

UNIVERSIDADE DA AMAZÔNIA - UNAMA

RELATÓRIO TÉCNICO: PACUBOS

Autores:

João Carlos Viana Teixeira — Matrícula: 04194916

Tobias Nunes de Souza — Matrícula: 04179379

Resumo

O presente relatório descreve o desenvolvimento de "Pacubos", um jogo de estilo Pac-Man desenvolvido em Python utilizando a biblioteca Pygame para a disciplina de Tópicos Especiais. O projeto seguiu as diretrizes do PyGaming Hub, não usando ativos visuais externos e empregando exclusivamente primitivas geométricas para a renderização. A configuração do jogo é externa, via `conf/conf.ini`. Mecânicas principais incluem coleta de pellets, múltiplos inimigos (fantasmas), powerups (speed/giant/power) e um sistema de leaderboard persistente com entrada de iniciais.

Abstract

This report documents the development of "Pacubos", a Pac-Man style game written in Python using Pygame for the Special Topics course. The project follows the PyGaming Hub rules (no external visual assets) and uses geometric primitives for rendering. Configuration is external via `conf/conf.ini`.

Sumário

1. Introdução
2. Especificações Técnicas
3. Desenvolvimento e Mecânicas
4. Persistência de Dados (I/O)
5. Manual do Usuário
6. Conclusão
7. Referências

1. Introdução

Este relatório descreve o desenvolvimento do jogo "Pacubos", criado como atividade da disciplina de Tópicos Especiais. O objetivo foi aplicar lógica de programação e utilizar Pygame para criar um jogo íntegro que respeite as restrições do PyGaming Hub: sem ativos externos, configuração por `conf.ini`, e com um `data.inf` contendo metadados.

2. Especificações Técnicas

2.1 Tecnologias Utilizadas

- Python 3.x
- Pygame (>=2.0)

- JSON para persistência (leaderboard)
- Git/GitHub para controle de versão

2.2 Arquitetura de Configuração

O jogo lê `conf/conf.ini` usando `configparser` e converte strings de teclas para constantes do Pygame com uma função utilitária (`key_from_name`), permitindo alteração de controles e resolução sem mudar código.

2.3 Renderização "Vanilla"

Não há imagens externas; todos os elementos são desenhados com primitivas: `pygame.draw.rect()`, `pygame.draw.circle()` e fontes do sistema.

3. Desenvolvimento e Mecânicas

3.1 Dinâmica de Jogo

O jogador controla um quadrado que coleta pellets em uma grade; fantasmas vermelhos são inimigos. Colidir com fantasmas reduz vidas; quando vulneráveis, os fantasmas podem ser derrotados.

3.2 Powerups e Efeitos

Powerups: **power** (fantasmas vulneráveis), **speed** (velocidade aumentada), **giant** (tamanho aumentado). Cada powerup tem duração configurada por constantes no código.

3.3 Inimigos (Fantasmas)

A classe `ghost` controla movimento, timers de mudança, vulnerabilidade e respawn. Novos fantasmas surgem com intervalos e existe um limite `GHOST_MAX`.

3.4 Paredes/Obstáculos

Paredes com tipos distintos (queda lenta/rápida e movimentos laterais) aparecem periodicamente e interagem com jogador e fantasmas, causando empurrões ou bloqueios.

3.5 Sistema de Leaderboard

Top-10 persistente em `games/Pacubos/leaderboard.json`. Ao perder todas as vidas, o jogador insere iniciais (3 letras) e o resultado é gravado.

4. Persistência de Dados (I/O)

Leaderboard gravado/ler com JSON. O arquivo é atualizado ao fim da partida quando necessário.

5. Manual do Usuário

5.1 Controles (padrão)

- Movimento: W A S D (ou setas)
- Action A: O
- Action B / Pause: P
- Restart: R

5.2 Como Jogar

Inicie no menu com ENTER. Colete pellets para pontuar, colete powerups para vantagem, evite perder todas as vidas. Ao fim, insira iniciais para salvar no ranking.

6. Conclusão

Pacubos atende aos requisitos do PyGaming Hub: utiliza apenas primitivas gráficas, configurações externas e inclui persistência de scores. O projeto consolidou habilidades em programação, máquinas de estado e I/O.

7. Referências

- Documentação Pygame — <https://www.pygame.org/docs/>
- Guia de Contribuição: PyGaming Hub