 Figura.fill((0,0,255))
26
27 #esse metodo vai colorir completamente nossa imagem na cor azul
```

Neste código estão as explicações detalhadas para cada método básico de inicialização, criação de janela, criação de uma figura e, uma imagem do computador

```
while True:
    #para detectar os dados de entrada e saída você deve utilizar o loop de
    #eventos, nesse caso o loop esta verificando se o evento é o de apertar
    #No X para fechar a janela

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            #quando isso acontecer esse metodo finaliza o processo

    Tela.blit(Figura,(0,50))
    Tela.blit(cobra,(100,100))
    #o metodo blit esta presente nas classes screen, a variavel Tela é uma instancia
    #dessa variavel e o metodo Blit faz com que nossa imagem: Figura apareça na
    #tela, para isso, nos parametros coloque primeiro a variavel surface
    # e as cordenadas de onde você quer que ela apareça na tela

    pygame.display.flip()
    #esse metodo atualiza a variavel screen Tela

    clock.tick(60)
    #limitador de frames
```

E as documentações para os loops de eventos de entrada e saída, o método de colocar as surfaces no display e limitação da taxa de imagens por segundo

A quarta aula é uma introdução às variáveis do tipo *rectangle* que são variáveis que envolvem a imagem facilitando o posicionamento, é através dela que os métodos de colisão funcionam, além da introdução os alunos serão desafiados, através de um exercício a criar e transformar um retângulo vermelho em uma língua para uma surface com imagem de cobra

```
#o metodo scale_by recebe como parametros uma surface e um inteiro, o metodo vai aumentar
#o tamanho da imagem ,sendo x o inteiro, em x vezes. depois o metodo retorna uma variavel
#surface com o tamanho alterado
cobra = pygame.transform.scale_by(cobra,15)

#O Pygame utiliza objetos "Rect" para armazenar e manipular áreas retangulares.
#Um "Rect" pode ser criado a partir de uma combinação de valores de "left" (esquerda),
#"top" (topo), "width" (largura) e "height" (altura). Rects também podem ser criados
#a partir de objetos em Python que já sejam um Rect ou tenham um atributo chamado rect.

Figura_rect = Figura.get_rect()
cobra_rect = cobra.get_rect()
#esse metodo cria uma variavel rectangle que envolva da variavel surface
#as variaveis rectangles nos ajudam a posicionar as surfaces de forma mais eficiente
#e tambem é atravez delas que será utilizado os metodos de colisão

cobra_rect.center = (400,300)

#Exercicio Transformar o retangulo vermelho na lingua da cobra

#Resposta
#Figura = pygame.Surface((20,12)) ou Figura = pygame.transform.scale_by(Figura,0.1)
#Figura_rect = (330,243)
```

```

if cobra_rect.colliderect(Figura_rect):
    # todos as variaveis rect possuem o metodo colliderect,
    # ele é um metodo que verifica um rectangle esta colidindo
    # com o rectangle que foi passado no parametro

    print('Colisão')

```

A próxima aula tem como objetivo mostrar os métodos de entrada e saída de dados que a biblioteca possui, mas como foco nos dados dos periféricos teclado e mouse. O pygame possui métodos específicos que utilizam listas para receber as informações e um loop de eventos que a cada frame verifica todos os eventos que estão em uma lista da biblioteca

```

#Atravez desse algoritmo desse loop nos podemos obter os dados de todos os eventos que a biblioteca consegue acessar
for event in pygame.event.get():

    if event.type == pygame.QUIT:
        #O evento pygame.QUIT é o evento de quando clica no X da janela e fecha o programa Se esse for o
        # evento(clicar no X) nos iniciamos o metodo quit() que encerra o programa
        pygame.quit()
    if event.type == pygame.KEYDOWN:
        #Se o tipo de evento for uma tecla pressionada iniciamos a verificação de todas as teclas
        if event.key == pygame.K_UP:
            #Se a tecla desse evento for K_UP a cordenada x do cobra_rect vai somar com e -5 pixels
            print('Seta para cima')
            cobra_rect.center = (cobra_rect.center[0], cobra_rect.center[1]-5)
        if event.key == pygame.K_DOWN:
            #cobra_rect.center = (cobra_rect.center[0], cobra_rect.center[1]+5)
            print('Seta para baixo')
        if event.key == pygame.K_LEFT:
            #cobra_rect.center = (cobra_rect.center[0]-5, cobra_rect.center[1])
            print('Seta para esquerda')
        if event.key == pygame.K_RIGHT:
            #cobra_rect.center = (cobra_rect.center[0]+5, cobra_rect.center[1])
            print('Seta para Direita')

#Na documentação https://www.pygame.org/docs/ref/key.html é possível encontrar todos os tipos de eventos possíveis

```

```

keys = pygame.key.get_pressed()
# pygame.key.get_pressed() retorna uma lista de variaveis booleanas de todas as teclas do teclado, se uma tecla
# esta sendo pressionada a variavel dela na lista recebe o valor True
if keys[pygame.K_UP]:
    #Se a tecla pressionada for a seta para cima ele usa a posição atual dela e soma com um valor
    # e assim por diante
    cobra_rect.center = (cobra_rect.center[0], cobra_rect.center[1]-2.5)
elif keys[pygame.K_DOWN]:
    cobra_rect.center = (cobra_rect.center[0], cobra_rect.center[1]+2.5)
if keys[pygame.K_LEFT]:
    cobra_rect.center = (cobra_rect.center[0]-2.5, cobra_rect.center[1])
elif keys[pygame.K_RIGHT]:
    cobra_rect.center = (cobra_rect.center[0]+2.5, cobra_rect.center[1])

```

A Sexta aula é uma aula de revisão dos conceitos de programação orientada a objeto, mais especificamente das classes de forma a consolidar os fundamentos para fortalecer a compreensão e preparar os alunos não ficarem confusos com as classes Sprites da biblioteca pygame, métodos mais complexos para manipulação de imagem. As classes Sprites fazem a união entre as variáveis surface e rect, por

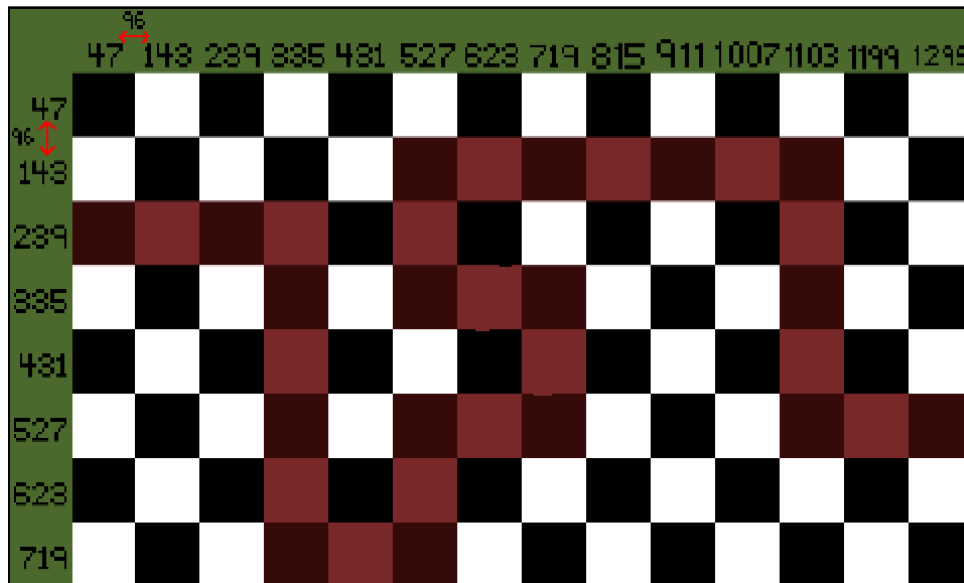
conta disso os estudantes vão reconstruir tudo que fizeram nas aulas anteriores dentro de classes Sprites

```
class cobra(pygame.sprite.Sprite):
    #como é uma classe tipo Sprite ela leva esses parametro
    def __init__(self):
        super().__init__()
        #este é o metodo de inicialização sempre que a classe for instanciada esse metodo vai ser executado então todos os metodos
        #que foram utilizados antes para criar a cobra e que não foram para o loop principal do jogo ficam nesse metodo
        self.hp = 100
        self.image = pygame.image.load('D:\VscodeTests\cobra.png').convert_alpha()
        self.image = pygame.transform.scale_by(self.image,2)
        self.rect = self.image.get_rect()
        self.rect.center = (400,300)
    def movimentação(self):
        #o metodo de movimentação que ficava no no loop principal
        if keys[pygame.K_UP]:
            self.rect.center = (self.rect.center[0],self.rect.center[1]-2.5)
        elif keys[pygame.K_DOWN]:
            self.rect.center = (self.rect.center[0],self.rect.center[1]+2.5)
        if keys[pygame.K_LEFT]:
            self.rect.center = (self.rect.center[0]-2.5,self.rect.center[1])
        elif keys[pygame.K_RIGHT]:
            self.rect.center = (self.rect.center[0]+2.5,self.rect.center[1])
    def update(self):
        #o metodo update sempre esta presente na classe mesmo que não seja criado,como esta feito ele serve para atualizar os
        # metodos que o programador necessitar
        self.movimentação()
```

Depois desta revisão será dada uma explicação sobre a diferença entre os *Group()* e *Group Single()* métodos que agrupam as classes e os alunos vão criar outra classe, Jacaré, para ser como um inimigo e testar os métodos de colisão entre as classes do tipo sprite. Se houver colisão entre as classes o Jacaré desaparece e ao apertar R uma nova instância é criada

```
def update(self):
    #o metodo update sempre esta presente na classe mesmo que não seja criado,como esta feito ele serve para atualizar os
    # metodos que o programador necessitar
    pygame.sprite.groupcollide(ClasseCobra,ClasseJacaré,False,True)
    #esse metodo da biblioteca recebe 4 parametros dois grupos de sprites e o metodo verifica se há colisão
    #se houver ele pode matar um dos sprites, os dois ou nenhum deles, depende do 3º e 4º parametro
    #se o 3º parametro tiver valor True ele mata o o grupo de sprites que foi informado no primeiro parametro
    #se o 4º parametro tiver valor True ele mata o o grupo de sprites que foi informado no segundo parametro
    self.movimentação()
```

A partir da Sétima aula os estudantes vão estar preparados para iniciar o projeto de criação do jogo idealizado, nessa aula será apresentando a ideia de um jogo tower defense que como diferencial utiliza peças de xadrez, e como será a arte do jogo, começando pelo desenho de mapa inicial e os desenhos das peças de xadrez. Será recomendado aos alunos que eles mesmos façam a arte para o jogo, e que seja feito em baixa resolução para que se torne mais fácil o desenho das peças



A construção do caminho será feito utilizando essas coordenadas, todos os centros dos quadrados que formam o caminho devem estar em uma lista para cada mapa, a distância entre os centros é de 96. A lista vai é o que dita o caminho para as futuras peças.

Na Oitava aula os métodos básicos como inicialização dos módulos, criação de variáveis tipo display, evento de encerramento, serão inicializados e todo ambiente básico será configurado. A partir dessa aula os alunos serão instigados a refletir e serão ativos na criação do projeto, enquanto o professor vai auxiliando e guiando os pensamentos dos estudantes. Então a classe peão inimigo será desenvolvida junto com o método de movimentação da peça, a partir dos dados do mapa que foram criados.

```
# O metodo verifica a posição atual e a posterior e verifica se ele precisa
# aumentar o X ou o Y para atingir o proximo estagio
def caminhoPeao(self,rect):
    if rect == pontos[self.stg]:
        self.stg+=1
    #Quando a posição atual é igual a do destino soma-se mais 1 na variavel estagio

    if self.stg < len(pontos):
        #stage/ Cada ponto da lista é um estagio, quando chega ao final ele é deletado\
        pos=[rect[0],rect[1]]
        #ele verifica a posição atual e compara com a posição destino, dependendo da coordenada o metodo
        #soma ou subtrai a coredenada X ou Y até que se torne igual a do destino
        if pos[0] < pontos[self.stg][0]:
            pos[0] += self.speed
        if pos[0] > pontos[self.stg][0]:
            pos[0] -= self.speed
        if pos[1] < pontos[self.stg][1]:
            pos[1] += self.speed
        if pos[1] > pontos[self.stg][1]:
            pos[1] -= self.speed
        return (pos[0],pos[1])
    else:
        self.kill()
        return(-47,-47)
```

Na Nona aula os estudantes vão criar uma classe e o método de movimentação para um peão aliado seguindo a mesma metodologia da aula anterior com o professor instigando e guiando os alunos enquanto eles vão sugerindo seus pensamentos sobre o algoritmo, se baseando nos dados do mapa e do caminho que é percorrendo pelo peão inimigo.

A Decima aula é uma continuação da oitava, a qual será desenvolvido um método para detectar os inimigos e iniciar a movimentação do peão branco no momento que o inimigo for detectado, por conta disso o método de movimentação será aprimorado.

Na décima primeira aula outra peça inimiga será desenvolvida, uma torre, funcionará de forma semelhante a um peão, contudo ele percorre uma reta e ao chegar no último ponto ele aguarda uns instantes e vai até o fim da outra reta, e então outro método. Além disso, nessa aula também será desenvolvido uma torre aliada com método de movimentação e detecção.

A última peça será desenvolvida na décima segunda aula, uma bispo aliado seguindo a metodologia das outras aulas. Em seguida, será abordado a criação de um menu para seleção das peças, a qual os estudantes terão de usar métodos de busca. E então finalizando com a criação dos diferentes estados do jogo, Tela inicial, Menu, Pause...etc.

5. CONSIDERAÇÕES FINAIS

Através da construção do jogo, é esperado os estudantes não apenas adquiriram habilidades técnicas valiosas, mas também desenvolveram uma mentalidade analítica e lógica que são essenciais para qualquer programador. É almejado que o processo de planejar, projetar e implementar um jogo promova o incentivo a resolução criativa de problemas e o pensamento crítico, habilidades que são fundamentais no mundo da programação.

E com a finalização do jogo e dos materiais de ensino é previsto que este projeto seja completamente finalizado e implementado, através dos vídeos do curso online, pelo autor

6. REFERÊNCIAS

COSTA, J. B. Desenvolvimento de uma arquitetura de game para auxiliar o processo de ensino-aprendizagem em programação de jogos digitais no curso técnico de informática. Separata de: **Edutec - Revista de Estudos e Pesquisas sobre Ensino Tecnológico**, Manaus, Brasil, v. 5, n. 10, 2019.

CARVALHO, Walter R. B.; RODRIGUEZ, Carla L.; ROCHA, Rafaela V.. Aprendizagem Baseada em Projetos no Contexto do Desenvolvimento de Jogos: uma Revisão Sistemática de Literatura. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 33. , 2022, Manaus. **Anais do simpósio brasileiro de informática na educação**. Porto Alegre: Sociedade Brasileira de Computação, 2022. p. 267-277.

CLEARCODE. **The ultimate introduction to Pygame**, 2022. 1 vídeo (3 hrs). Disponível em: <https://www.youtube.com/watch?v=AY9MnQ4x3zk>. Acesso em: 28 abril. 2023.

GUANABARA, G. **Curso Python #01 - Seja um Programador**, 2022. 1 vídeo (29 min). Disponível em: <https://www.youtube.com/watch?v=S9uPNppGsGo>. Acesso em: 28 abril. 2023.

GALLINDO, S.P.; OLIVEIRA, M; Demanda de Talentos em TIC e Estratégia Σ TCEM. **Relatório de inteligência e Informação**, dez. 2021. Disponível em: <https://brasscom.org.br/pdfs/demanda-de-talentos-em-tic-e-estrategia-tcem/> Acesso em: 28 abr. 2023.

PAIVA, F. A. P. *et al.* **Introdução a Python com aplicações de sistemas operacionais**. Natal-RN: Editorairfn, 2020.

SWEIGART, A. **Invent Your Own Computer Games with Python**. 4. ed. San Francisco: No Starch Press, 2017.