Universidade Federal de Minas gerais

PROGRAMAÇÃO DE SOFTWARE II

PLAY

NOSSA EQUIPE

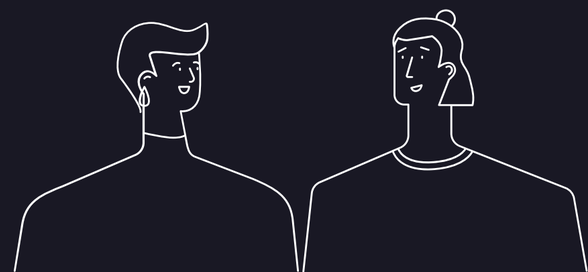
ALINE WINDERS BARBOSA

ANDRÉ LUIZ ALVES COSTA

CAMILA SILVA ALVES

MATHEUS ALVES KÜHL

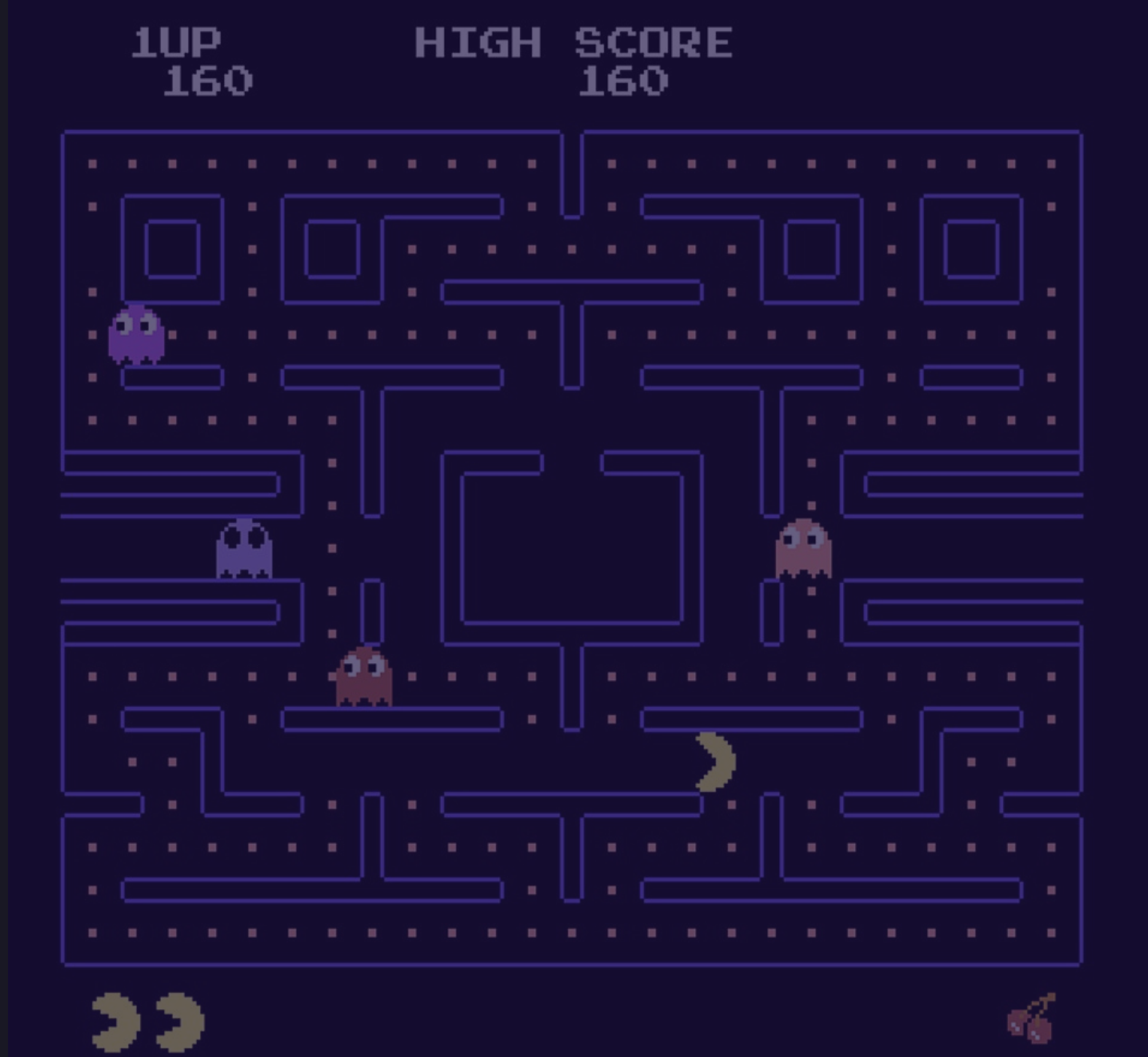
VICTOR PRATES FIGUEIREDO



SOBRE O JOGO

Pacman é um jogo eletrônico desenvolvido pela empresa Namco.

O objetivo do jogador é percorrer um labirinto, comendo pontos normais e de energia, enquanto o ele foge de quatro fantasmas.



MODELAGEM

- Cartões CRC
- Histórias de usuário

Notion

METODOLOGIA ÁGIL KANBAN

- Maior dinamismo entre as pessoas
- Promove um ambiente de colaboração
- Cria independência
- Melhora a visualização dos problemas
- Otimiza o tempo
- O planejamento diário é mais eficaz

KANBAN É UM QUADRO DE SINALIZAÇÃO QUE CONTROLA OS FLUXOS DE PRODUÇÃO

TECNOLOGIAS E FERRAMENTAS

C++

Linguagem de Programação usada para desenvolvimento do projeto.

SFML

Biblioteca de desenvolvimento de jogos em C++.

Plataforma de gerenciamento de código.

Editor de código. Foram utilizadas extensões para C++ do VS code para auxiliar o desenvolvimento.

Princípio da responsabilidade única

Princípios
SOLID

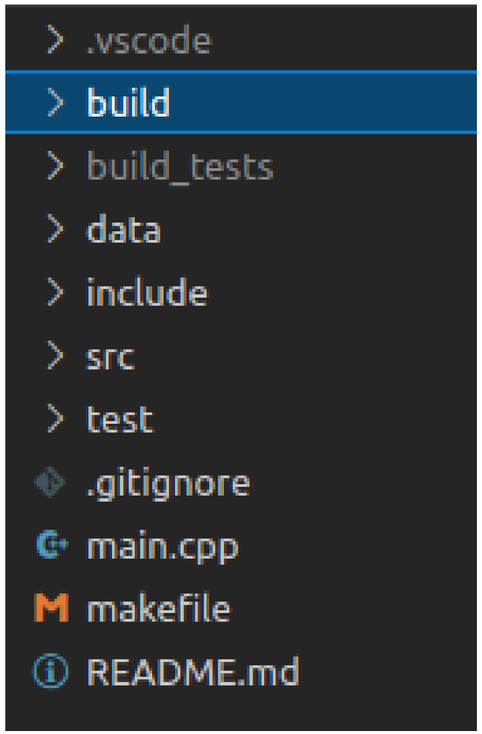
Princípio de inversão de dependencia

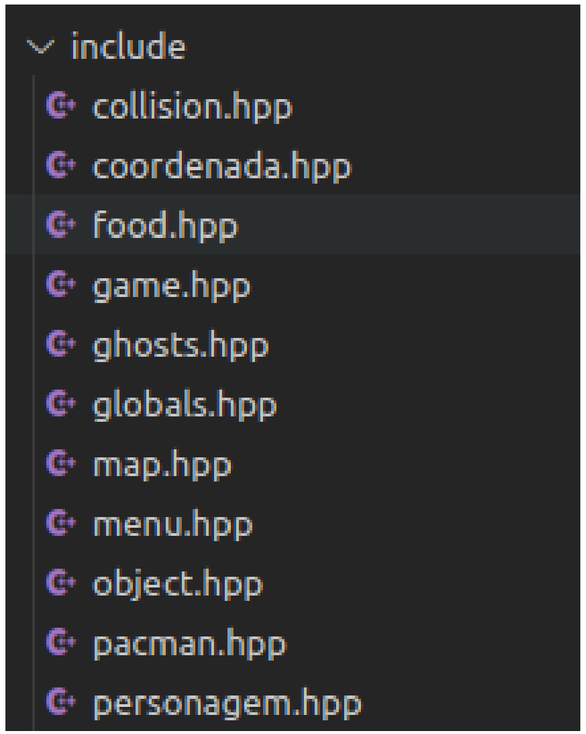
Princípio de abertura-fechamento

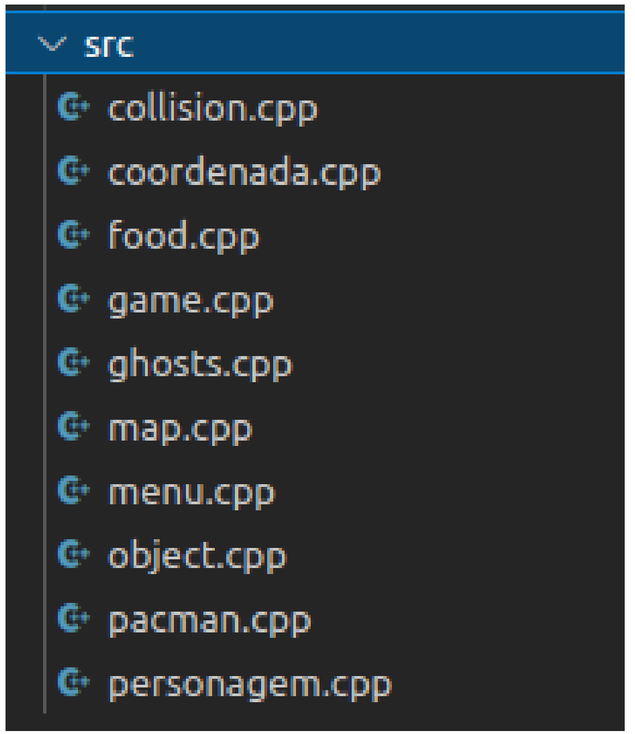
Princípio da segregação de interface

Princípio da substituição de Liskov

O PROJETO


.vscode
build
build_tests
data
include
src
test
.gitignore
main.cpp
makefile
README.md


include
collision.hpp
coordenada.hpp
food.hpp
game.hpp
ghosts.hpp
globals.hpp
map.hpp
menu.hpp
object.hpp
pacman.hpp
personagem.hpp


src
collision.cpp
coordenada.cpp
food.cpp
game.cpp
ghosts.cpp
map.cpp
menu.cpp
object.cpp
pacman.cpp
personagem.cpp

TRABALHANDO COM CONTRATO

Contratos bem definidos facilitam o entendimento do código e garantem um bom funcionamento de acordo com o solicitado.

```
...
#ifndef GLOBALS_H
#define GLOBALS_H

constexpr int MAP_HEIGHT = 21;
constexpr int MAP_WIDTH = 21;
constexpr int CELL_SIZE = 16;
constexpr int FOOD_POINTS = 10;
constexpr int DRUG_POINTS = 100;
constexpr int SCREEN_RESIZE = 2;
constexpr int FONT_HEIGHT = 16;
constexpr int TUNEL_HEIGHT = 9;
constexpr int PACMAN_SPEED = 1;

enum Type{
    blank,
    wall,
    food,
    drug,
    red,
    blue,
    pink,
    orange,
    pacman,
    door,
};

#endif
```

```
André Luiz, 4 hours ago | 1 author (André Luiz)
class Menu {
    private:
        int pos;
        bool pressed, theselect;

        sf::RenderWindow * window;
        sf::Font * font;
        sf::Texture * image;
        sf::Sprite * bg;

        std::vector<const char *> options;
        std::vector<sf::Vector2f> coords;
        std::vector<sf::Text> texts;
        std::vector<std::size_t> sizes;

    protected:
        void set_values();      André Luiz, yesterday • #01: Construindo
        void loop_events();
        void draw_all();

    public:
        Menu();
        ~Menu();
        void run_menu();
        void init_screen();
};
```

```
André Luiz, 8 hours ago | 3 authors (vprates-22 and others)
class Coordenada{
    private:
        int x;
        int y;

    public:
        std::pair<int,int> getPair();

        Coordenada();

        int get_X();
        int get_Y();
              André Luiz, 8 hours ago • #Refat
        void set_X(int _x);
        void set_Y(int _y);
};

#endif
```

```cpp
#ifndef PERSONAGEM_H
#define PERSONAGEM_H

#include "coordenada.hpp"
#include "food.hpp"
#include <vector>

using Mapa = std::vector<std::vector<Object*>>;        vprates-22, 3 days ago • Comentário + Tela Game O

class Personagem
```

André Lui ... prates-22 and others)

```cpp
class Personagem{
    protected:
        Coordenada coord_init; //coordenada aonde o pacman começa, será util quando ele morrer
        Coordenada coord_atual; //coordenada atual, usada no movimento do pacman

        short direction;

    public:
        virtual ~Personagem();

        int get_X();
        int get_Y();

        void set_X(int _x);
        void set_Y(int _y);

        int get_X_init();
        int get_Y_init();

        void set_X_init(int _x);
        void set_Y_init(int _y);

        //virtual void reviver()=0;
        virtual void mover(Mapa* map)=0;
        virtual void comer(Mapa* map)=0;

        virtual void setDirection(short dir) = 0;
        int getDirection();
};
```

```
André Luiz, 8 hours ago | 3 authors (vprates-22 and others)
1  #ifndef PACMAN_H
2  #define PACMAN_H
3
4  #include "personagem.hpp"
5  #include <string>
6  #include <ctime>
7
   André Luiz, 8 hours ago | 3 authors (vprates-22 and others)
8  class Pacman : public Personagem{
9      private:
10         unsigned score = 0;
11         short lifes = 3;
12         bool invencibility = false;
13       André Luiz, 8 hours ago • #Refatoração: Arrumando a identação
14         clock_t invencibilityTimer;
15
16     public:
17         Pacman();
18         //virtual void reviver() override;
19         virtual void mover(Mapa* map) override;
20         virtual void comer(Mapa* map) override;
21
22         void lose_life();
23         short get_lifes();
24
25         virtual void setDirection(short dir) override;
26
27         void setInvencibility();
28         void timeInvencibility();
29         bool getInvencibility();
30         void unsetInvensibility();
31
32         unsigned get_score();
33         void sumScore();
34 };
35
36 #endif
```

```
André Luiz, 8 hours ago | 2 authors (malviska and others)
#include"personagem.hpp"
#include"coordenada.hpp"
#include"collision.hpp"
#include<array>
#ifndef GHOST_H
#define GHOST_H

André Luiz, 8 hours ago | 2 authors (malviska and others)
class Ghost : public Personagem{
    private:
        bool isFrightened = false;

    public:
        Ghost();

        void update();

        void setIsFrightened();
        bool getIsFrightened();

        virtual void mover(Mapa* map) override;
        virtual void comer(Mapa* map) override;
        virtual void setDirection(short dir) override;
};

#endif      malviska, 2 days ago • red ghost, but if bugs …
```

```cpp
#ifndef OBJETO_H
#define OBJETO_H

#include "coordenada.hpp"
#include "globals.hpp"
```

André Luiz, 8 hours ago | 2 authors (vprates-22 and others)

```cpp
class Object{
    private:

    protected:
        Type type;

    public:
        Object();
        Object(Type tipo);
        virtual ~Object();

        int get_type_int();
        Type get_type();
};

#endif      vprates-22, yesterday • unir branch …
```

```cpp
#ifndef FOOD_H
#define FOOD_H

#include "object.hpp"

…
class Food : public Object{
    friend class Game;

    private:
        static int count;
        int points = FOOD_POINTS;

    public:
        Food();
        Food(Type tipo);

        int get_points();
        void eaten();
        bool is_food_over();
};


#endif
```

```cpp
#ifndef MAPA_H
#define MAPA_H

#include <iostream>
#include <fstream>
#include <string>
#include <cmath>
#include "pacman.hpp"
#include "ghosts.hpp"

using Mapa = std::vector<std::vector<Object*>>;

...
class Map{

    public:
        Mapa ler_mapa(std::ifstream&, Pacman&, Ghost&, Ghost&, Ghost&, Ghost&);
};

#endif
```

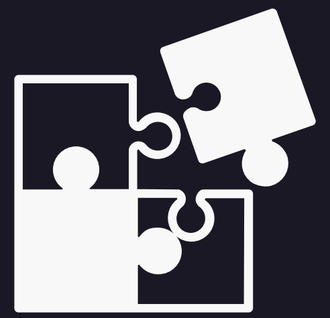
```
1   #include "globals.hpp"
2   #include"map.hpp"
3   #ifndef COLLISION_H
4   #define COLLISION_H
5
6   bool map_collision(bool i_collect_pellets,
7                      bool i_use_door, int x,
8                      int y,
9                      std::vector<std::vector<Object*>>& i_map);
10      malviska, 2 days ago • red ghost, but if bugs
11  #endif
```

```cpp
#ifndef GAME_H
#define GAME_H

#include "map.hpp"
#include <chrono>
#include <sstream>
#include <SFML/Graphics.hpp>
#include "ghosts.hpp"
#include"collision.hpp"

André Luiz, 8 hours ago | 3 authors (vprates-22 and others)
class Game{
    private:
        sf::RenderWindow *window;
        sf::VideoMode video_mode;
        sf::Event ev;

        sf::Font font;

        sf::Text uiText;
        sf::Text GOText;

        Pacman *pacman;
        Ghost *red;
        Ghost *blue;
        Ghost *pink;
        Ghost *orange;

        std::ifstream File;
        Mapa map_sketch;
        bool pacmanSituation = false;

        void initWindow();
        void initPacman();
        void initText();
        void initFonts();
```

```cpp
    public:
        Game();
        virtual ~Game();

        const bool running() const;        vprates-22, 3 days ago •
        bool getEndGame();

        void pollEvents();
        void updatePacman();
        void updateGhost();
        void updateText();
        void update();

        void renderMap(sf::RenderTarget& target);
        void renderPacman(sf::RenderTarget& target);
        void renderGhost(sf::RenderTarget& target, int ghost);
        void renderText(sf::RenderTarget& target);
        void renderGameOver(sf::RenderTarget& target);
        void render();

        double distance(Pacman&, Ghost&);
        std::array<double, 4> verifyDistances();
};

#endif
```

CONCLUSÃO

A Programação Orientada permite o desenvolvimento de softwares complexos por meio de trabalho colaborativo.



OBRIGADA!