Space Battle

Maria Clara de Souza Informática Instituto Federal Catarinense Timbó mariaclaratimbo3@gmail.com

Lucas Eduardo Kreuch
Informática
Instituto Federal Catarinense
Blumenau
Iucaskreuch77@gmail.com

Haidy Jandre
Informática
Instituto Federal Catarinense
Blumenau
haidyjandree10@gmail.com

Davi Gabriel Krueger Informática Instituto Federal Catarinense Blumenau davigk13@gmail.com

1 Proposta

Pretendemos fazer um jogo 2D com um tipo de sistema em score utilizando Python 3, inspirado em R-Type III. O jogo terá uma fase única onde a dificuldade irá aumentar de acordo com a progressão do jogador.

2 Justificativa

Decidimos fazer um jogo 2D em Python pois é um estilo que todos do grupo gostam, além de que poderemos obter conhecimento ao decorrer do desenvolvimento da programação do jogo, sendo também um game simples que qualquer um pode aprender a jogar sem muito tutorial.

Jogos 2D fizeram parte da ocupação e pavimentação da indústria de games atual, sendo o iniciador de quaisquer gêneros. É um jogo que usa das imagens em pixels, ou não, para constituir os elementos do jogo, como pessoas, cenários e objetos. Ao fazer esse tipo de jogo precisa-se ter um rigoroso cuidado na hora de desenvolver o scroll de tela de fundo, pois é normal encontrar um jogo com breves falhas na ligação de uma ou mais imagens que fazem parte do background.

Um exemplo de jogo pixelado que utilizaremos como inspiração é o jogo R-Type 3, sendo um vídeo jogo arcade de tiro bidimensional de nave, com uma resolução

gráfica de 384 x 256 pixel e uma paleta de 512 cores. O jogo fala sobre a história da nave R-9, que em busca de paz para a raça humana, luta contra o império Bydo. De acordo com o site Wikipédia (2020): "A série R-Type é lembrada pelos veteranos jogadores de vídeo game como um dos mais belos jogos de tiro já lançados para o vídeo game, com sua concepção futurista original e estilo único."

Nosso jogo irá utilizar o mesmo estilo pixel art retrô de R-Type 3, com a mesma base de um game de nave (sendo o player) que vai destruir outras naves inimigas em sistema de score, sem história. De acordo com a progressão do jogador e os pontos/dinheiros adquiridos durante o jogo será possível comprar naves e fazer atualizações melhores nas mesmas.

3 Método de Trabalho

Passo 1: Decidir o que iremos desenvolver no projeto (se iremos desenvolver um jogo, um site, entre outros);

Passo 2: Conversar sobre como irá ser o estilo do jogo (pixelado, em sistema de score, compras dentro do jogo, personagens, entro outros);

Passo 3: Exercer pesquisas de outros games como inspiração para a ideia do desenvolvimento do jogo;

Passo 4: Organizar tecnologias e aplicativos que iremos recorrer (linguagens de programação como python, bibliotecas como o pygame, site para pixel art como o piskel, aplicativos, entre outros);

Passo 5: Fazer a divisão de tarefas para cada integrante do grupo;

Passo 6: Criar as sprites do jogo (design);

Passo 7: Pesquisar sobre as mecânicas do jogo (pesquisas sobre como irá funcionar o sistema de tiro, movimentação, inimigos, colisão, entre outros);

Passo 8: Começar a programar a base do jogo;

Passo 9: Programar a lógica do jogo;

Passo 10: Colocar sons ao jogo;

Passo 11: Fazer a serialização (transformar o estado de um objeto em uma sequência byte);

Passo 12: Arrumar os bugs do jogo e limpar o código;

4 Materiais e Equipamentos Necessários

Python 3; Piskel;

Visual Studio Code:

Pycharm;

Pygame;

Computador.

5 Como os materiais/equipamentos serão obtidos?

Os materiais que utilizaremos serão os computadores que o IFC disponibiliza nas salas de informática e os nossos próprios equipamentos (notebook, celular, entre outros). As linguagens de programação, aplicativos e sites que utilizaremos serão baixados nos mesmos.

6 Cronograma

O cronograma de atividades está disponível no Apêndice I.

7 Observações

Iremos utilizar o piskel para fazer as sprites do jogo;

O Visual Studio Code ou o Pycharm para programar o jogo;

A biblioteca pygame do Python (que deverá ser instalada nos computadores da escola);

Link do nosso repositório do github https://github.com/Luc4sKr/Desenvolvimento-de-projeto-IFC

Referências

- [1] Patricia S. Abril and Robert Plant, 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan, 2007), 36-44. DOI: https://doi.org/10.1145/1188913.1188915.
- [2] Sten Andler. 1979. Predicate path expressions. In Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79). ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774
- [3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.
- [4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..

Apêndice I - Cronograma

Ações	m	а	m	j	j	а	S	0	n	d
	a r	r	a i	u n	u -	g, o	e t	u t	0	e z
Decidir o que iremos desenvolver no projeto:	Х									
Conversar sobre como irá ser o estilo do jogo:	Х									

Exercer pesquisas de outros jogos como inspiração para a ideia do desenvolvimento do jogo:	Х	Х								
Organizar tecnologias e aplicativos que iremos recorrer:	Х	Х								
Fazer a divisão de tarefas para cada integrante do grupo:	Х	Х								
Criar as sprites do jogo:		Х	Х	Х	Х					
Pesquisar sobre as mecânicas do jogo e como exercer:		Х	Х	Х	Х					
Começar a programar a base do jogo:		Х	Х							
Programar a lógica do jogo:			Х	Х	Х	Х	Х			
Colocar sons ao jogo:						Х	Х	Х		
Fazer a serialização:						Х	Х			
Arrumar os bugs do jogo e limpar o código:								Х	Χ	