**Bird The Game**

*Bird The Game*

Kadidja Valéria Reginaldo de Oliveira¹ Orientador

1 *Centro Universitário do Distrito Federal UDF, Curso Ciências da Computação, Brasília, DF*

2 *Centro Universitário do Distrito Federal UDF, Coordenação de Tecnologia da Informação, Brasília, DF*

**RESUMO:**

**Introdução**: Bird The Game é um jogo de ação no estilo arcade, onde o jogador controla um pássaro enfrentando inimigos em um cenário repleto de desafios. O objetivo é derrotar as aves inimigas, que atacam com ovos, enquanto o jogador revida com flechas. A cada fase, o desafio aumenta, com novos inimigos e padrões de ataque. O jogo, inspirado nos clássicos dos anos 80 e 90, oferece uma jogabilidade rápida e dinâmica, com controles simples e uma experiência envolvente, onde a precisão e a estratégia são fundamentais para avançar.

**Palavras-chave:** Arcade, Pássaro, Inimigos, Flechas

**ABSTRACT:**

**Introduction**: Bird The Game is an action arcade game where the player controls a bird facing enemies in a challenging environment. The goal is to defeat enemy birds, which attack with eggs, while the player retaliates with arrows. As the player progresses, the difficulty increases with new enemies and attack patterns. Inspired by the classics of the 80s and 90s, the game offers fast-paced and dynamic gameplay, with simple controls and an engaging experience where precision and strategy are key to advancing.

**Keywords:** Arcade, Bird, Enemies, Arrows

Orientadora: Profa. Kadidja Valéria Reginaldo de Oliveira

**ANÁLISE**

O jogo é um arcade inspirado nos clássicos do gênero, no qual o jogador controla um pássaro determinado a superar desafios em um cenário repleto de inimigos. Os principais oponentes do jogador são outras aves que atacam constantemente com ovos, enquanto o pássaro revida lançando suas próprias flechas. O objetivo é derrotar todos os adversários presentes na fase para avançar, enfrentando desafios progressivamente mais difíceis. A estética remete aos jogos de ação rápidos e dinâmicos dos anos 80 e 90. Cada nível apresenta adversários com padrões de ataque variados e visuais que estimulam o jogador a continuar mantendo a experiência desafiadora e recompensadora.

A natureza do problema está em capturar a essência dos clássicos arcades, entregando uma experiência de ação rápida e intuitiva. Os desafios incluem equilibrar a dificuldade para garantir que o jogo seja acessível e, ao mesmo tempo, desafiador, criando controles precisos e ataques responsivos para o jogador. Além disso, os adversários têm visuais marcantes inspirados em animais, projetados para não sobrecarregar o cenário, mantendo a jogabilidade clara e fluida. Outro aspecto é a criação de uma experiência envolvente dentro da demo, com mecânicas bem definidas e adversários variados que ofereçam um equilíbrio entre desafio e diversão, mantendo o jogador engajado ao longo de toda a partida.

O modelo teórico para o jogo "Bird The Game" pode ser representado pela combinação da Teoria de Sistemas Dinâmicos e Algoritmos de Detecção de Colisão. A Teoria de Sistemas Dinâmicos é relevante, pois o jogo descreve a interação entre várias entidades (como o pássaro, projéteis e adversários) dentro de um sistema dinâmico de eventos. Variáveis como posição, velocidade e vida de cada entidade são atualizadas a cada iteração do loop principal, criando uma simulação baseada no tempo. O comportamento do jogo é governado por essas interações, que são modeladas ao longo de ciclos temporais.

Por outro lado, a Detecção de Colisão é fundamental para o funcionamento do jogo. No caso de "Bird The Game", a colisão entre os projéteis e os adversários pode ser detectada usando máscaras de pixel ou verificações geométricas, como o uso de retângulos delimitadores (bounding boxes). Para duas entidades, AAA e BBB, a colisão ocorre se houver interseção entre suas caixas delimitadoras.

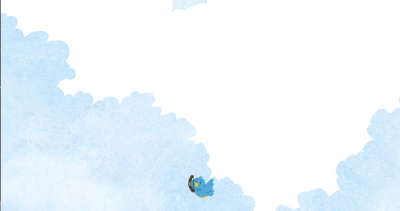
**PLANEJAMENTO**

O objetivo principal do algoritmo é controlar a interação entre o pássaro e os adversários, garantindo que os ataques sejam realizados corretamente e que a movimentação do pássaro seja fluida e responsiva. O algoritmo deve assegurar que os ataques do pássaro atinjam os adversários corretamente e que a mecânica de derrota dos adversários funcione conforme o esperado, sem envolver colisões entre o pássaro e os adversários, já que o pássaro apenas ataca e não se move para frente.

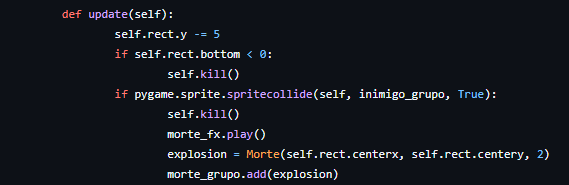


As métricas para avaliação de eficiência do algoritmo do jogo "Bird The Game" incluem o desempenho, o uso de recursos e a precisão. O desempenho é medido pelo tempo de resposta do jogo entre as entradas do jogador e as atualizações visuais, como o movimento do pássaro e os efeitos de ataque. O uso de recursos analisa a quantidade de memória e poder de processamento necessários para calcular os movimentos, ataques e detecção de colisões, especialmente considerando a quantidade de adversários e projéteis na tela. Já a precisão é avaliada pela taxa de acerto na detecção de colisões entre os projéteis e os adversários, garantindo que o jogo funcione corretamente, sem falhas na interação entre as entidades.

A estratégia geral de resolução do problema proposta para o jogo "Bird The Game" baseia-se em um modelo de evento dinâmico, onde as interações entre o pássaro, os adversários e os projéteis são atualizadas em ciclos regulares. O algoritmo monitora constantemente a posição do pássaro e a entrada do jogador, permitindo que o pássaro se mova e ataque conforme o controle. Os adversários, por sua vez, atacam de maneira automática, e suas ações são programadas para manter o desafio. O sistema de detecção de colisão verifica se os ataques do pássaro atingiram os adversários ou se os adversários acertaram o pássaro, com base nas posições e no tempo. Além disso, o algoritmo gerencia o estado de vitória ou derrota, avançando o jogador para a próxima fase quando todos os adversários forem derrotados, ou finalizando o jogo caso o pássaro seja atingido.

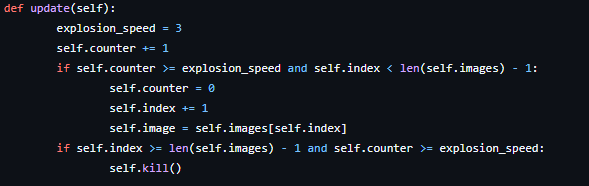


**DESENHO**

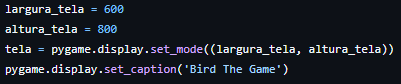
A complexidade temporal de um jogo geralmente depende de quantas operações o sistema precisa executar a cada quadro, que é chamado de frame. Neste jogo, a principal operação é o loop de atualização, que verifica a colisão, atualiza as posições dos personagens e projéteis, além de desenhar as imagens.

**Atualização do jogador e projéteis**: O movimento do jogador e o disparo de flechas são eventos simples, mas cada projétil gerado é processado individualmente.

**Atualização dos inimigos**: O movimento dos inimigos é verificado, assim como suas colisões com as flechas e com o jogador.

**Detecção de colisões**: A verificação de colisões entre o pássaro, projéteis e inimigos é feita usando pygame.sprite.spritecollide(). A complexidade dessa operação é **O(n)**, onde **n** é o número de inimigos ou projéteis. Como cada colisão é verificada em cada quadro para cada entidade no jogo, a complexidade total por quadro será **O(n + m + p)**, onde **n** é o número de inimigos, **m** é o número de projéteis inimigos, e **p** é o número de flechas disparadas pelo jogador. 

**Desenho da tela**: O processo de renderização envolve desenhar todos os elementos do jogo (jogador, inimigos, projéteis, etc.), o que também tem uma complexidade linear em relação ao número de elementos na tela.

****

**PROGRAMAÇÃO E TESTE**

Durante os testes, foi identificado que, após o usuário ganhar ou perder uma partida, o jogo simplesmente parava sem exibir nenhuma tela indicando vitória ou derrota. Esse comportamento foi causado pela ausência de uma verificação explícita do estado de vitória no final da fase.

Outro erro ocorreu quando a flecha do pássaro foi inserida no jogo. Inicialmente, a flecha estava configurada com um tamanho muito grande, o que fazia com que ela atingisse mais de um inimigo ao mesmo tempo, resultando em múltiplos adversários sendo derrotados simultaneamente. Esse problema foi corrigido ajustando o tamanho da flecha para garantir que ela atingisse apenas um inimigo por vez, mantendo o equilíbrio e a integridade do jogo.

**DOCUMENTAÇÃO E AVALIAÇÃO DO PROJETO**

O jogo utiliza a biblioteca pygame, com o objetivo de criar uma experiência dinâmica e desafiante para o jogador. O algoritmo principal do jogo envolve a interação entre o pássaro controlado pelo jogador, inimigos que atacam com ovos, e projéteis disparados pelo jogador.

**Estrutura do Algoritmo:**

**Inicialização:** O jogo inicia com a configuração da tela e a inicialização dos sons e fontes.

**Criação de Entidades:**

Pássaro: O personagem principal do jogador, que se move horizontalmente e dispara flechas.

Ovos: Projéteis disparados pelo pássaro.

Inimigos: Entidades controladas por IA que se movem e atacam o jogador com ovos.

Ovos Inimigos: Projéteis disparados pelos inimigos.

Morte: Efeitos visuais de explosão quando uma entidade morre.

* **Lógica de Jogo:**
  + O jogador controla o pássaro movendo-o para os lados e disparando flechas para destruir os inimigos.
  + Inimigos se movem automaticamente e disparam ovos em direção ao pássaro.
  + A colisão entre os projéteis e as entidades é verificada a cada quadro.
* **Controle de Fases:** O jogo apresenta fases em que o número de inimigos varia, e o jogador deve derrotar todos os inimigos para avançar.
* **Final do Jogo:** O jogo termina quando todos os inimigos são derrotados ou quando o pássaro é atingido pelos ovos inimigos.

**Análise de Complexidade:**

* **Espaço:** A complexidade espacial é dominada pelas listas de grupos de entidades que armazenam os objetos do jogo. Cada grupo pode conter até 25 inimigos, e o uso de memória é linear em relação ao número de entidades presentes na tela.
* **Tempo:** A principal complexidade temporal ocorre nas verificações de colisão e no loop principal do jogo, que executa operações como o movimento dos personagens, detecção de colisões e atualizações gráficas em cada quadro. O tempo de execução de cada ciclo é constante, mas o número de operações cresce linearmente com o número de objetos na tela (inimigos, projéteis, etc.), resultando em uma complexidade de tempo O(n), onde n é o número de entidades.

**Avaliação da Eficácia do Algoritmo em Termos de Tempo de Execução, Uso de Recursos e Precisão na Resolução do Problema**

* **Tempo de execução:** O algoritmo executa em tempo real, com uma taxa de quadros de 60 FPS, o que garante uma experiência fluida para o jogador. O tempo de execução depende do número de entidades na tela, mas permanece dentro de limites aceitáveis devido à complexidade linear.
* **Uso de recursos:** O uso de memória e CPU é eficiente para um jogo de arcade simples. A alocação de memória é baseada no número de entidades, e o jogo consegue funcionar bem em sistemas com recursos limitados, como PCs mais antigos ou emuladores.
* **Precisão:** A detecção de colisões é precisa, utilizando máscaras de colisão para garantir que os projéteis interajam corretamente com os inimigos. No entanto, o tamanho da flecha foi ajustado após um erro inicial, garantindo que ela atingisse apenas um inimigo por vez

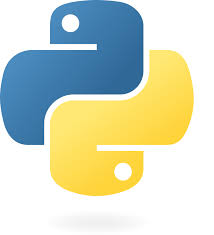
**CONCLUSÃO**

**Bird The Game** é um jogo de arcade inspirado nos clássicos dos anos 80 e 90, onde o jogador controla um pássaro que deve derrotar inimigos com flechas. O jogo oferece uma experiência dinâmica e desafiadora, com uma jogabilidade simples, mas envolvente, equilibrando acessibilidade e desafio.

O desenvolvimento focou em garantir uma jogabilidade fluida e precisa, utilizando a **Teoria de Sistemas Dinâmicos** e algoritmos de **detecção de colisão** para interações corretas entre as entidades no jogo. A análise de **complexidade temporal e espacial** mostrou que o jogo é eficiente, mantendo uma taxa de 60 FPS e bom desempenho, mesmo em sistemas mais modestos.

Durante os testes, ajustes importantes foram feitos, como a correção do tamanho da flecha e a adição de telas de vitória ou derrota, melhorando a experiência do jogador. O algoritmo foi eficaz em termos de **tempo de execução**, **uso de recursos** e **precisão** na detecção de colisões, proporcionando uma jogabilidade sem falhas.

Em resumo, **"Bird The Game"** oferece uma experiência sólida, desafiadora e divertida, mantendo a essência dos jogos de arcade clássicos com um bom equilíbrio entre jogabilidade e desempenho técnico.

**TECNOLOGIAS USADAS**

**REFERÊNCIAS BIBLIOGRÁFICAS**

BRODIS, G. **Como Os jogos dos fliperamas Eram feitos? (arcades)**. Disponível em: <https://www.youtube.com/watch?v=FOSl6PnBhtc>. Acesso em: 10 nov. 2024.

**Pygame Front Page — pygame v2.6.0 documentation**. Disponível em: <https://www.pygame.org/docs/>. Acesso em: 10 nov. 2024.

**O tutorial do Python**. Disponível em: <https://docs.python.org/pt-br/3/tutorial/>. Acesso em: 11 nov. 2024.

SILVA, F. **Arcade**. Disponível em: <https://bdjogos.com.br/console.php?id=6&g=7>. Acesso em: 11 nov. 2024.

GAMEDEV, C. E. **6 JOGOS Feitos na GODOT para te INSPIRAR! - Semana 46**. Disponível em: <https://www.youtube.com/watch?v=BzHOCHKiFu0>. Acesso em: 10 nov. 2024.