**Licenciatura em Engenharia Informática**

Escola Superior de Tecnologia e Gestão

*Instituto Politécnico de Viana do Castelo*

Unidade curricular de

Tecnologias Multimédia

2023/2024

*TP2 – Pesca Magica*

https://github.com/paulino25cc/TP2-32553-29375

Francisco Santos Nº 29375

Paulino Cerqueira Nº 32553

# Apresentação do Jogo

O jogo desenvolvido consiste numa espécie de jogo de apanhar peixes ao qual demos o nome de “Pesca Magica” desenvolvido através do Phaser. Resumidamente, o utilizador controla um isco e tem como objetivo apanhar peixes para conseguir moedas para melhorar os iscos e aumentar os boosts, os boosts servem para conseguir um melhor score.

Uma imagem com recife, Gráficos, design gráfico, desenho

Descrição gerada automaticamente

Quando entramos no jogo, o jogo é iniciado com uma música para tornar o jogo mais animado.

Uma imagem com captura de ecrã

Descrição gerada automaticamente

A musica pode ser controlada na barra verde que se encontra no fundo da tela.

O jogo em si conciste em apanhar peixes para aumentar as moedas e aumentar o score para fazer o melhor score possivel. Durante o jogo vao aparecendo peixes dos dois lados que valem 10 pontos e 3 moedas, aparecem lulas com movimento aleatorio e velocidade aleatoria que valem 50 pontos e 5 moedas, tambem aparecem tubaroes dos dois lados. Quando estamos com o anzol normal se apanharmos o tubarao, aperece a mensagem de “Game Over ” em seguida podemos reiniciar o jogo mas o score seria resetado, apenas o score.

Uma imagem com captura de ecrã, desenho

Descrição gerada automaticamente

Porem ao longo do jogo tambem aparece uma estrela aleatoriamente no mapa que é estatica mas só permanece no ecra durante 3 segundos se conseguirmos apanhar a estrela o isco transforma-se numa rede essa rede pode apanhar tudo até os tubaroes se apanhar o tubarao o score aumenta 100 e fornece 10 moedas. A durançao da rede inicialmente é de 5 segundos mas pode ser aumentado no menu de boosts. No lado superior esquerdo podemos ver quanto tempo o utilizidar tem ainda com a rede para não ser apanhado de surpresa por um tubarao no momento em que volta a ser um isco normal.

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Durante o jogo se pressionarmos a tecla “ESC” abre um menu de pausa onde todo o jogo é pausado.

Uma imagem com captura de ecrã

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Se pressionarmos a tecla “R” da restart ao jogo o que significa que da restart ao score, mas as moedas continuam igual porque as moedas é como objetivo externo.

Se pressionarmos a tecla “B” abre um menu de boosts onde se pode aumentar a velocidade do isco ou aumentar o tempo de duração da rede. Esses boosts têm no máximo nivel 10.



Se clicarmos no + o nível a frente aumentará.

Uma imagem com texto, captura de ecrã, aeronave

Descrição gerada automaticamente

Se pressionarmos a tecla “S” abre um menu de shop que podemos comprar novos iscos os novos iscos têm uma velocidade maior o que torna o jogo mais fácil.

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Na imagem a baixo podemos o novo isco.

Uma imagem com captura de ecrã

Descrição gerada automaticamente

# Scripts

No desenvolvimento do nosso jogo que adicionam funcionalidades e comportamentos aos objetos do jogo, tornando-os interativos e dinâmicos. O script está dividido em várias funções, funções essas que vou explicar agora.

**Criação de variáveis**

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

O código fornecido configura variáveis que são declaradas para controlar diferentes aspetos do jogo. As variáveis score e coins mantêm a pontuação do jogador e o número de moedas coletadas, respetivamente. As variáveis como “scoreText”, “coinsText”, “gameOverText”, “pauseText”, “netTimerText”, “shopText” e “boostText” são usadas para armazenar elementos de texto que serão exibidos no ecrã.

Existem variáveis booleanas para controlar o estado do jogo, como “isPaused”, “isGameOver”, “isShopOpen”, “isBoostOpen”, e “isNetActive”, que indicam se o jogo está pausado, se terminou, se a loja está aberta, se o menu de boost está aberto e se a rede está ativa, respetivamente.

Outras variáveis controlam a velocidade e comportamento de objetos no jogo, como “hookSpeed”, “fishSpeedL”, “fishSpeedR”, “lulaSpeed”, “sharkSpeed”, “sharkSpawnDelay”, “sharkTimer” e “netTime”, que define a velocidade do anzol, dos peixes, da lula, do tubarão, o tempo de spawn do tubarão, o temporizador do tubarão e o tempo da rede.

A variável “currentVolume” controla o volume atual da música de fundo “currentHook” armazena as propriedades do anzol atual, como textura, escala e velocidade.

Os arrays “shopItemsConfig” e “boostItemsConfig” configuram objetos da loja e boosts disponíveis no jogo, definindo propriedades como nome, custo, velocidade, textura, escala, efeito, nível inicial e nível máximo desses itens.

**Preload**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

O código na função preload () carrega todos os recursos necessários antes de o jogo começar. Ele carrega uma fonte personalizada, várias imagens (como o fundo do mapa, o pescador, peixes, lula, anzóis, tubarões, moeda, estrela e rede) e uma música de fundo. Esses recursos são carregados a partir do diretório 'assets' e são associados a chaves para facilitar a referência durante o jogo, garantindo que tudo esteja disponível sem atrasos quando o jogo iniciar.

**Create**

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

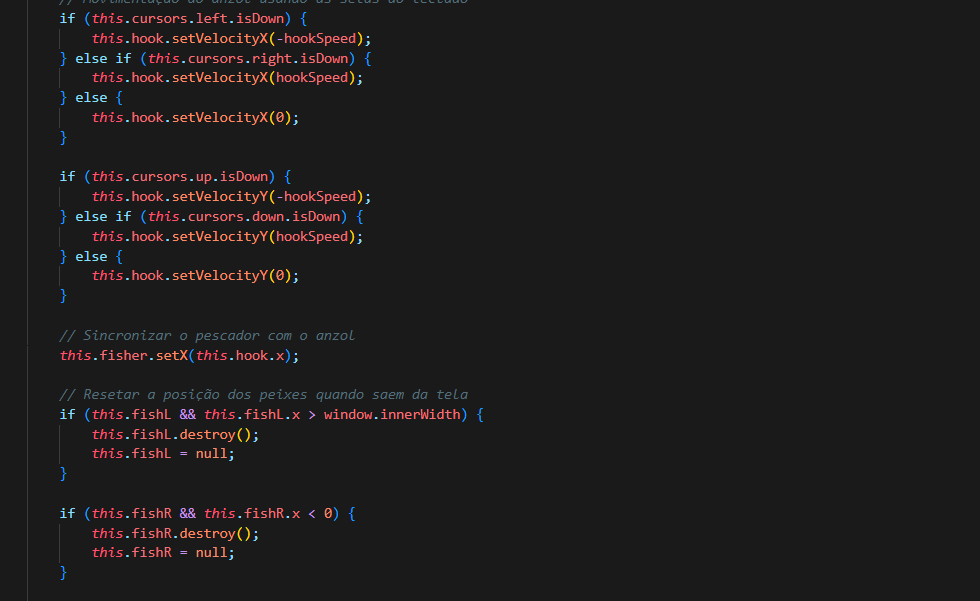
Descrição gerada automaticamente

O código na função create () configura os elementos visuais e a lógica inicial do jogo. Ele adiciona o fundo ao centro da tela e ajusta sua escala para se adaptar à janela. Em seguida, adiciona o sprite do pescador no topo da tela, configurado para não colidir com os limites do mundo, e adiciona a imagem da moeda perto do canto superior direito da tela. O sprite do anzol também é adicionado e configurado para colidir com os limites do mundo. Textos para a pontuação e quantidade de moedas são adicionados no topo da tela, com fontes e cores específicas. Textos de "Game Over" e "Pausa" são adicionados, mas inicialmente invisíveis. As teclas de cursor e outras teclas (ESC, R, S, B) são configuradas para pausar, reiniciar o jogo, abrir a loja e abrir o menu de boosts. Temporizadores são configurados para adicionar peixes, lulas e tubarões ao jogo em intervalos regulares. Por fim, um evento é adicionado para ajustar os elementos do jogo quando a janela é redimensionada. Este código configura todos os elementos necessários para o início do jogo, como fundo, personagens, textos e controles, além de configurar eventos periódicos para adicionar novos elementos ao jogo.

**Update**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente



O código na função update() gerência a lógica de atualização contínua do jogo.

Primeiro, a função verifica se o jogo está pausado, se a loja está aberta, se o menu de boosts está aberto ou se o jogo acabou. Se qualquer uma dessas condições for verdadeira, o tempo do jogo é pausado, a física do jogo é pausada e os textos correspondentes (pausa, loja, boosts, game over) são tornados visíveis conforme necessário. Se nenhuma dessas condições for verdadeira, o tempo e a física do jogo são retomados e todos os textos (pausa, loja, boosts, game over) são tornados invisíveis.

Em seguida, a função verifica as entradas do teclado para mover o anzol. Se as setas esquerda ou direita estiverem pressionadas, o anzol se move horizontalmente conforme a velocidade do anzol (hookSpeed). Se as setas para cima ou para baixo estiverem pressionadas, o anzol se move verticalmente. Se nenhuma tecla estiver pressionada, a velocidade do anzol é definida como zero, ou seja, permanece estático.

O pescador é sincronizado com a posição do anzol horizontalmente. Além disso, a função verifica se os peixes saem da tela. Se um peixe que se move para a esquerda sair da tela pela direita, ou se um peixe que se move para a direita sair da tela pela esquerda, ele é destruído.

**“Catches”**



Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

A função catchFish () desativa o peixe capturado e verifica seu tipo para adicionar moedas e pontos, atualizando os textos de moedas e pontuação, e ajustando a dificuldade do jogo. A função catchTubarao () verifica se a rede está ativa; se estiver, adiciona moedas e pontos, se não estiver, termina o jogo, pausando tudo e exibindo o texto de "Game Over". A função catchEstrela () desativa a estrela capturada, ativa uma rede temporária que protege o jogador, atualiza um temporizador na tela e, após o tempo se esgotar, restaura o anzol ao estado original e desativa a rede.

**“Spawns”**

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

As funções de spawn geram diferentes tipos de objetos no jogo em posições aleatórias e com comportamentos específicos. spawnFishL gera um peixe que se move da esquerda para a direita, enquanto spawnFishR gera um peixe que se move da direita para a esquerda. spawnEstrela gera uma estrela em uma posição aleatória que desaparece após 3 segundos se não for capturada. spawnLula gera uma lula que se move horizontalmente em uma direção aleatória, e spawnShark gera um tubarão que pode vir da esquerda ou da direita. Cada função também define colisões entre o anzol e os objetos, chamando funções específicas (catchFish, catchEstrela, catchTubarao) quando ocorrem colisões.

**Aumento de dificuldade**

Uma imagem com texto, captura de ecrã, ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

As funções de aumento de dificuldade ajustam o jogo com base na pontuação do jogador. checkScoreAndIncreaseDifficulty () verifica a pontuação e, se for igual ou superior a 100 e múltipla de 100, chama increaseSpeed () para aumentar a velocidade dos peixes, lula, tubarões e anzol em 10 unidades, e chama increaseSharkFrequency () para reduzir o intervalo de tempo para a aparição dos tubarões, tornando-os mais frequentes. Se um temporizador de tubarão já estiver ativo, ele é removido antes de definir um novo intervalo reduzido.

**Compras (iscos e boosts)**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

As funções de compra e incremento de boosts gerenciam a aquisição de itens e a aplicação de melhorias no jogo. A função purchaseItem () permite ao jogador comprar um item se tiver moedas suficientes, atualizando a textura, escala e velocidade do anzol, e fechando a loja após a compra. A função incrementBoost () permite aumentar o nível de um boost, aplicando seu efeito se o jogador tiver moedas suficientes e o boost não estiver no nível máximo. As funções increaseHookSpeed () e increaseNetTime () aumentam a velocidade do anzol em 15 unidades e o tempo da rede em 0.5 segundos, tornando o jogo progressivamente mais interessante.

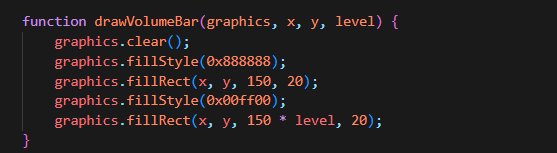
**Controlo de estados**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

As funções togglePause (), toggleShop () e toggleBoosts () controlam os estados de pausa, abertura da loja e abertura do menu de boosts no jogo. A função togglePause () inverte o valor de “isPaused” para pausar ou despausar o jogo. A função toggleShop () inverte o valor de “isShopOpen” para abrir ou fechar a loja. A função toggleBoosts () inverte o valor de “isBoostOpen” para abrir ou fechar o menu de boosts. Essas funções permitem ao jogador controlar facilmente esses estados durante o jogo.

**Barra de volume**



A função drawVolumeBar () desenha uma barra de volume na tela. Primeiro, limpa qualquer desenho anterior. Depois, desenha um retângulo cinza como fundo da barra na posição especificada por x e y. Em seguida, desenha um retângulo verde sobre o fundo, com a largura proporcional ao nível do volume. Essa função visualiza graficamente o nível de volume atual, facilitando ao jogador ajustar o som no jogo.

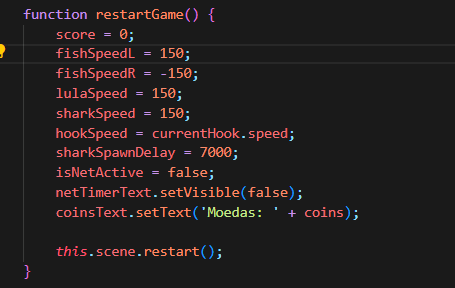
**Redimensionar o jogo**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

A função resizeGame () ajusta o tamanho do jogo quando a janela do navegador é redimensionada. Define a largura e a altura do canvas para corresponder à largura e à altura da janela do navegador, especificando os valores em pixels. Por fim, redimensiona o jogo para se ajustar às novas dimensões da janela. Isso garante que o jogo se mantenha corretamente dimensionado e visível, independentemente do tamanho da janela do navegador.

**Restart**



A função restartGame () reinicializa o jogo, configurando a pontuação para 0 e as velocidades dos peixes, lula, tubarão e anzol para seus valores iniciais. O intervalo de spawn do tubarão é reiniciado para 7000 milissegundos, e o estado da rede é definido como falso, ocultando o texto do temporizador da rede. O texto das moedas é atualizado para refletir o número atual de moedas. Por fim, a cena do jogo é reiniciada, recarregando todos os elementos e reiniciando o jogo. Isso permite que o jogo comece novamente com as configurações iniciais.

# Conclusão

Este trabalho consistiu no desenvolvimento de um jogo com a biblioteca Phaser.js, destacando a estrutura do código e as funções principais que controlam diversos aspetos do jogo.

O jogo é controlado ou gerado através de funções bem definidas que lidam com a lógica de inicialização (preload e create), atualização contínua (update), e manipulação de eventos específicos, como capturar peixes, tubarões e estrelas, além de funções para compra e incremento de boosts. Funções auxiliares, como resizeGame, garantem que o jogo se adapte a diferentes tamanhos de tela, proporcionando uma melhor experiência de jogo.

As funções de troca de estado, como togglePause, toggleShop e toggleBoosts, e funções de reinicialização, como restartGame, demonstram como o jogo oferece uma interface interativa. Além disso, as funções de desenho, como drawVolumeBar, contribuem para a visualização de informações importantes, como o nível de volume, de maneira intuitiva.

Em suma, este trabalho destacou a importância de uma estrutura de código clara e modular em desenvolvimento de jogos. Cada função desempenha um papel crucial na criação de uma experiência de jogo fluida e envolvente. Ao entender essas funções, é possível apreciar a complexidade e a atenção aos detalhes necessários para desenvolver um jogo interativo e divertido com Phaser.js.