

O Processo de criação gráfica de um jogo em *Flash*

ANDRÉ LUIZ BATTAIOLA

LAI / albattaiola@ufpr.br

TÁSSIA VIDAL VIEIRA

LAI / tassiadesign@yahoo.com.br

GUSTAVO KIRA

LAI / gustavokira@hotmail.com

LAI - Laboratório de Animação Interativa

Departamento de Design – Universidade Federal do Paraná

Rua General Carneiro, 460, 8º andar, sala 814, Curitiba, Paraná

Resumo

Edugraph é um jogo, em desenvolvimento, para o ensino de conceitos de computação gráfica. Ele conta com ambientes 2D e 3D. Este artigo descreve o processo de criação gráfica de uma das fases 2D deste jogo a partir do pressuposto de que este jogo será publicado na *Web* através da tecnologia *Macromedia Shockwave*.

Palavras-chave: Jogos educativos, Jogos em *Flash*.

1 Introdução

1.1 O que é *Edugraph*

Edugraph é um jogo educativo que visa ensinar princípios básicos da Computação Gráfica. A etapa introdutória do jogo é constituída de um vídeo, onde é apresentado Edu, o personagem principal. Em função de uma trama, Edu é forçado a manipular um controle, cujos botões ativam funções de deslocamento, rotação e mudança de eixo para controlar um conjunto de esferas, que abrem um portal ao serem encaixadas nele. O portal atua como um teletransportador que move Edu para dentro de um cubo, em cujas paredes e teto se encontram novas fechaduras que atuarão como a passagem de Edu para outros mundos, onde ele deverá realizar determinadas atividades. Dois destes mundos são bidimensionais e três são tridimensionais.

O objetivo do jogo é ensinar ao jogador os conceitos de transformações afins, (deslocamento, rotação e escalamento), bem como de *operações booleanas*, comandos básicos da computação

gráfica, que permitem a modelagem de objetos complexos a partir da combinação de formas básicas, seja por união, intersecção ou subtração dessas formas.

Considerando-se que antes de aprender a utilizar esses comandos dentro de um ambiente 3D, é mais fácil ensinar seu princípio em um ambiente bidimensional, as duas primeiras fases do jogo são realizadas em 2D.

O que motiva o personagem a percorrer as fases do jogo aprendendo esses princípios é a necessidade de recuperar as cores de seu corpo, as quais ele perde ao passar pelo portal exibido no início do jogo. Conforme ele acaba uma fase, ele recupera uma de suas cores.

Este artigo explora aspectos do processo de desenvolvimento do cenário e da interface de uma das fases deste jogo em desenvolvimento.

1.2 Especificações da primeira fase 2D

Na primeira fase serão apresentadas ao jogador todas as operações booleanas, e a cor que o personagem irá recuperar é o amarelo de seu

braço. Ele também terá contato com operações de rotação e translação.

O roteiro aborda uma situação em que o personagem necessita montar uma chave utilizando as operações booleanas, para que possa concluir esta fase. Antes de cumprir seu objetivo, ele se depara com uma série de obstáculos pelo caminho, de forma a tornar a experiência do jogador mais interessante.

Para que o jogador compreenda sua missão, optou-se por mostrar a chave completa logo no início da fase, quando o personagem surge no primeiro nível do cenário. Ela fica posicionada ao lado da porta que leva a próxima fase, e nesta porta há uma fechadura com o mesmo formato desta chave.

Espera-se que o jogador veja como única possibilidade pegar a chave e abrir a porta logo no início. Porém, quando o Edu tenta alcançar a chave, ela se parte em 7 pedaços e esses se espalham pelo cenário, um pedaço em cada nível, de forma que, para recuperar tais fragmentos, o jogador precise passar pelos obstáculos presentes no cenário.

Esses obstáculos foram definidos com o objetivo de testar a coordenação entre as mãos e os olhos do jogador, bem como apresentar todos os movimentos possíveis do personagem, o que será útil para o desenvolvimento do jogo.

Nessa fase do jogo também haverá uma contagem de pontos e vidas. Pontos são ganhos quando o personagem recupera uma peça e, dependendo de quantos pontos possui, ele ganhará novas vidas. Assim, alguns desafios que ele encontra pelo caminho servem apenas para atrasá-lo, mas há perda de vidas quando ele sofre grandes quedas ou é atingido por rolos ou projeteis. Ao se zerar os pontos de vida, ele retorna ao início da fase.

Superado esses desafios, o jogador deve levar Edu de volta ao primeiro nível deslizando por um cano que fica no fim do último corredor. De volta a porta, ele deve montar a chave e passar para a próxima fase.

1.3 Limitações e necessidades detectadas

Apesar do objeto a ser montado ser uma chave, foi decidido que ela não possuiria o formato de

uma, mas uma forma geométrica mais regular, facilitando a sua construção pelo jogador.

Ressalte-se que não se considerou o uso de animais ou outros seres vivos nos cenários dessa fase, bem como nenhuma forma de inimigo, porque o roteiro principal não mostra a existência de um antagonista ou de outras formas de vida nesse ambiente em que o personagem se encontra.

Para que o jogador note claramente que a recuperação de suas cores está ligada à conclusão da fase, o cenário deve ser basicamente monocromático, na cor que ele irá recuperar ao final da fase, nesse caso, o amarelo.

Também é importante que se evite ao máximo utilizar objetos com volume e sombra no cenário, para marcar a diferença entre uma representação 3D e uma 2D.

2 Estado da Arte

2.1 Cenários e Interfaces de jogos 2D

Analisando a interface dos jogos eletrônicos na sua origem, o elemento que mais se destaca é a simplicidade dos gráficos. As primeiras interfaces eram constituídas de imagens “rasterizadas” de baixa resolução, apresentando objetos com pouca definição e formas simplificadas (figura 1). Devido às limitações tecnológicas, eram usadas poucas cores, e essas sem variação de tom (“chapadas”), ou seja, não buscavam criar volume através do uso de sombras. Os objetos eram normalmente vistos frontalmente, ou lateralmente, sem nenhuma forma de perspectiva.

Além das características gráficas, observou-se que os elementos mais comuns das histórias e dos cenários eram: chaves, elementos que possibilitam mudanças de fases; desafios limitados à tela, não havendo a rolagem do cenário; labirintos; desafios consistindo em subir do nível mais abaixo da tela até o topo, onde está o vilão, utilizando-se escadas, e com o nível de dificuldade aumentando conforme se sobe.

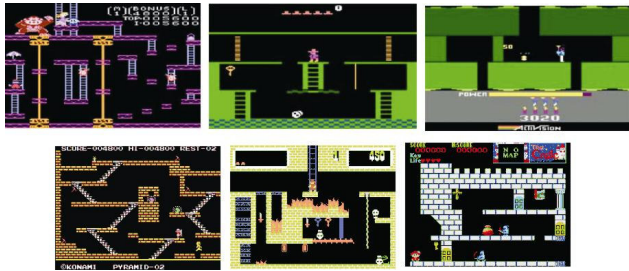


Figura 1 – Alguns dos primeiros jogos 2D, que serviram como base ao desenvolvimento do jogo *Edugraph* [1].

Nota-se claramente que a evolução das interfaces e cenários dos jogos 2D acompanha o desenvolvimento tecnológico na área computacional. Graficamente, os cenários passaram a ser mais detalhados, a utilizar mais cores e sombras e, aos poucos, a considerar a idéia de perspectiva. Paralelamente, também os tipos de jogos e histórias foram se distinguindo, surgindo novas modalidades de jogos, muitos deles totalmente ambientados em 3D (figura 2), mas com movimentação ainda bidimensional.



Figura 2 – Alguns exemplos do caminho seguido pelos jogos 2D [2].

Para o desenvolvimento da interface e do cenário do jogo *Edugraph*, preferiu-se usar como base os jogos mais antigos, pois eles acentuam a diferença entre ambientes 2D e 3D.

2.2 Sprites

Os *sprites* [3] são um dos elementos essenciais para o desenvolvimento de um jogo, pois são seqüências pré-definidas de animações de personagens e objetos.

Nos primeiros jogos, os *sprites* constituíam animações simples e de pouca qualidade. A

quantidade de quadros por animação era a mínima possível, o que gerava movimentos quebrados. Além disso, utilizavam-se poucas cores e resoluções baixas.

Com o desenvolvimento dos jogos, as animações foram se tornando mais suaves, com mais cores e gráficos mais definidos, bem como ampliando a quantidade e variedade de movimentos. Por outro lado, simplificações persistiam. Por exemplo, apesar das animações terem se tornado mais suaves, não se considera animações intermediárias entre dois movimentos distintos, pois essa mudança de movimento é tão rápida que não atrapalha a experiência do jogador. Também era comum o uso de imagens espelhadas, por exemplo, para representar o movimento de retorno dos personagens.

Com o surgimento de jogos 3D também ocorreram muitas mudanças no sistema de *sprites*. Em alguns desses jogos, o personagem 3D é simulado através de *sprites* “planos”, mas reproduzidos em vários ângulos. Em outros jogos, os *sprites* são verdadeiros objetos 3D, de poucos vértices, mas com movimentos previamente animados como em sua versão 2D [4].

Apesar dos métodos de uso dos *sprites* terem sido aperfeiçoados, eles continuam sendo limitados pelo fato da movimentação estar baseada em animações pré-definidas, o que implica em se considerar o compromisso de que sofisticar a animação implica em aumentar o número de *sprites*. Por outro lado, requerem pouco processamento computacional, o que os tornam atrativos na implementação de jogos simples.

2.3 Plataforma do jogo.

Optou-se por desenvolver as etapas 2D do jogo *Edugraph* no programa *Macromedia Flash*, para a publicação na *Web* através da tecnologia *Shockwave*. As razões para essa decisão foram: independência de plataforma, acesso fácil e gratuito ao *plug-in*, extensa base de usuários. Além disso, a linguagem *ActionScript*, utilizada para programar no *Flash*, está em constante evolução, viabilizando novos recursos e, conseqüentemente, a implementação de jogos cada vez mais sofisticados.

Um outro ponto positivo do *Flash* é o fato de ter sido inicialmente uma ferramenta destinada à animação, possuindo diversas ferramentas e recursos

que permitem um bom desenvolvimento gráfico desses jogos. O destaque é o uso de gráficos vetoriais, que além de permitir redimensionamento para se adaptar ao tamanho da janela, também cria arquivos mais leves com gráficos de boa qualidade. O recurso de *movieclips** é uma ótima opção para o desenvolvimento de *sprites* e o reaproveitamento de símbolos ajuda a criar arquivos menores e mais fáceis de serem construídos.

3 Etapas de Desenvolvimento

3.1 Redesenhando e movimentando o Edu

Ao se exportar o jogo de um ambiente 3D para o 2D usado no programa *Flash*, foi necessário se adaptar o personagem, redesenhando-o com vetores (figura 3). Optou-se por realizar esse processo manualmente, pois, apesar de existirem programas que gerem esse tipo de imagem automaticamente, era necessário fazer algumas alterações para torná-lo uma figura funcional, com formas fáceis de se distinguir e de se animar, o que implica em transformar cada parte de seu corpo em um objeto independente.

Foram realizadas mudanças nos ombros e na bacia do personagem, para simplificar suas formas, optando por transformá-las em peças inteiras de formato mais regular do que sua versão 3D.

Também foi necessário utilizar no personagem um contorno preto de espessura média e alterar sua cor de preto para branco, para que tornasse mais fácil de representar o Edu sem suas cores, pois isto facilita



distinguir as partes da frente das de trás.

Figura 3 – A esquerda versão original em 3D, a direita, versões em 2D.

* Recurso do *Macromedia Flash* que permite criar uma seqüência de animação que pode ser reaproveitada em diversas partes do arquivo. Ele é executado independente da animação principal.

Com o personagem redesenhado, iniciou-se o processo de criação dos *sprites* do personagem, onde cada movimento dele constitui uma animação distinta.

Para realizar essas animações o primeiro passo foi listar os movimentos necessários ao personagem. Nessa primeira fase em que o personagem está em suas cores, é possível reaproveitar cada movimento para fazer a versão espelhada, mas nas próximas fases será necessário definir uma versão para a direita e outra para a esquerda de cada movimento.

O processo escolhido para realizar as animações consistiu em primeiro animar o personagem em 3D e depois exportar os diferentes estágios da animação como arquivos de imagens separadas. Devido à diferença de velocidade entre os dois meios (as animações 3D utilizam 36 quadros por segundo e a animação 2D no *Macromedia Flash* usa 12 quadros por segundo) foi necessário selecionar os quadros a serem usados. Posteriormente, eles foram importados para o *Flash* em seqüência, e alocados um a um em quadros separados. Feito isso, bastou posicionar as partes do corpo do Edu conforme a sua pose no 3D.

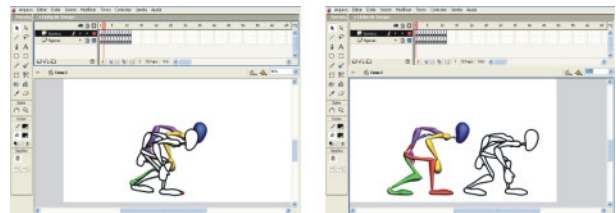


Figura 4 – Animação em desenvolvimento, com base em modelo 3D.

Trabalhar com esse intercâmbio entre animações 3D e 2D apresenta as seguintes vantagens:

- a) Maior praticidade e economia de tempo. A animação em 3D é criada apenas definindo os quadros-chaves, o programa (*Discreet 3D Studio Max*) sabe como o corpo humano deve se mexer e gera os quadros intermediários sozinho. Em 2D isso não ocorre, e quando é necessário criar movimentos complexos, a animação deve ser feita quadro a quadro. Por essa razão se torna mais rápido e prático criar os movimentos em 3D e depois copiá-los em 2D.

- b) Animações mais uniformes. Animações realizadas por mais de uma pessoa podem ocasionar diferenças de estilo de uma para a outra. No entanto, isso não ocorre se todos os desenhistas se basearem na mesma fonte, no caso a animação 3D. Adicionalmente, permite uma maior similaridade entre os movimentos do personagem em 3D e sua versão 2D.
- c) Aumento da qualidade. As animações aumentaram em qualidade, pois a base foi feita por um animador com maior experiência, não importando então a experiência dos animadores 2D, permitindo que mais pessoas pudessem participar do trabalho, tornando-o ainda mais rápido.

Apesar do uso do auxílio 3D ter sido essencial em nosso projeto, não quer dizer que o seu uso seja útil em todos os outros projetos de animações 2D. Há muitos aspectos a serem considerados.

Primeiramente a praticidade que as ferramentas 3D proporcionam a animação de personagens só se torna importante se a animação desenvolvida requisitar animações realistas. Nesse caso, sua praticidade já está com tanta evidencia que surgiram vários novos programas especializados e *plug-ins* que tentam dar um aspecto cada vez mais semelhante as da coloração e da estética encontrada em animações 2D. Esses programas criam novas possibilidades, tais como, um uso mais arrojado de sombras, que são essenciais para criar a sensação de maior realismo.

Se o resultado desejado for uma animação 2D com os movimentos dos personagens mais "cartunizados", o uso do programa 3D pode na verdade consumir mais tempo, ao invés de ajudar, um exemplo de animação em que isso ocorreria é no desenho animado *As Meninas Super Poderosas* [5], onde o movimento dos personagens é tão simples que teria pouca utilidade usar um programa de animação 3D. Mas esse não é o único caso em que o uso de programas 3D não é atrativo, algumas vezes ao se optar por fazer uma animação bidimensional, a razão pode estar no processo de criação de uma animação desse estilo, em seu processo mais manual que busca levar as qualidades gestuais dos desenhos às produções animadas.

A questão das qualidades gestuais também é levantada quando se considera o uso de programas 3D na construção de cenários. Fica claro suas

vantagens em animações com cenários mais realistas ou exatos como os do desenho animado *Futurama* [6], que já utiliza esse recurso em seus cenários. Mas em uma animação como a vencedora do Oscar de 1999, *The old man and the sea* [7], não há espaço para a intervenção de ferramentas 3D, a não ser que seu uso se baseie na técnica usada no filme *Rugrats in Paris*[8], onde a animação de objetos como carros e cenários foram modelados e animados no computador, mas pintados manualmente.

Por isso a aplicação de ferramentas 3D é algo a ser estudado caso a caso, e sempre tendo em mente que isso não é uma segurança de qualidade, pois um animador 2D experiente pode produzir um trabalho de maior qualidade do que um animador 3D inexperiente.

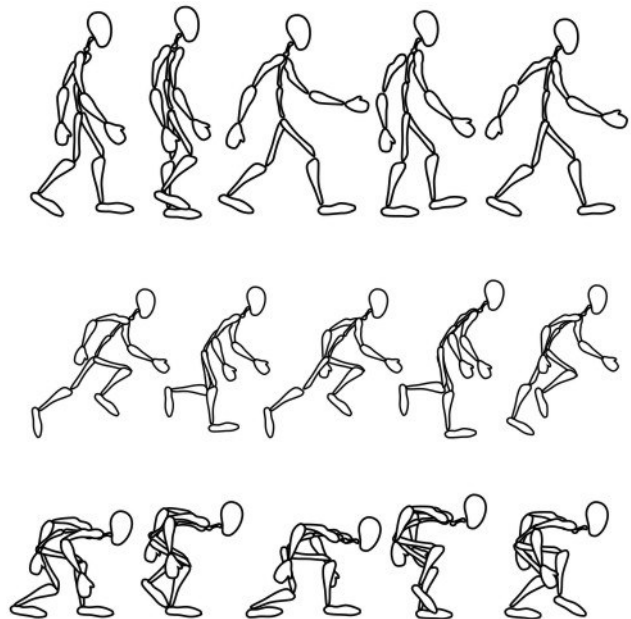


Figura 5 – Alguns exemplos das animações desenvolvidas: ciclo de caminhada, ciclo de corrida e o movimento de caminhar agachado.

3.2 Cenário

Para que não haja desperdício de recursos e de tempo, é importante que as idéias geradas para o jogo sejam testadas antes de serem finalizadas, por isso as primeiras soluções encontradas para o cenário devem ser consideradas como protótipos.

Assim, os primeiros cenários desenvolvidos possuem gráficos extremamente simples, sem se

preocupar com o apelo visual do cenário ao jogador. Pois, para que seja fácil realizar modificações e novos testes, é necessário manter a simplicidade das formas.

No processo de criação desse jogo, o teste do cenário foi feito através de animações, pois os controles ainda estão em processo de desenvolvimento. Através dessas animações é possível analisar a lógica dos obstáculos e como as informações seriam recebidas pelo jogador conforme o personagem se movimenta pelo cenário. Mas para viabilizar essa análise, foi importante instruir os animadores de maneira que suas reações em frente aos obstáculos se aproximassem da reação do jogador.

3.2.1 O primeiro protótipo

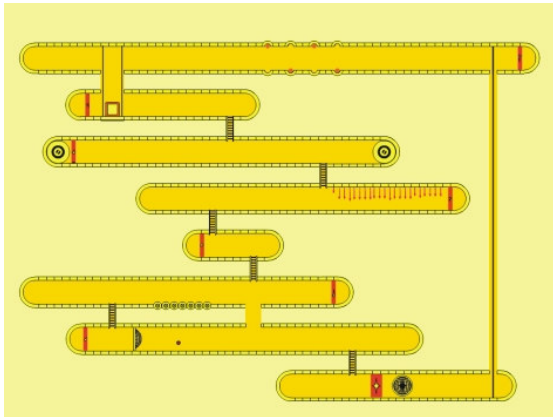


Figura 6 – Imagem do primeiro cenário.

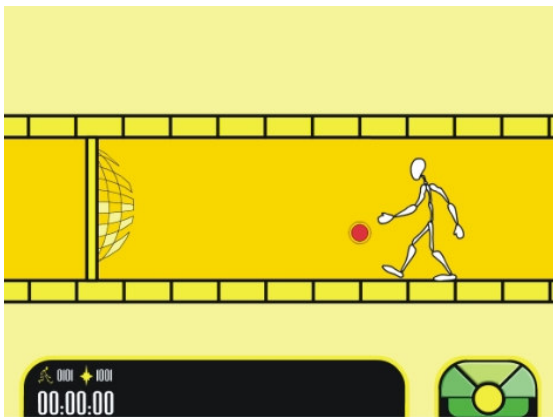


Figura 7 – Imagem do ponto de vista do jogador.

Considerou-se que os níveis do cenário funcionariam como corredores dispostos em vários andares de uma estrutura fechada, como um subsolo. Esses andares seriam unidos um ao outro por uma escada. Os desafios ficaram separados de forma a ter um único desafio por andar.

Os obstáculos que podem ser vistos nesse primeiro protótipo (figura 6) são:

- i. Porta que abre ao ser ativada por um sensor, mas que se fecha logo em seguida.
- ii. Chão com rolos que impulsionam o personagem para a direção oposta ao do fragmento de chave que o personagem deseja recuperar.
- iii. Escada posicionada sobre um pequeno precipício.
- iv. Há lanças saindo do teto, apontando para ele. Para desviar-se ele é forçado a andar agachado.
- v. Rolos gigantes caminham em direção ao personagem depois de recuperar um fragmento.
- vi. O personagem deverá passar por baixo de um elevador para pegar o fragmento e depois, para passar ao próximo nível, deve remover o bloco que está no elevador para poder usá-lo como meio de transporte.
- vii. Bolas de energia são atiradas do teto e do chão, o personagem deve passar por elas sem ser atingido.

Esse cenário foi baseado nos jogos observados durante a pesquisa, principalmente as diversas versões de *Super Mário*. Porém ao contrário desses jogos, o cenário não fica limitado ao que aparece na tela, mas vai se revelando conforme o personagem o percorre (figura 7). Nesse aspecto ele se assemelharia mais aos jogos *HERO* [9] e *Prince of Pérsia* [10], onde o ponto de vista do jogador se move junto com a câmera pelo cenário, de cima para baixo ou para os lados, mostrando somente o que é de interesse ao jogador no momento.

Os corredores que compõem o cenário, assim como os sensores e outros objetos, possuem formas arredondadas, seguindo a forma da porta que o personagem encontra no começo do jogo, de forma a

criar uma unidade visual com o ambiente 3D e os cenários que ainda serão desenvolvidos.

Apesar do cenário ter sido definido como monocromático (amarelo), era necessário utilizar pelo menos uma outra cor, não só para quebrar a monotonia, como também para destacar alguns objetos quando necessário. A cor escolhida foi o vermelho, uma cor que se destaca no olho do jogador ao mesmo tempo em que se harmoniza com o amarelo.

Os problemas detectados durante o teste foram: a) a altura dos níveis não permitiam ao personagem pular, b) a existência de grandes espaços vazios entre o momento que o personagem chega a um nível e o momento em que se encontra com o obstáculo, devido ao tamanho excessivo de alguns níveis, cria não só momentos de tédio, como quebras graves no ritmo do jogo.

Também se detectou que a lógica de alguns desafios estava incorreta e poderia levar a situações em que o jogo entraria em *dead-lock*, o que obrigaria o jogador a reiniciar, fato que lhe causaria desconforto e decepção. Erros graves de projeto, como este, têm de ser detectados no início, antes que acarretem grandes problemas no futuro.

Ao se corrigir esse problemas, foi gerado o segundo protótipo.

3.2.2 O segundo protótipo

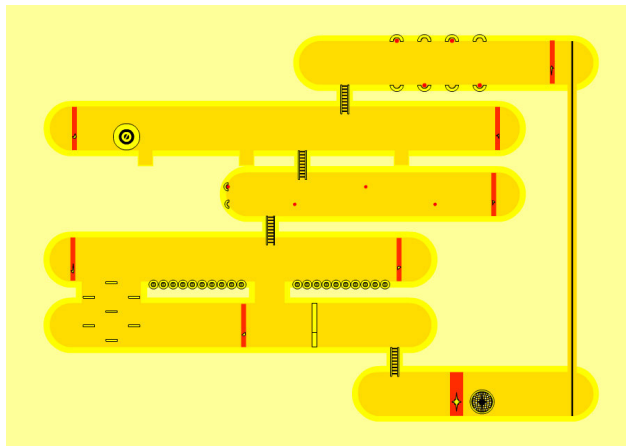


Figura 8 – Imagem do segundo cenário

Os desafios foram condensados em 5 níveis, (figura 8) além do nível inicial, eliminando os

grandes espaços vazios entre um desafio e outro. Além disso, novos desafios foram adicionados, outros modificados.

O elevador que causava grandes problemas ao jogo (pois podia travá-lo) foi substituído por um conjunto de plataformas instáveis que caem com o “peso” do personagem. O conjunto de lanças foi substituído por um conjunto de dois atiradores, um que obriga o personagem a pular para desviar, enquanto o outro obriga o personagem a se abaixar.

A porta que era aberta por sensor foi substituída por uma porta que abre e fecha automaticamente, o que obriga o jogador a calcular o tempo que necessita para ultrapassá-la.

Os dois “rolos compressores” acionados no momento que o Edu pegava a chave foram trocados por um único rolo em movimento constante e um conjunto de buracos que permitem a fuga do personagem.

Após o teste alguns novos problemas foram detectados, como a necessidade de se aumentar a altura entre as plataformas no nível 2 e a necessidade de que os tiros no nível 4 só se iniciassem após o personagem subir a escada (pois o jogador não tem controle do personagem durante a subida da escada).

As correções foram feitas e após uma nova animação-protótipo iniciou-se a finalização do cenário.

3.2.3 Desenvolvimento do cenário final

Constatou-se que para tornar o cenário visualmente mais agradável ao jogador era necessário não só a remodelagem dos níveis, criando novos contornos e melhorando a integração entre cenário e obstáculos, mas também a criação de um plano de fundo que ocupasse as áreas vazias por entre os níveis.

O tema escolhido para o novo cenário é baseado em elementos das antigas casas de máquinas de navio. Esse cenário busca resgatar elementos das fases em 3D já desenvolvidas para o *Edugraph*, de forma a criar uma unidade visual entre eles.

O contorno dos níveis e o fundo simulam chapas metálicas presas com parafusos, sendo as do fundo, chapas lisas e as do chão e paredes, chapas rugosas. O cenário também é cercado de tubulações com medidores de pressão, calor e válvulas, exatamente como uma casa de máquina. Os obstáculos ou se assemelham aos tubos ou a outros

elementos presentes nessas construções: válvulas, motores com correias, tubos de ar comprimido, entre outros.

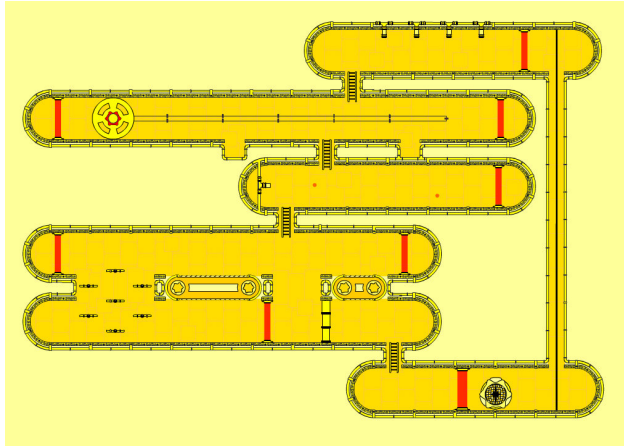


Figura 9 – Imagem do cenário final

3.3 Interface

Como não há uma imersão do jogador no ambiente 2D como no 3D, é necessário apresentar os controles especiais, que são a base do jogo, através de uma interface, além da necessidade de apresentar outras informações ao jogador como tempo transcorrido, pontos e número de vidas.

O formato da interface é similar ao dos corredores que compõem o cenário, mas ao invés de ter todos os extremos arredondados, seu lado inferior está ligado ao extremo da tela, de forma a não ser confundido como uma parte do cenário e sim como um elemento externo aos acontecimentos, indicando ser parte do monitor, o que consolida sua função de interface entre o usuário e o mundo 2D apresentado.



Figura 10 – A interface desenvolvida.

Os botões foram moldados de acordo com o formato da interface, perdendo qualquer semelhança aos botões do console que aparece em 3D. As cores e o posicionamento são os principais elementos que caracterizam esses botões, por essa razão foram mantidos.

A interface é dividida em dois painéis. O primeiro contém os indicadores de tempo, vida, pontos e peças coletadas. O segundo contém os três botões de transformações afins mais um botão que tem a função de alternar entre esse e um novo conjunto de botões, que tem como função as operações booleanas.

Os indicadores estão no lado esquerdo do jogador, ficando em uma posição de destaque (se considerado o modo de leitura ocidental), pois devem ser consultados com frequência durante o jogo. Os controles estão no extremo direito, deixando o centro reservado aos fragmentos da chave que forem coletados. Assim, apesar de estar no mesmo painel que os outros indicadores, os fragmentos também podem ser associados aos controles pela sua proximidade. Essa associação é importante, pois os botões estão ali para interagir com essas peças.

A razão de existirem dois painéis é que, ao final da fase, o painel que contém os indicadores irá aumentar, se tornando algo similar a uma janela dentro do jogo, e nesta tela a chave irá ser montada com o auxílio dos botões no outro painel.

3.4 Preparando o cenário para o jogo

Para a geração do cenário no *Flash*, foi necessário se estudar os seus elementos, de forma a detectar aqueles que poderiam ser transformados em símbolos reutilizáveis, diminuindo o tamanho do arquivo final. Em um experimento utilizando somente a base dos andares do cenário, houve uma diferença de 30 Kb, que pode parecer um valor irrelevante, mas em um cenário de proporções maiores essa diferença só iria aumentar.

Após a finalização do desenho do cenário, começou a preparação deste para o uso no jogo. Mas não antes de preparar uma nova animação, que servirá de referência aos programadores.

Os *sprites* do personagem foram unidos em um único arquivo, para que se torne mais fácil à transição de um movimento para o outro. Se isso não fosse realizado, toda vez que o personagem mudasse de um estado para o outro seria necessário importar um novo arquivo para dentro do palco, gerando atrasos no processamento.

As animações dos obstáculos foram geradas como *movieclips*, para que rodassem independente do número de quadros da animação total, que no

caso de um jogo é usualmente um só quadro, variando para mais apenas se necessário.

Cada nível do cenário foi salvo como um *movieclip* à parte dos outros níveis. Cada *movieclip* de nível contém os *movieclips* dos obstáculos e dos outros elementos daquele nível. O chão, o teto e as paredes, apesar de estarem juntos dentro do *movieclip* do nível, estão separados um do outro visando facilitar a implementação do processo de detecção de colisão.

Além disso, o fato dos níveis estarem organizados e identificados de forma única e independente facilita a manipulação, por exemplo, durante o processo de mover ou enquadrar cada nível na tela.

A interface foi posicionada em uma camada acima do destinado aos níveis para que fique sempre presente.

Na biblioteca de símbolos todos os itens foram organizados em pastas e subpastas, sendo mantidos somente os arquivos que realmente serão usados.

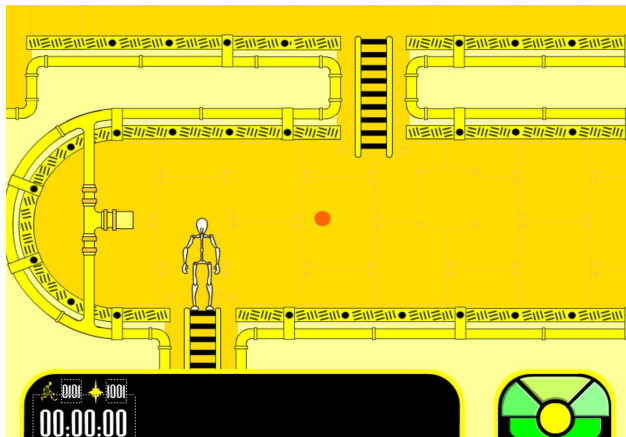


Figura 11 – Ponto de vista do jogador, mostrando o cenário final e a interface desenvolvida.

4 Dificuldades

No decorrer do desenvolvimento do projeto novos problemas foram detectados, além daqueles abordados na introdução, mas dessa vez limitações de cunho mais técnico.

O primeiro problema surgiu ao se deparar com a questão de como transpor o personagem de um nível para o outro, considerando que não

havia uma maneira de trabalhar com o cenário como um objeto inteiro no tamanho necessário para que o jogador visse apenas um nível por vez, pois seu tamanho total ultrapassava a área útil do programa *Flash* e se tornava muito complexo para ser processado com a eficiência desejada. Optou-se, então, por transformar cada nível em um objeto a parte (um *movieclip*), bem como tirar o controle do jogador no momento em que o personagem sobe uma das escadas, o que permite que se implemente uma transição controlada pelo programa.

Notou-se outra peculiaridade ao se desenvolver o teste dos protótipos em forma de animação. Percebeu-se que enquanto o processo de colocar um *movieclip* dentro do outro ajuda na organização e processamento do jogo, não é uma boa forma de gerar animações, pois não permite que haja interação com o cenário.

A geração dos *sprites* do personagem também apresentou problemas inesperados. Considerando que o personagem foi desenhado de forma modular, acreditava-se que seria possível animá-lo sem uso da animação quadro-a-quadro, mas se aproveitando do *motion tween*[†] e da ferramenta de caminho[‡]. Porém como a seqüência animada possui poucos quadros, percebeu-se que a animação quadro-a-quadro (com auxílio da animação 3D) seria mais fácil de ser obtida e não ocuparia muito mais espaço do que a com caminho e *motion tween*.

No que se refere às dificuldades da definição do cenário, acredita-se que o processo teria sido mais simples se o tema estivesse definido desde o princípio do jogo, ao invés de ter sido definido após a criação dos obstáculos.

[†] Recurso do *Macromedia Flash* que calcula automaticamente o posicionamento intermediário de um objeto, após o usuário fornecer a posição inicial e final. Também é responsável pelas alterações de tamanho e rotação.

[‡] Recurso do *Macromedia Flash* que funciona como guia para o deslocamento de um objeto, normalmente é criado a partir de uma curva de *Bezier*.

5 Conclusão

Analisando o desenvolvimento desse primeiro cenário, é possível traçar um novo plano de trabalho para o desenvolvimento do nível seguinte desse jogo.

Em primeiro lugar, o tema para o cenário seria desenvolvido antes da definição do roteiro, facilitando a criação dos obstáculos. Assim seria possível representar nos protótipos as formas do obstáculo além do seu mecanismo, mas de forma simplificada.

Espera-se que com o fim do desenvolvimento do primeiro nível, o controle do jogo seja criado, facilitando assim o processo de teste dos novos protótipos. Caso não seja desenvolvido antes da criação do segundo nível, os novos protótipos serão desenhados e organizados para a realização de animações, ao invés de preparar para a construção do jogo como foi feito com o nível atual.

No que se refere à animação do personagem, inicialmente houve atrasos em seu desenvolvimento pois as animações de seus movimentos começaram a ser feitas diretamente em 2D, obtendo resultados de baixa qualidade. Caso surja a necessidade de desenvolver novos movimentos, esses serão animados primeiramente em 3D, para depois serem animados em 2D.

Por último o desenvolvimento do desenho final do cenário será analisado em detalhe para o melhor aproveitamento do recurso de símbolos do *Flash*.

Espera-se que dessa forma o desenvolvimento do novo nível seja feito com mais praticidade e menor desperdício de tempo.

6 Referências

1. BLAIR, Preston. *Cartoon Animation*. Westport: Walter Foster Publishing, 1994.
2. HUNTER, William. *The Hystory of Video Games: From Pong to Pac-man*. Disponível em: <<http://www.designboom.com/eng/education/pong.html>>. Último acesso em: 01 mar. 2004.
3. RETRO SPACE: *A história dos videogames em 40 capítulos*. Disponível em:

<<http://outerspace.terra.com.br/retrospace/>>. Último acesso em: 01 mar. 2004.

4. SIEKS, David. *Different Perspectives on 3D Sprites*. Game Developer Magazine, p. 55-60, abr./maio 1997.

-
1. <http://outerspace.terra.com.br/retrospace> (5.8.2004)
 2. <http://outerspace.terra.com.br/retrospace> (5.8.2004)
 3. <http://www.videogamesprites.net/> (20.8.2004)
 4. SIEKS, David. *Different Perspectives on 3D Sprites*. Game Developer Magazine, p. 55-60, abr./maio 1997.
 5. <http://www.cartoonnetwork.com.br/ppg/index.html> (20.8.2004)
 6. <http://www.foxworld.com/futurama/> (20.8.2004)
 7. <http://www.oldmansea.com/> (20.8.2004)
 8. <http://www.rugratonline.com/makingrr.htm> (20.8.2004)
 9. http://www.atariage.com/software_page.html?SoftwareID=2069 (5.8.2004)
 10. <http://izuminka.free.fr/articles/prince/> (5.8.2004)