

JOGO: FLAPPY GHOST

Alunos: Matheus William Motta Santos (29351219), João Pedro Mourão Marques (29259169), Gustavo Martins Figueiredo (29527741)

Professora orientadora: Kadidja Valéria Reginaldo de Oliveira

Centro Universitário do Distrito Federal UDF, Coordenação de Tecnologia da Informação

RESUMO:

Flappy Bird é um jogo eletrônico simples e viciante no qual o jogador controla um pássaro que precisa atravessar uma série de obstáculos representados por pares de canos, evitando colisões enquanto acumula pontos a cada passagem bem-sucedida. Optamos por fazer um jogo como esse, porém utilizando outras animações, ilustrações e temáticas, mas claro, mantendo a mesma lógica e jogabilidade. O jogo é composto por elementos essenciais como um cenário em movimento que cria a ilusão de deslocamento, o personagem que responde a forças físicas como gravidade e impulso, obstáculos gerados aleatoriamente para aumentar o desafio, e um sistema de pontuação que motiva o jogador a avançar. Sua mecânica é centrada na simplicidade: ao clicar ou tocar na tela, o fantasma ganha altura temporariamente, o que exige precisão para manter um equilíbrio entre subir e desviar dos canos. Computacionalmente, o jogo explora conceitos fundamentais de algoritmos, como a aplicação de física para simular movimento, a randomização para criar desafios únicos, a detecção de colisões para definir o término da partida e a interação em tempo real com o usuário, demonstrando a integração entre lógica de programação e elementos visuais de forma prática e acessível.

PALAVRAS-CHAVE:

Flappy Bird, Flappy Ghost, programação, C++, Unity, jogo 2D, gravidade, pontuação, algoritmos, resposta a eventos, colisão, controle por clique/toque, geração randômica, simulação física.

ABSTRACT:

Flappy Bird is a simple and addictive electronic game in which the player controls a bird that must navigate through a series of obstacles represented by pairs of pipes, avoiding collisions while accumulating points with each successful pass. We chose to create a similar game, but using different animations, illustrations, and themes, while of course maintaining the same logic and gameplay. The game consists of essential elements such as a moving background that creates the illusion of movement, a character that responds to physical forces like gravity and thrust, randomly generated obstacles to increase the challenge, and a scoring system that motivates the player to progress. Its mechanics are centered around simplicity: by clicking or tapping on the screen, the ghost temporarily gains height, requiring precision to maintain a balance between rising and avoiding the pipes. Computationally, the game explores fundamental algorithm concepts, such as applying physics to simulate movement, randomization to create unique challenges, collision detection to define the end of the game, and real-time interaction with the user, demonstrating the integration of programming logic

and visual elements in a practical and accessible way. Tower of Hanoi, C programming, algorithms, dynamic memory allocation, programming best practices.

1. ANÁLISE

1.1. PROBLEMÁTICA

A análise do projeto "*Flappy Ghost*" envolve uma reflexão profunda sobre os desafios, a estrutura e a abordagem adotada para resolver os problemas inerentes ao desenvolvimento do jogo. A partir dessa análise, é possível identificar as questões mais relevantes que o sistema precisa solucionar, os métodos escolhidos para lidar com elas e as limitações que surgem durante a implementação.

1.2. COMPREENSÃO E DESAFIOS DO PROBLEMA

O principal problema a ser abordado no desenvolvimento do "*Flappy Ghost*" é a simulação de um jogo com física e mecânica simples, mas que seja ao mesmo tempo dinâmico e desafiador. Inspirado no clássico "*Flappy Bird*", o objetivo do jogo é controlar um fantasma que deve passar por entre obstáculos, sem colidir com eles. Esse tipo de mecânica parece simples, mas envolve uma série de desafios relacionados à dinâmica de movimento, geração aleatória de obstáculos e o gerenciamento de colisões. A compreensão desse problema exige uma análise detalhada do comportamento do personagem no espaço, a interação entre ele e os obstáculos, além da necessidade de implementar uma experiência de jogo que seja tanto acessível quanto divertida. Os desafios principais incluem garantir que o movimento do fantasma seja fluido, que a geração de obstáculos seja eficiente, e que o jogador tenha uma experiência justa, mas ao mesmo tempo desafiadora. A dificuldade reside na criação de um equilíbrio entre esses fatores, de modo que o jogo não se torne monótono ou excessivamente difícil, mas também não seja fácil demais.

1.3. MODELO MATEMÁTICO

O modelo matemático utilizado no desenvolvimento de "*Flappy Ghost*" baseia-se na física simplificada de movimento e interação do personagem com o ambiente do jogo, além de técnicas de detecção de colisões para garantir uma jogabilidade fluida. O movimento do fantasma, a geração dos obstáculos e as colisões são modelados utilizando algumas fórmulas matemáticas fundamentais, com o objetivo de criar uma experiência divertida e desafiadora para o jogador.

Para simular a gravidade no movimento do fantasma, utilizamos uma equação do movimento uniformemente acelerado (MUA). A posição vertical do fantasma no tempo t é dada por:

$$y(t) = y_0 + v_0 * t + \frac{1}{2} a * t^2$$

Onde:

- $y(t)$ é a posição vertical do fantasma no tempo t ,
- y_0 é a posição inicial do fantasma no jogo,
- v_0 é a velocidade inicial do fantasma (que pode ser zero ou um valor pequeno no início),
- a é a aceleração (no caso da gravidade, um valor negativo que representa a aceleração para baixo),
- t é o tempo.

Durante o jogo, o jogador pode interagir para impulsionar o fantasma para cima, o que altera a aceleração do movimento. Esse impulso pode ser modelado pela alteração instantânea na velocidade vertical do fantasma. A equação que descreve o movimento do fantasma, levando em consideração o impulso dado pelo jogador, pode ser representada por:

$$v(t) = v_0 - g * t + v * \text{impulso}$$

Onde:

- $v(t)$ é a velocidade vertical do fantasma no tempo t ,
- g é a aceleração devido à gravidade (um valor ajustado para simular a força de atração do solo no jogo),
- $v * \text{impulso}$ é o valor de impulso aplicado quando o jogador interage.

Além disso, a detecção de colisões é fundamental para garantir que o jogo responda corretamente quando o fantasma colide com os obstáculos. No jogo, os obstáculos são representados por retângulos, assim como o próprio fantasma. A colisão ocorre quando as áreas dos retângulos do fantasma e do obstáculo se sobrepõem. A verificação de colisão entre dois retângulos é feita pelas seguintes condições:

$$x * \text{fantasma} + w * \text{fantasma} > x * \text{obstáculo} \text{ E } x * \text{fantasma} < x * \text{obstáculo} + w * \text{obstáculo}$$

$$y * \text{fantasma} + h * \text{fantasma} > y * \text{obstáculo} \text{ E } y * \text{fantasma} < y * \text{obstáculo} + h * \text{obstáculo}$$

Onde:

- $x * \text{fantasma}$ e $y * \text{fantasma}$ são as coordenadas do canto superior esquerdo do retângulo que representa o fantasma,
- $w * \text{fantasma}$ e $h * \text{fantasma}$ são a largura e altura do retângulo do fantasma,
- $x * \text{obstáculo}$ e $y * \text{obstáculo}$ são as coordenadas do canto superior esquerdo do retângulo que representa o obstáculo,
- $w * \text{obstáculo}$ e $h * \text{obstáculo}$ são a largura e altura do retângulo do obstáculo.

Se as condições acima forem verdadeiras, significa que há uma interseção entre os retângulos e, portanto, uma colisão ocorreu.

A movimentação dos obstáculos é outro aspecto importante do jogo, pois eles se movem de forma contínua da direita para a esquerda da tela, criando o desafio de evitar a colisão. A posição horizontal de um obstáculo em função do tempo é dada pela fórmula:

$$x * \text{obstáculo}(t) = x * \text{obstáculo inicial} - v * \text{obstáculo} * t$$

Onde:

- $x * \text{obstáculo}(t)$ é a posição horizontal do obstáculo no tempo t ,
- $x * \text{obstáculo inicial}$ é a posição inicial do obstáculo quando ele aparece na tela,
- $v * \text{obstáculo}$ é a velocidade com que o obstáculo se move em direção ao jogador,
- t é o tempo desde que o obstáculo começou a se mover.

Essa fórmula é essencial para garantir que os obstáculos se movam de forma consistente e previsível, permitindo ao jogador antecipar os movimentos necessários para evitar as colisões.

Além disso, a pontuação no jogo é incrementada conforme o jogador consegue passar pelos obstáculos. A pontuação pode ser modelada como um simples contador, que é incrementado cada vez que o fantasma passa com sucesso por um obstáculo, sendo dada por:

$$\text{pontuação} = \text{pontuação inicial} + 1 \text{ se o fantasma ultrapassou um obstáculo}$$

Por fim, a animação do movimento do fantasma, especialmente para simular o "pulo" ou a flutuação, pode ser modelada por funções matemáticas como uma função sinusoidal ou quadrática, que criam a sensação de um movimento contínuo e natural. Uma fórmula possível para a animação do fantasma pode ser:

$$y * \text{fantasma}(t) = y * \text{inicial} + A * \sin(\omega * t)$$

Onde:

- A é a amplitude da oscilação, que determina a altura máxima do pulo,
- ω é a frequência da oscilação, que determina a velocidade do movimento,
- t é o tempo.

Essa fórmula permite que o movimento do fantasma seja fluido e que a sensação de pulo seja mais realista e contínua durante a interação do jogador.

Essas fórmulas matemáticas fornecem a base para a física do movimento, a detecção de colisões e a dinâmica do jogo. Elas são essenciais para garantir uma experiência de jogo consistente e desafiante, ao mesmo tempo em que mantêm o jogo acessível e divertido para o jogador.

2. PLANEJAMENTO

2.1. OBJETIVOS DO ALGORITMO

O planejamento do desenvolvimento de *Flappy Ghost* foca na criação de um jogo funcional e divertido, integrando conceitos de algoritmos para modelar a jogabilidade e os desafios enfrentados pelo jogador. O primeiro aspecto a ser definido são os objetivos do algoritmo, que incluem: simular o movimento do fantasma, gerar obstáculos de forma dinâmica, calcular colisões com precisão e gerenciar a pontuação. Esses objetivos garantem que o jogo seja consistente, responsivo e que a dificuldade evolua de maneira progressiva.

2.2. MÉTRICAS ESTABELECIDAS

Para avaliar a eficácia do algoritmo, foram estabelecidas métricas, como o tempo médio de execução de cada frame, a precisão na detecção de colisões e a consistência no comportamento do movimento dos objetos. Outra métrica importante é a fluidez percebida durante a jogabilidade, medida pela ausência de travamentos ou atrasos durante as interações do jogador.

2.3. ESTRATÉGIA GERAL

A estratégia geral baseia-se em dividir o jogo em componentes independentes, como movimentação do fantasma, geração dos obstáculos, verificação de colisões e cálculo de pontuação. Esses componentes interagem através de funções bem definidas, minimizando dependências e facilitando ajustes e melhorias. A abordagem também busca equilibrar

simplicidade e eficiência, utilizando métodos de otimização quando necessário, como limitar o número de cálculos por frame.

2.4. SUBPROBLEMAS IDENTIFICADOS

Durante a elaboração do algoritmo, foram identificados vários subproblemas, como o ajuste da gravidade e do impulso aplicado ao fantasma, a disposição aleatória dos obstáculos sem criar barreiras impossíveis, e a sincronização do movimento horizontal dos obstáculos com a velocidade de jogo. Outro subproblema notável foi implementar uma detecção de colisão eficiente que funcione em tempo real, sem comprometer o desempenho.

2.5. ESTRUTURA GERAL DO ALGORITMO

A estrutura geral do algoritmo foi organizada em ciclos contínuos de atualização e renderização, típicos de jogos baseados em frames. O ciclo principal engloba funções para processar entradas do jogador, atualizar estados dos objetos no jogo e redesenhar a tela. Dentro dessa estrutura, os componentes interagem por meio de variáveis globais e funções específicas para manter o estado do jogo de forma consistente entre os ciclos.

2.6. CASOS LIMITE/SITUAÇÕES ESPECIAIS

Casos limite e situações especiais foram analisados cuidadosamente, como o que ocorre se o jogador permanecer inativo, se os obstáculos se acumularem de forma incoerente ou se o fantasma ultrapassar os limites visíveis da tela. Outro caso avaliado foi a gestão de frames por segundo em máquinas mais lentas, garantindo que o jogo funcione de forma semelhante independentemente do hardware.

2.7. ANÁLISE TEÓRICA

A análise teórica do algoritmo considerou a complexidade computacional das operações mais críticas, como a detecção de colisões, que foi projetada para ter eficiência linear em relação ao número de obstáculos ativos. Além disso, os cálculos relacionados à física do jogo foram simplificados para reduzir o custo computacional, sem comprometer a experiência do jogador. A análise também incluiu cenários extremos, como aumento na dificuldade com o tempo, para garantir que o jogo seja desafiador, mas justo, mesmo em sessões mais longas.

3. CONCLUSÃO

O desenvolvimento do jogo *Flappy Ghost* proporcionou uma experiência rica em aprendizado, que envolveu a aplicação de conceitos matemáticos, lógicos e de design de jogos. Ao longo do processo, foi possível explorar a construção de um sistema de física simplificada, focado principalmente no movimento do personagem e na interação com os obstáculos, além de lidar com desafios técnicos e criativos que surgiram durante a implementação.

Um dos maiores aprendizados ao longo do desenvolvimento foi a importância de entender como aplicar fórmulas matemáticas de maneira eficaz para simular a física de movimento no jogo. A utilização de equações para a aceleração e a movimentação do fantasma, bem como a implementação de fórmulas de colisão, foram fundamentais para garantir que o jogo tivesse um comportamento consistente e uma jogabilidade fluida. Além disso, a prática de gerar obstáculos de forma aleatória e animá-los de maneira dinâmica foi um exercício importante em termos de algoritmos de movimentação e controle de tempo, essenciais para a criação de desafios progressivos no jogo. Outro aspecto relevante foi a importância de balancear a dificuldade. Ajustar os parâmetros do jogo, como a velocidade dos obstáculos e a intensidade

da gravidade, foi crucial para criar uma experiência desafiadora, mas que ao mesmo tempo fosse justa e agradável para o jogador. Isso envolveu testes contínuos e ajustes finos para garantir que o jogo fosse desafiador sem se tornar frustrante.

Durante o desenvolvimento, surgiram várias dificuldades, especialmente no que diz respeito ao ajuste da física de movimento. Encontrar um valor adequado para a aceleração da gravidade e para a força do impulso aplicado pelo jogador foi um processo que exigiu vários testes e ajustes. O controle do movimento do fantasma, de forma que ele não se movesse nem muito rápido nem muito devagar, foi um dos maiores desafios, pois influenciava diretamente na jogabilidade e na sensação de fluidez do jogo. Outro desafio significativo foi a detecção de colisões. A princípio, a implementação de colisões não era completamente precisa, resultando em falhas na interação entre o fantasma e os obstáculos. Ajustar os parâmetros de detecção e garantir que as colisões fossem verificadas de forma eficaz sem causar "falsas colisões" foi um trabalho que exigiu um bom entendimento de como os retângulos e as coordenadas funcionam no espaço 2D. Além disso, a geração aleatória dos obstáculos, embora parecesse simples, exigiu uma abordagem cuidadosa para garantir que os obstáculos fossem distribuídos de maneira interessante, sem criar espaços muito largos ou muito estreitos entre eles, o que afetaria a jogabilidade e a diversão.

Um ponto que se destacou durante o desenvolvimento foi a importância de realizar testes constantes. A interação do jogador com o jogo precisa ser intuitiva e agradável, e isso só pode ser validado por meio de testes contínuos, ajustando variáveis conforme o feedback gerado. A adaptação da física e da geração dos obstáculos foi iterativa, sendo refinada ao longo do processo até atingir um nível de satisfação quanto à jogabilidade. Além disso, o projeto reforçou a relevância de uma abordagem modular na programação. Organizar o código em funções específicas para cálculo da física, detecção de colisões, movimentação dos obstáculos e pontuação facilitou a manutenção e a evolução do projeto. Uma estrutura modular bem definida não só permitiu que novas funcionalidades fossem adicionadas com facilidade, como também ajudou a identificar e corrigir erros de forma mais rápida. Em termos de design, o jogo *Flappy Ghost* mostrou que é possível criar uma experiência envolvente com uma mecânica simples e uma abordagem minimalista. A jogabilidade é baseada em um conceito fácil de entender, mas que ainda assim oferece um desafio constante ao jogador, o que é um bom exemplo de como simplicidade pode ser poderosa quando bem executada.

Em resumo, o projeto *Flappy Ghost* foi uma experiência valiosa que envolveu aprendizado técnico e criativo, além de desafios práticos que ajudaram a aprimorar habilidades na área de programação, design de jogos e resolução de problemas. Embora tenha sido desafiador, o processo de desenvolvimento contribuiu significativamente para a compreensão dos conceitos fundamentais por trás da construção de jogos, principalmente no que se refere à física de movimento, detecção de colisões e manipulação de elementos aleatórios dentro do ambiente de jogo. Com base nas dificuldades enfrentadas e nas soluções encontradas, o projeto não só foi uma oportunidade de crescimento, mas também uma boa base para futuros desenvolvimentos e aprimoramentos.