

GAME DESIGN DOCUMENT

ESPACIAL

Maury Lukas Oliveira Freitas

MERCADO ALVO

O subgênero Shoot'em Up de jogos foi criado por Tomohiro Nishikado com o jogo Space Invaders nos arcades japoneses em 1978. Muito popular nas décadas de 80 e 90, o gênero consiste, geralmente, em controlar uma aeronave ou espaçonave no céu ou no espaço e atirar em quase tudo o que se move, enquanto o fundo 2D da tela se move na horizontal ou vertical, acumulando pontos e desviando dos perigos até o jogador não puder mais, devido à dificuldade e, conseqüentemente, perda de vidas.

Atualmente, não há mais muitos representantes desse subgênero e a possibilidade de implementar um jogo assim em uma plataforma web, tornando-o mais facilmente acessível, pois independe do sistema operacional ou hardware do usuário, possível de jogar on-line ou off-line, em qualquer lugar do mundo, aumenta a probabilidade de maior aceitação por parte dos jogadores, especialmente os casuais.

TECNOLOGIA E RECURSOS

O HTML5 é a quinta versão da Hypertext Markup Language, uma linguagem para estruturação e apresentação de conteúdo para a World Wide Web. Esta nova versão traz importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade. Possibilita o uso de novos recursos antes possíveis apenas com a aplicação de outras tecnologias.

Em particular, HTML5 adiciona várias novas funções sintáticas. Elas incluem as tags de video, audio, e elementos canvas, destinados a delimitar uma área para renderização dinâmica de gráficos. Todo o trabalho de criação e animação é realizado através de linguagens de programação dinâmica (geralmente Javascript).

JavaScript é uma linguagem script de programação interpretada com orientação a objetos baseada em protótipos, tipagem fraca e dinâmica e funções de primeira classe. Possui suporte à programação funcional e apresenta recursos como fechamentos e funções de alta ordem, comumente indisponíveis em linguagens populares como Java e C++. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade de passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

As figuras foram encontradas gratuitamente na internet e a programação e criação dos arquivos podem ser feitas desde em um simples editor de texto, como o Bloco de Notas, até um avançado ambiente de desenvolvimento, como o NetBeans.

DESCRIÇÃO DO PRODUTO

O jogador controlará a espaçonave (Fig. 1) de um simpático alienígena ajudando-o a sobreviver o máximo que puder, enquanto desvia e atira em asteroides. Conforme acerta e destrói os asteroides, acumula pontos, consequentemente o jogo vai ficando mais difícil, com mais asteroides na tela, porém sem chegar a ser impossível e frustrante de se jogar.



Fig. 1: Espaçonave

Em um canvas de 975 pixels de largura por 600 de altura, o fundo da tela, representando o espaço sideral, sorteado entre duas opções (Fig. 2 e Fig. 3), para não ser muito repetitivo, sobe 1 pixel de baixo para cima a cada frame do navegador. Descer 1 pixel, nos testes, causava uma sensação incômoda na visão de alguns testadores ao parar de jogar.



Fig. 2: Opção de fundo 1



Fig. 3: Opção de fundo 2

A espaçonave, de 100 pixels de largura por 51 de altura, sempre começa o jogo no centro do topo do canvas e só pode se locomover para a esquerda ou para a direita, 5 pixels por frame, sem nunca sair pelas bordas laterais do canvas.



Fig. 4: Tiro

O tiro (Fig.4), de 14 pixels de largura por 19 de altura, só é disparado para baixo em linha reta, locomovendo-se a 5 pixels por frame. Após um tiro, o próximo só pode ser disparado depois de 20 frames, para evitar uma linha de tiros consecutivos. E apenas 3 tiros por vez na tela podem ser disparados. Isso faz com o que o jogo enfatize a precisão nos disparos, em vez de se atirar negligentemente.

Os asteróides (Fig. 5), de 50 por 50 pixels, são gerados aleatoriamente na parte de baixo do canvas e sobem, girando, a uma velocidade que varia de 4 a 6 pixels por frame, mais do que isso, nos testes, ficavam quase impossíveis de desviar ou destruir. Quanto mais pontos o jogador acumular, mais asteróides serão gerados para dificultar.



Fig. 5: Asteróides

No início, a cada 15 pontos a mais, aumenta 1 asteróide extra na tela, para se certificar de que o jogador está aprendendo a controlar, desviar, mirar e disparar. Depois, o intervalo de pontos entre os aumentos sucessivos de asteroides diminui (a cada 10 pontos a mais, a cada 5 a mais) até o limite de 11 asteróides por vez na tela, mais do que isso, nos testes, ficava praticamente impossível, com o foco passando a ser apenas desviar até morrer, o que rapidamente acontecia.

Uma explosão (Fig. 6), de 64 por 64 pixels, acontecerá no local exato, toda vez que um asteroide se chocar com um tiro ou com a espaçonave. Sempre que um tiro destruir um asteroide, há 5 % de chance de um bônus aparecer no lugar do asteroide e continuar o trajeto que ele faria, na velocidade de 5 pixels por frame, para poder ser coletado pela espaçonave.

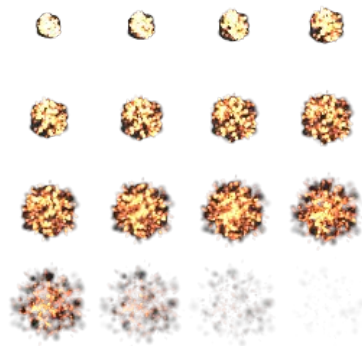


Fig. 6: Explosão

O bônus pode ser tanto uma vida, no formato de um coração vermelho (Fig. 7) de 60 pixels de largura por 50 de altura, quanto um escudo, que, se coletado, dura 30 segundos ou aguenta 3 impactos de asteroides, no formato de uma esfera azul (Fig. 8), de 50 por 50 pixels. Caso o escudo seja coletado, a espaçonave fica com um contorno azul (Fig. 9) de 5 pixels ao redor dela enquanto estiver ativo, para que o jogador possa identificar facilmente.



Fig. 7: Vida



Fig. 8: Escudo



Fig. 9: Contorno

Tudo o que pode aparecer mais de uma vez ao mesmo tempo na tela, isto é, tudo exceto a espaçonave, o fundo e o contorno, é armazenado em vetores, pois na implementação, e nos testes, foi mais fácil e prático verificar as colisões, através de recursos da programação chamados laços, além de não se necessitar destruir e recriar as imagens, poupando processamento e memória, bastando reutilizar as ociosas alocadas nos vetores.

A quantidade de vidas que o jogador tem é informada no lado superior esquerdo do canvas (Fig. 10). E a contagem de pontos acumulados, no lado superior direito (Fig. 10). O controle por frames do navegador é feito com o recurso `requestAnimationFrame` da linguagem JavaScript que não sobrecarrega o processador, gera a animação muito mais suavemente, a taxa de frames é constante e consistente, sincroniza com a taxa de atualização dos monitores e, quando a aba do navegador perde o foco atual, pausa a animação, poupando o processador.

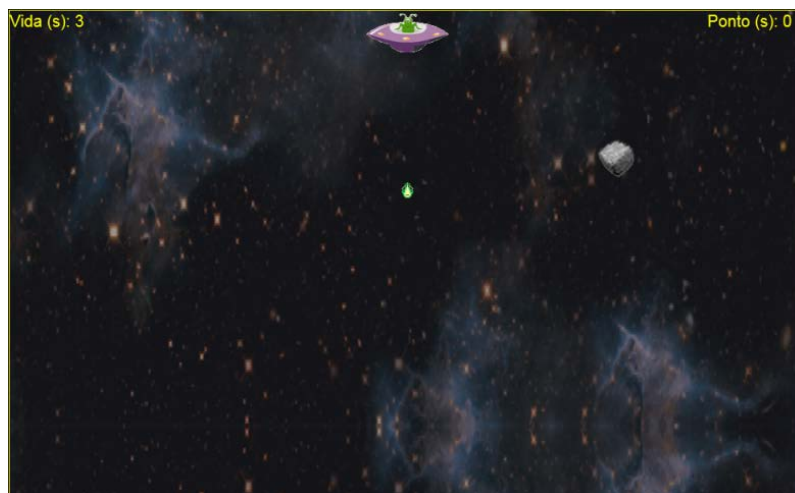


Fig. 10: Jogo em execução

Haverá sons dos tiros disparados, das explosões, quando acontecerem, e uma música de ritmo rápido, eletrizante, impactante e emocionante como a trilha sonora dos recentes filmes de Jornada nas Estrelas. Haverá um botão de Play/Pause (Fig. 11 e Fig. 12), abaixo do canvas, para que o jogador possa parar e continuar quando quiser. A música também pára ao pausar.



Fig. 11: Botão Play

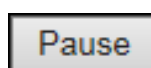


Fig. 12: Botão Pause

Sempre se começa com 3 vidas e, ao morrer, o jogo pausa e aparece “Game Over” (Fig. 13) no centro do canvas. Assim, se o jogador quiser, pode capturar a exibição da tela, em forma de imagem, com a quantidade de pontos que conseguiu atingir, e compartilhá-la em redes sociais para mostrar aos amigos e desafiá-los a fazer melhor.



Fig. 13: Game Over

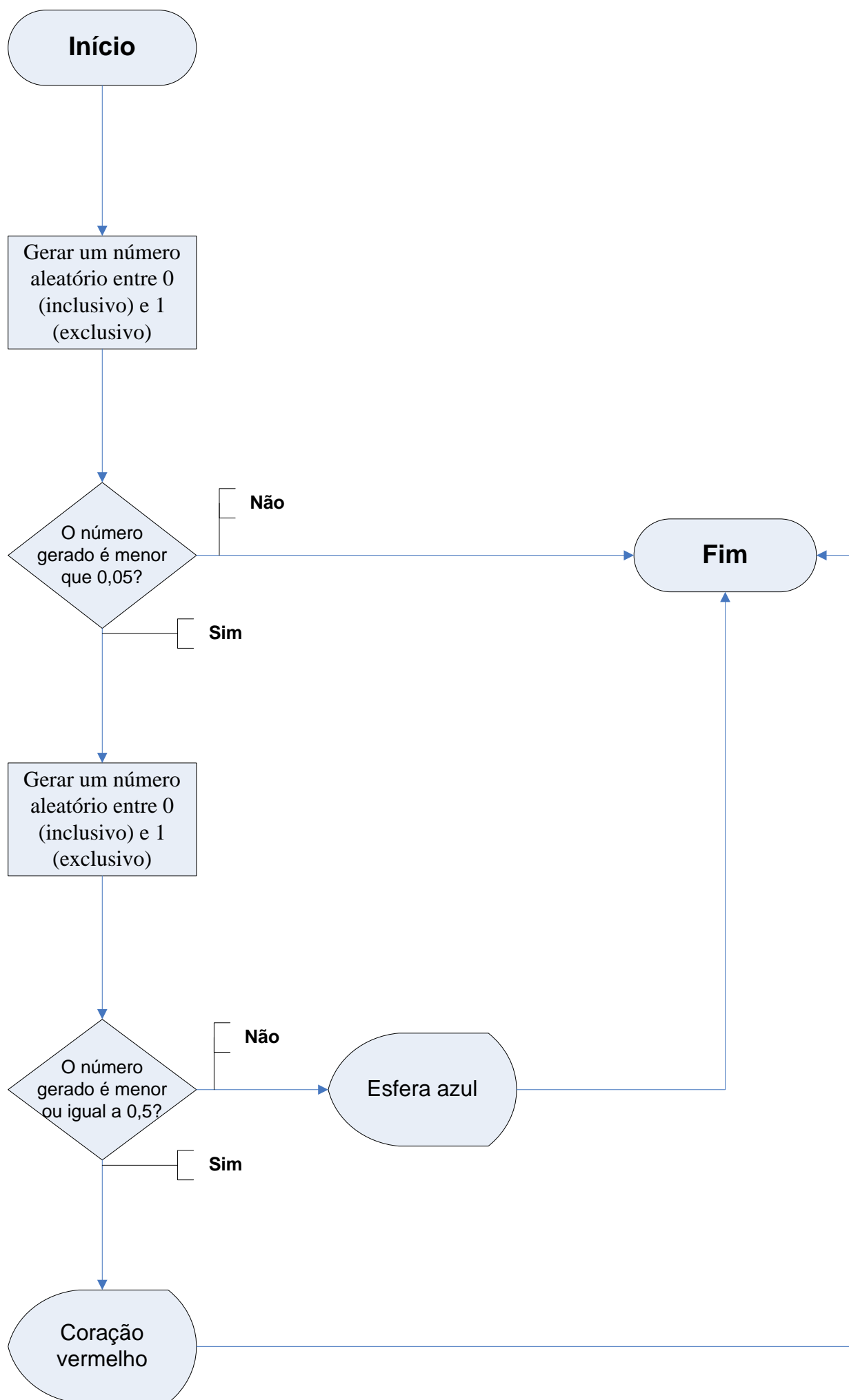
O resto da página do navegador é preta (Fig. 14), para que a visualização seja melhor, já que o branco padrão por tempo prolongado pode irritar o olho, e se economize energia, preocupação constante em um mundo cada vez mais eco-sustentável.



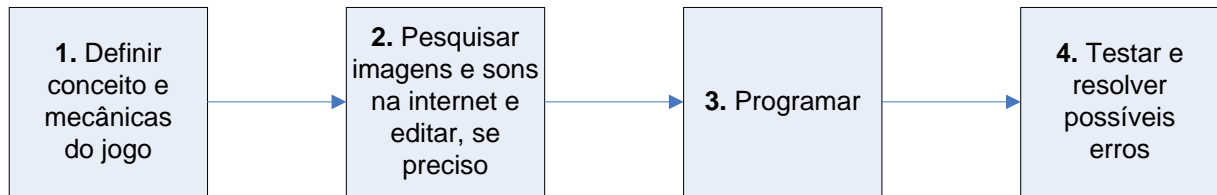
Fig. 14: Página do navegador

Acima do canvas, haverá uma breve, porém eficiente, instrução dos controles utilizáveis (setas de direção para se locomover e Ctrl para atirar) na cor amarela (Fig. 14), que faz uma combinação negativa com o fundo preto.

FLUXOGRAMA DO PSEUDOCÓDIGO DA PROGRAMAÇÃO DO BÔNUS



FASES E DESCRIÇÕES DO PROJETO



1. Definir conceito e mecânicas do jogo

Criar o conceito do jogo, a plataforma a ser usada, quais as mecânicas, se vai ter história ou não.

2. Pesquisar imagens e sons na internet e editar, se preciso

Procurar na internet por imagens e sons que se enquadrem no que foi decidido anteriormente e, caso seja necessário, editar para que fiquem melhor.

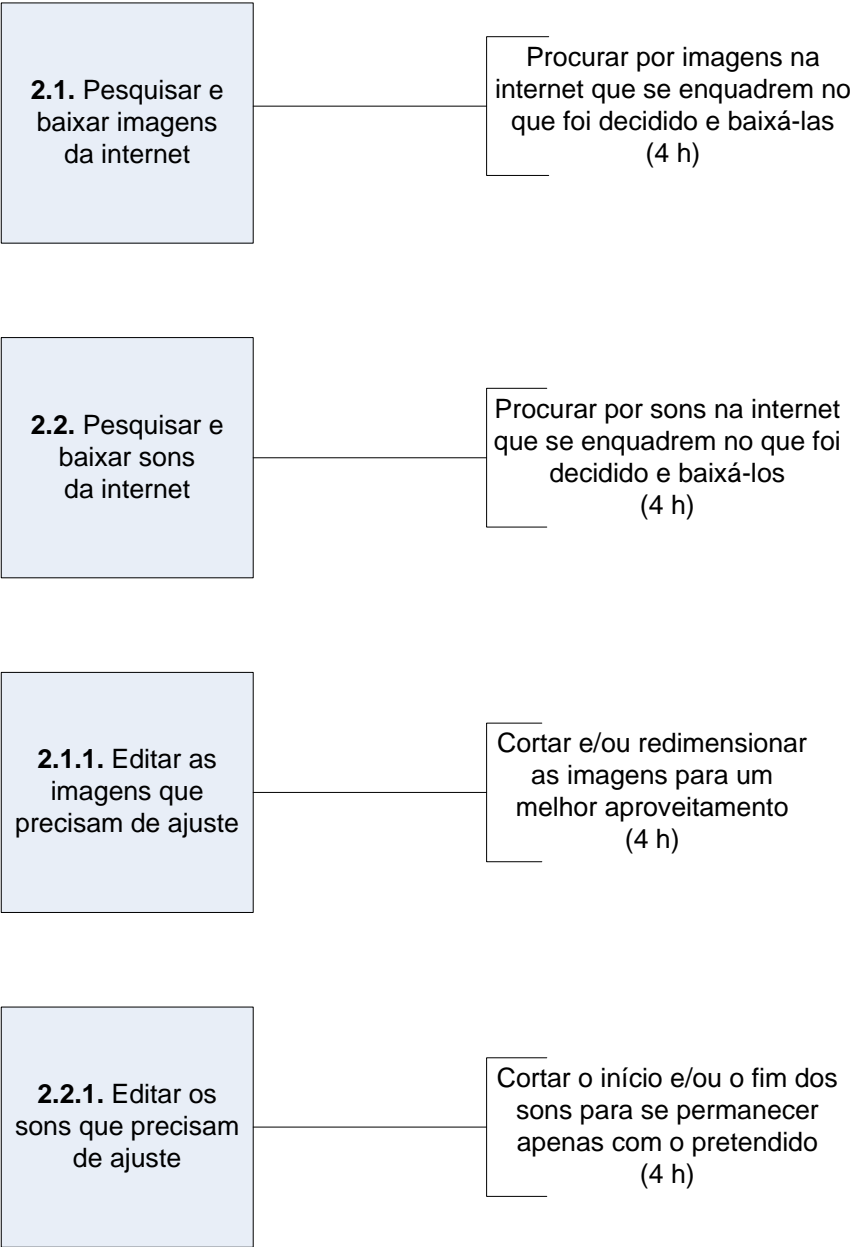
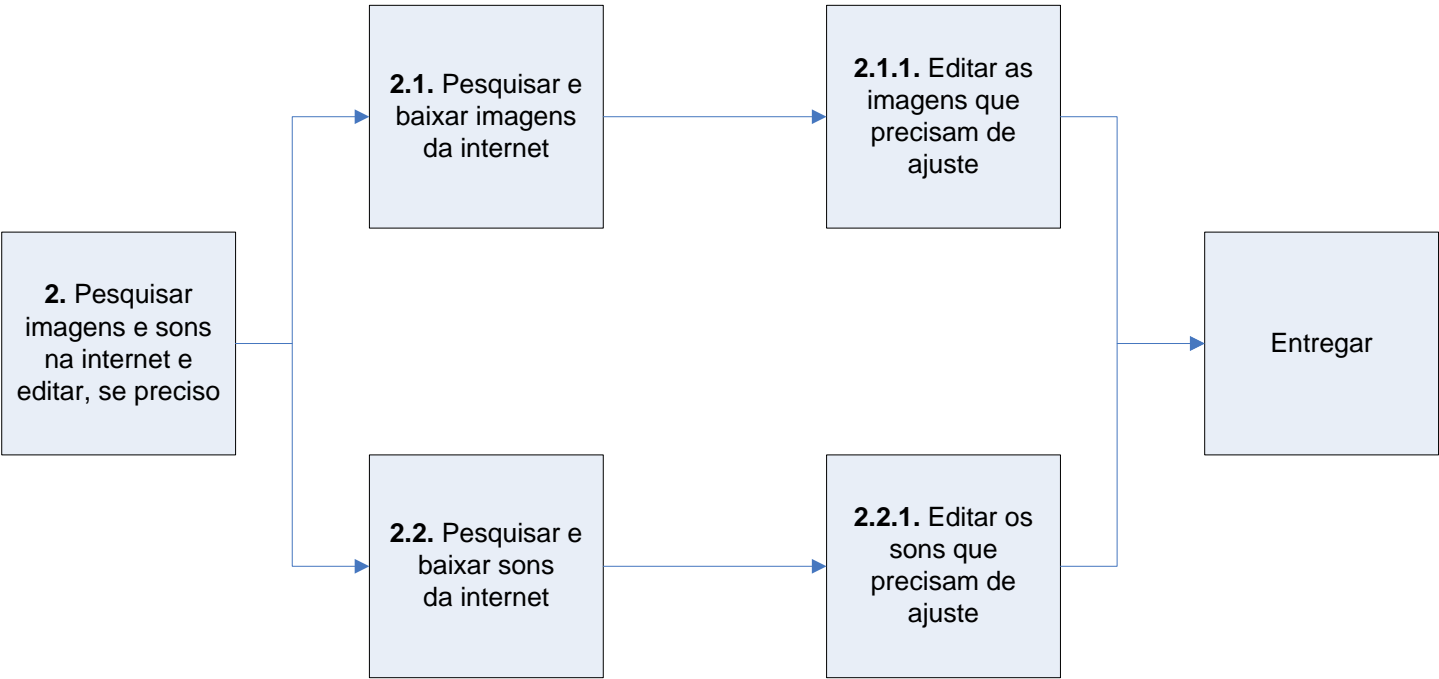
3. Programar

Usar os melhores meios disponíveis pela linguagem de programação decidida, para que o jogo não se torne muito dispendioso para a memória nem para o processador do usuário.

4. Testar e resolver possíveis erros

Descobrir os causadores dos erros e solucioná-los, buscando sempre a otimização do código e a ausência de bugs.

ESTRUTURA ANALÍTICA DA FASE 2 DO PROJETO



CRONOGRAMA DA FASE 2 DO PROJETO

Tarefa	Duração								
		1	2	3	4	5	6	7	8
2.1. Pesquisar e baixar imagens da internet	4h								
2.2. Pesquisar e baixar sons da internet	4h								
2.1.1. Editar as imagens que precisam de ajuste	4h								
2.2.1. Editar os sons que precisam de ajuste	4h								