



O PESADELO DE FLUFFY

Desenvolvimento de um jogo 3D baseado no mini-jogo Rapunzel (Catherine)

Inspiração: Mini-jogo Rapunzel (Catherine)

Catherine [1] foi lançado em fevereiro de 2011 para Xbox e Playstation 3. No mesmo ano, foram vendidas 500 mil cópias, o que tornou o jogo um dos maiores sucessos da companhia. O jogo ainda tem uma base fiel de fãs o que contribuiu para que fosse lançado novamente em 2020 numa versão para PC distribuída pela Steam.

O jogo Catherine envolve duas dinâmicas principais - quando o personagem está acordado controlando suas ações e tendo que lidar com as consequências de suas escolhas que impactam o rumo e final do jogo. E quando o personagem está no sonho - resolução rápida do puzzle de empurrar e puxar blocos escalando para sobreviver mais uma noite.

Dentro do jogo há um mini-jogo (Figura 2): Rapunzel. Ele envolve a mesma dinâmica da resolução de puzzle, mas de forma mais complexa num cenário menor e sem limitação de tempo. É nesse jogo que mais nos baseamos (Figura 1).



Figura 1: Captura de tela do jogo Rapunzel desenvolvido pela Atlus.

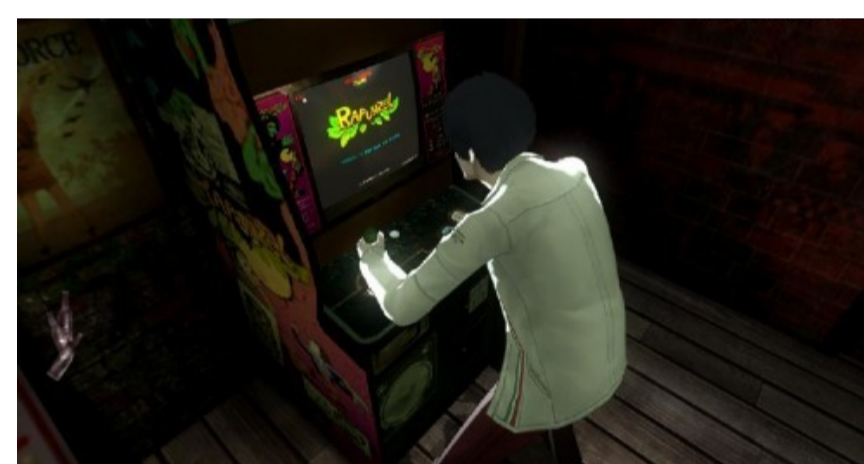


Figura 2: Captura de tela do jogo Catherine desenvolvido pela Atlus. Ilustra como o jogador acessa o mini-jogo.

Ferramentas e bibliotecas no desenvolvimento

O desenvolvimento de jogos na maioria das vezes se dá por meio de engines, programas que facilitam todo o processo de criação e implementação, mas muitas vezes vem com o custo de performance e limitações nas mecânicas do jogo criado. Considerando isso, resolvemos abraçar o desafio de criar um jogo sem o uso de engines.

O programa foi desenvolvido em C++ com o uso das bibliotecas OpenGL (renderização) e SDL (input e output).

A escolha em utilizar C++ está na enorme compatibilidade com as bibliotecas utilizadas, facilidade em trabalhar com classes (POO) e o potencial da linguagem em performance (por ser de baixo nível é possível manipular com precisão os dados sem prejudicar o tempo de processamento).

OpenGL [2] é uma biblioteca de renderização. Ele não mantém nenhuma informação sobre o que renderiza, tudo que ele vê é um conjunto de triângulos e estados para renderizá-los.

A partir da criação desses triângulos é que podemos criar figuras, usando funções que, por fórmulas geométricas, criam figuras 3D como cubos, pirâmides e cones (chamadas primitivas).

SDL [3] é uma biblioteca de desenvolvimento criada para dar acesso de baixo nível ao áudio, mouse, teclado e gráficos do computador. Desenvolvida em C, tem compatibilidade nativa com C++ e é muito popular em jogos e emuladores.

Código e Estrutura de Dados

O programa consiste de um arquivo principal e dois secundários (um com os objetos e outro com as funções de desenho). O arquivo principal é o jogo em si, isso é, tem um loop maior com as fases e telas e um loop

menor que mantém a jogabilidade (atualização de movimentos e cenário). O cenário é um objeto, -Torre- (Figura 3), composto por outro objeto, -Andares-. Cada andar tem -Blocos- que têm tipos variados (tipos que modificam as interações entre eles, o -Player- e a -Torre-).

O -Player- interage com os -Blocos- e causa atualizações em cadeia.

Há duas atualizações principais: atualização do player e da lista de blocos.

As atualizações do player são os input de controle e os de movimentação, mas checa também colisões com a -Torre- que são de tipos variados com efeitos diferentes, por exemplo, podem interromper a movimentação ou sinalizar o fim da fase.

As atualizações dos blocos se dão por meio de uma lista de blocos que entraram em movimento, a cada iteração sua posição é atualizada de acordo com sua velocidade. Quando a posição é "inteira" checa-se colisão com a -Torre- e com o -Player- e dependendo dessa colisão o bloco vem, ou não, a parar.

Outro objeto que vale menção é o -Desfaz-, um objeto que guarda uma lista de estados da -Torre- e do -Player- que permite ao jogador voltar a um estado do jogo antes de uma ação (interação com bloco). Que é uma mecânica importante no jogo.

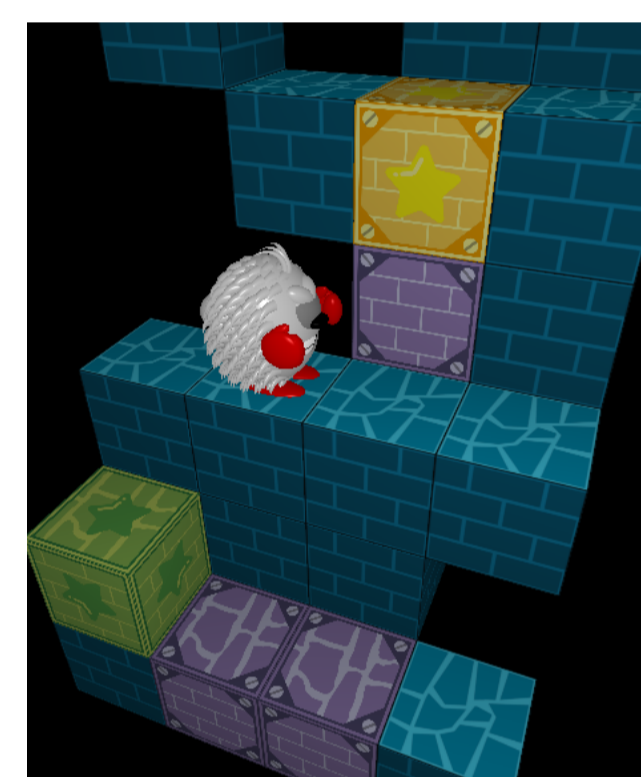


Figura 3: Captura de tela de "O Pesadelo de Fluffy". Na imagem se vê o -Player-, -Torre- e diferentes tipos de blocos

Gráficos e Animações

Todo o jogo é criado a partir de figuras primitivas e texturas projetadas nos planos dessas figuras. O objeto mais complexo criado é o player, mas até ele é um conjunto de primitivas (Figura 4).

Essa forma de gerar objetos gráficos é uma forma mais simples que mantém todo o processamento dentro do próprio código sem uso de arquivos exteriores. Um pró desse método é que toda compilação e geração da estrutura é confinada ao código, o que significa uma compilação e renderização mais simplificada. Um contra é que a complexidade e detalhes que se pode alcançar é limitada

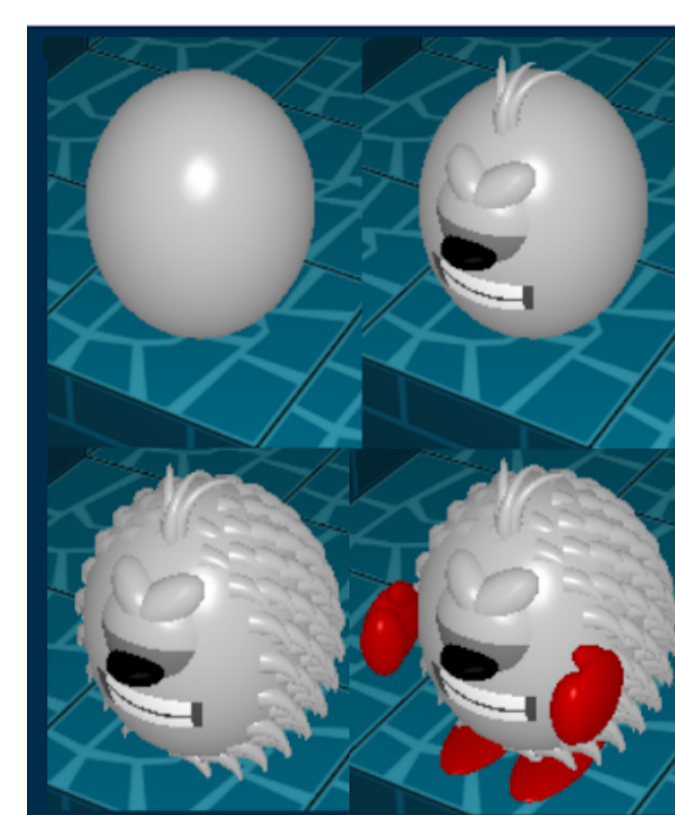


Figura 4: Ilustração das diferentes etapas de formação do -Player-.

Referências:

[1] [https://en.wikipedia.org/wiki/Catherine_\(video_game\)](https://en.wikipedia.org/wiki/Catherine_(video_game))

[2] https://www.khronos.org/opengl/wiki/Getting_Started#Writing_an_OpenGL_Application

[3] <https://www.libsdl.org/>