Курс С++ && промышленное программирование

Mail.Ru & МФТИ, 2017

SFML

- Simple and Fast Multimedia Library
- http://sfml-dev.org

- Обертка над OpenGL, OpenAL, FLAC, ogg, vorbis, freetype, jpeg и т.д.

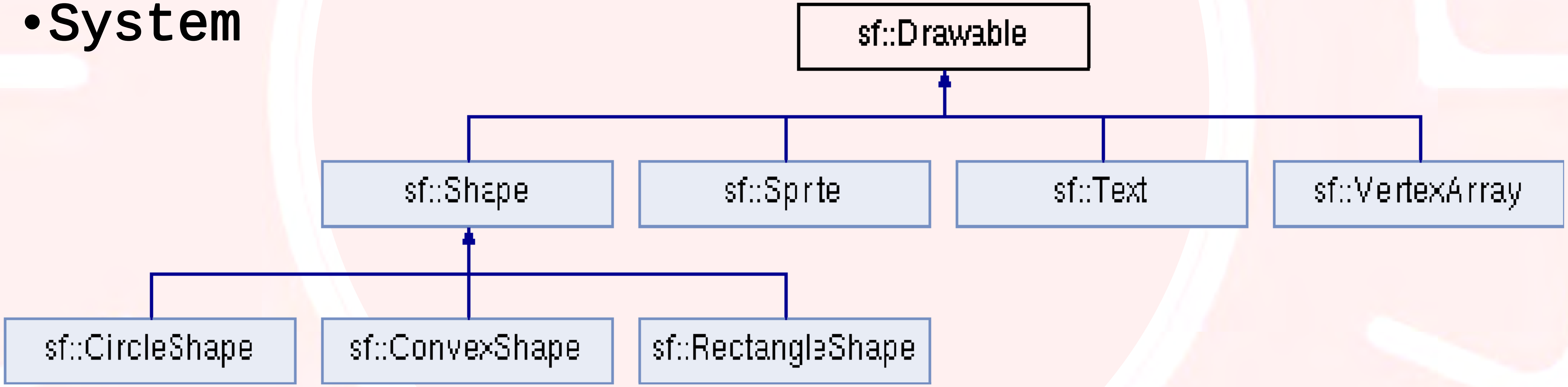
- Кроссплатформенная, многоязыковая

- Документация и туториалы - на английском

Модули

- Window
- Графика
- Аудио
- Сеть
- System

```
sfml-audio-2.dll          sfml-audio-d-2.dll
sfml-graphics-2.dll       sfml-graphics-d-2.dll
sfml-network-2.dll        sfml-network-d-2.dll
sfml-system-2.dll         sfml-system-d-2.dll
sfml-window-2.dll         sfml-window-d-2.dll
openal32.dll              openal32.dll
```



```
g++ КОТИКИ.cpp -lsfml-audio-d -lsfml-graphics-d -lsfml-main-d -lsfml-network-d
               -lsfml-system-d -lsfml-window-d
               –lfreetype –ljpeg -lopengl32 -lgdi32 –lwinmm
               -lopenal32 –logg –lvorbis –lvorbisenc -lvorbisfile –lFLAC
```

Окно

```
#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>

// ...

sf::RenderWindow window (sf::VideoMode (800, 600), “Title”);

while (window.isOpen())
    {
    sf::Event event;
    while (window.pollEvent (event))
        if (event.type == sf::Event::Closed)
            window.close();

    if (sf::Keyboard::isKeyPressed (sf::Keyboard::Escape))
        break;

    // ...
    }
```

Текстуры и спрайты

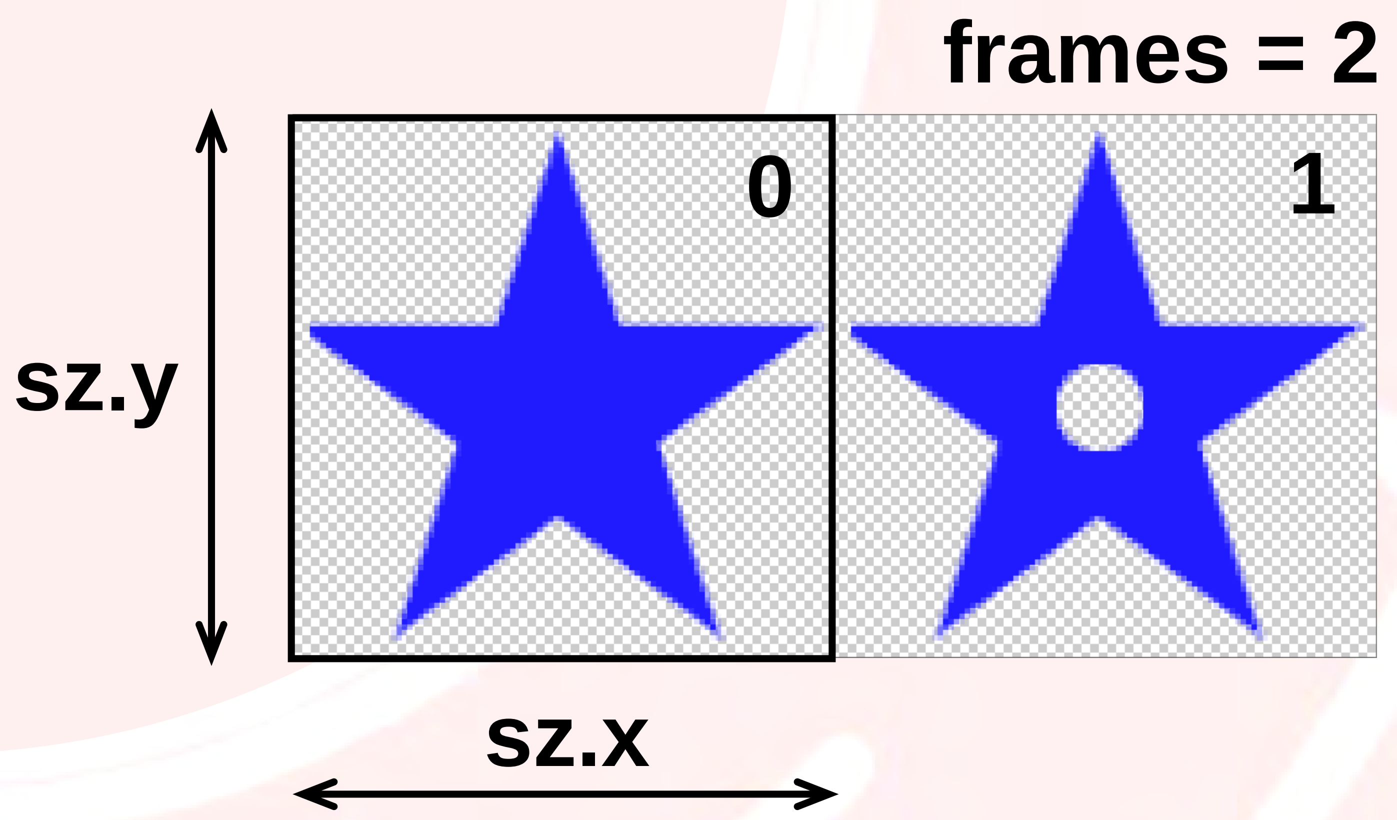
```
sf::Texture texture;
texture.loadFromFile (“Texture.png") || RETURN (EXIT_FAILURE);

sf::Sprite sprite (texture);
sprite.setPosition (x, y);

while (window.isOpen())
    {
    // ...

    window.clear();
    window.draw (backgroundSprite);

    window.draw (sprite);
    window.draw (otherSprite);
    // ...

    window.display();
    }
```

Шрифты и музыка

```
sf::Font font;
font.loadFromFile (“Font.ttf")   || RETURN (EXIT_FAILURE);

sf::Text text ("std::meow", font, 50);

sf::Music music;
music.openFromFile (“Music.ogg") || RETURN (EXIT_FAILURE);
music.setLoop (true);

music.play();

while (window.isOpen())
    {
    window.clear();
    // ...

    window.draw (text);
    // ...

    window.display();
    }
```

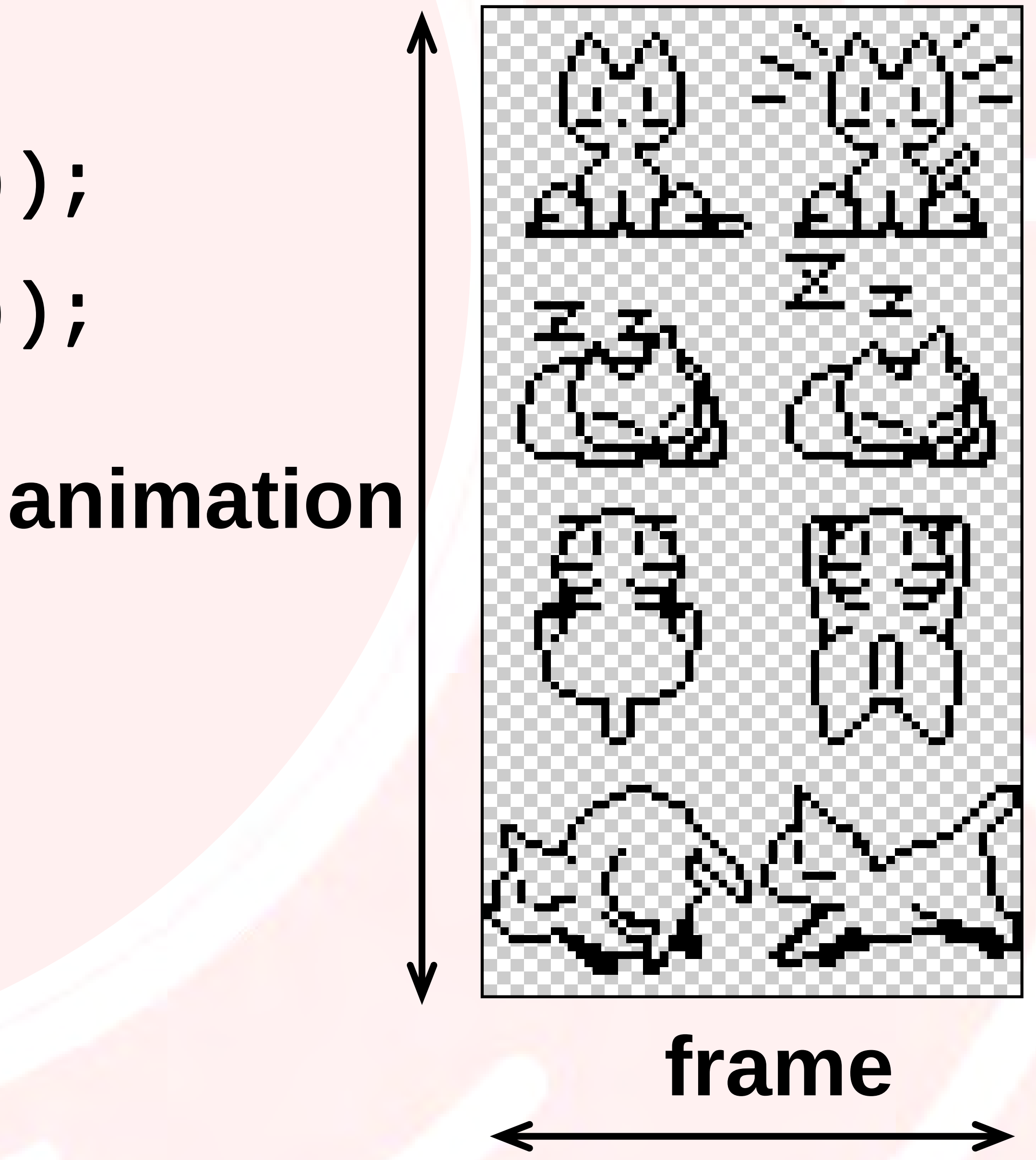
Анимация

```
while (...)
  {
  sprite.setTextureRect (sf::IntRect (t % frames      * sz.x, 0,
                                      (t % frames + 1) * sz.x, sz.y));

  sprite.setOrigin (sz.x/2, sz.y/2);
  sprite.rotate    (1.0);
  sprite.setScale  (sf::Vector2f (0.5, 2));

  window.clear (sf::Color (128, 128, 128));
  window.draw  (background);
  // ...

  window.draw (sprite);
  // ...

  window.display();

  t++;
  }
```



Анимация

```
while (...)
  {
  sprite.setTextureRect (sf::IntRect (t % frames      * sz.x, 0,
                                      (t % frames + 1) * sz.x, sz.y));

  sprite.setOrigin (sz.x/2, sz.y/2);
  sprite.rotate    (1.0);
  sprite.setScale  (sf::Vector2f (0.5, 2));

  window.clear (sf::Color (128, 128, 128));
  window.draw  (background);
  // ...

  window.draw (sprite);
  // ...

  window.display();

  t++;
  }
```



Фигуры

```
void DrawCircle (Vec pos, float radius,
                sf::Color fillColor, sf::Color outlineColor)
    {
    sf::CircleShape circle;

    circle.setRadius       (radius);
    circle.setFillColor    (fillColor);
    circle.setOutlineColor (outlineColor);

    if (outlineColor != sf::Color::Transparent)
        circle.setOutlineThickness (3);

    circle.setOrigin   (radius, radius);
    circle.setPosition (pos.x,  pos.y);

    Window->draw (circle);
    }
```

Пример программы

```
#include <math.h>

#include <SFML/Graphics.hpp>
#include <SFML/Audio.hpp>

sf::RenderWindow* Window = nullptr;

template <typename T>
struct Vector
    {
    T x, y;

    Vector()                          : x (0),        y (0)        {}

    template <typename U1, typename U2>
    Vector (U1 x, U2 y)               : x ((T) x),    y ((T) y)    {}

    template <typename U>
    Vector (const Vector <U>& vec)    : x ((T) vec.x), y ((T) vec.y) {}

    template <typename U>
    Vector (const sf::Vector2 <U>& vec) : x ((T) vec.x), y ((T) vec.y) {}
    };

using Vec = Vector <float>;
```

Пример программы

ТЕХНОТРЕК

```
struct Hero
    {
    Vec pos_;
    Vec v_;
    Vec size_;

    float m_;
    float friction_;
    float rotation_;

    sf::Sprite sprite_;



    Hero() :
        pos_      (Vec()),
        v_        (Vec()),
        size_     (Vec()),
        m_        (0),
        friction_ (0),
        rotation_ (0),
        sprite_   (sf::Sprite())
        {
        auto bounds = sprite_.getLocalBounds();
        size_.x = bounds.width;
        size_.y = bounds.height;
        }
```

```
Hero (const Vec& pos, const Vec& v,
      float m = 0, float friction = 0, float rotation = 0,
      sf::Sprite sprite = sf::Sprite()) :
    pos_      (pos),
    v_        (v),
    size_     (Vec()),
    m_        (m),
    friction_ (friction),
    rotation_ (rotation),
    sprite_   (sprite)
    {
    auto bounds = sprite_.getLocalBounds();
    size_.x = bounds.width;
    size_.y = bounds.height;
    }

void draw (         sf::RenderTarget* screen = Window) const;
void draw (Vec pos, sf::RenderTarget* screen = Window) const;
bool doPhysics (Vec sz, float dt);
void control();
};
```

Пример программы

```
void RunGame()
    {
    auto sz = Window->getSize();

    sf::Texture deathTex;
    deathTex.loadFromFile ("death.png") || RETURN();
    deathTex.setSmooth (true);

    sf::Texture vaderTex;
    vaderTex.loadFromFile ("vader.png") || RETURN();
    vaderTex.setSmooth (true);

    sf::Music music;
    music.openFromFile    ("vader.wav") || RETURN();
    music.setLoop (true);
    music.play();

    Hero death (Vec (    70,       70), Vec (300, 450), 0.033f, 0.0f, 5, sf::Sprite (deathTex));
    Hero vader (Vec (sz.x/2, sz.y*0.9), Vec (  0,    0), 0.003f, 0.2f, 0, sf::Sprite (vaderTex));
    int lives = 10;

    std::vector <Hero> stars (100);
    GenStars (stars.data(), stars.size(), sz);
    ...
```

```
int main()
    {
    sf::RenderWindow window
        (sf::VideoMode (1100, 650), __FILE__,
         sf::Style::Default & ~sf::Style::Resize);

    window.setVerticalSyncEnabled (true);
    window.setFramerateLimit (25);

    Window = &window;

    RunGame();

    return 0;
    }
```

Пример программы

```
void RunGame()
    {
    ...
    sf::Clock clock;

    for (;;)
        {
        float time = clock.restart().asSeconds();

        sf::Event event;
        while (Window->pollEvent (event)) if (event.type == sf::Event::Closed) break;

        if (sf::Keyboard::isKeyPressed (sf::Keyboard::Escape)) break;

        if (!Window->isOpen()) break;

        DoRender (death, vader, lives, stars);

        if (death.pos_.y + death.size_.y/2.0 >= sz.y) { death.pos_ = death.size_; lives--; }
        if (lives <= 0) break;

        vader.control();

        DoIntersection (&death, &vader);

        death.doPhysics (sz, time);
        vader.doPhysics (sz, time);

        vader.pos_.x = (float) ( ((int) vader.pos_.x + sz.x) % sz.x );

        for (auto& star : stars)
            if (star.doPhysics (sz, time))
                GenStars (&star, 1, sz);
        }
    ...
```

Пример программы

```
void RunGame()
    {
    ...

    sf::RenderTexture tex;
    tex.create (sz.x, sz.y);
    tex.clear();

    sf::Sprite texSprite (tex.getTexture());
    texSprite.setOrigin  (0, (float) sz.y);
    texSprite.setScale   (1, -1);

    for (size_t n = 0; n < 200; n++)
        {
        sf::Event event;
        while (Window->pollEvent (event)) if (event.type == sf::Event::Closed) break;

        if (sf::Keyboard::isKeyPressed (sf::Keyboard::Escape)) break;

        for (int i = 0; i < 10; i++) vader.draw (Vec (rand() % sz.x, rand() % sz.y), &tex);

        Window->clear();
        Window->draw (texSprite);
        Window->display();
        }

    Window->close();
    }
```

Пример программы

```
void DoRender (const Hero& death, const Hero& vader, int lives, const std::vector <Hero>& stars)
    {
    Window->clear();

    for (const auto& star : stars) star.draw();

    death.draw();
    vader.draw();

    for (int i = 0; i < lives; i++) vader.draw (Vec ((i + 1.0) * vader.size_.x, vader.size_.y/2));

    Window->display();
    }
```

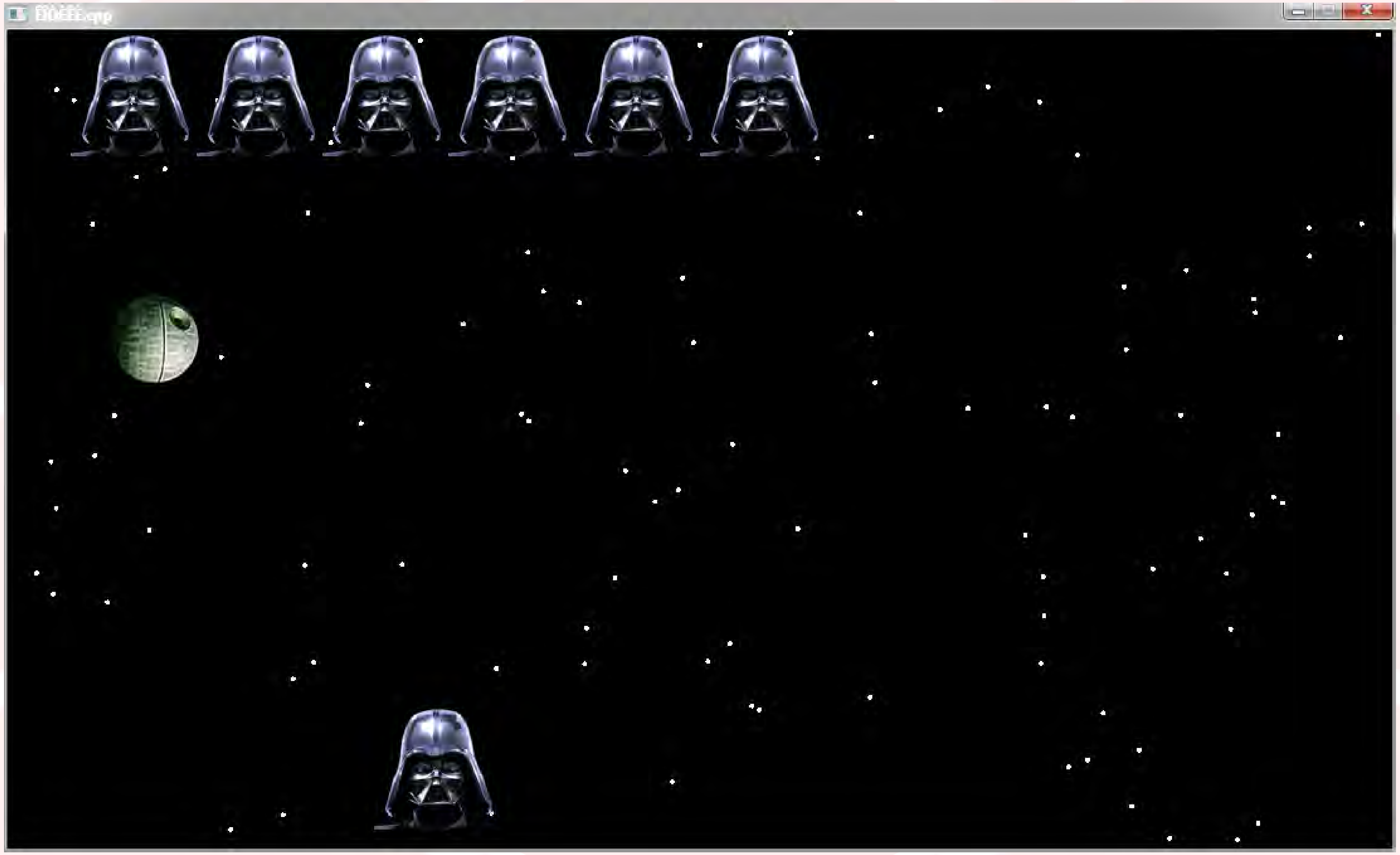
```
void Hero::draw (sf::RenderTarget* screen) const
    {
    if (sprite_.getTexture())
        screen->draw (sprite_);

    else
        DrawCircle (pos_, 2,
                    sf::Color::White,
                    sf::Color::Transparent,
                    screen);
    }
```

```
void Hero::draw (Vec pos,
                 sf::RenderTarget* screen) const
    {
    if (sprite_.getTexture())
        {
        sf::Sprite sprite = sprite_;

        sprite.setOrigin   (size_.x/2, size_.y/2);
        sprite.setPosition (pos.x, pos.y);
        screen->draw (sprite);
        }

    else
        DrawCircle (pos_, 2,
                    sf::Color::White,
                    sf::Color::Transparent,
                    screen);
    }
```

Пример программы

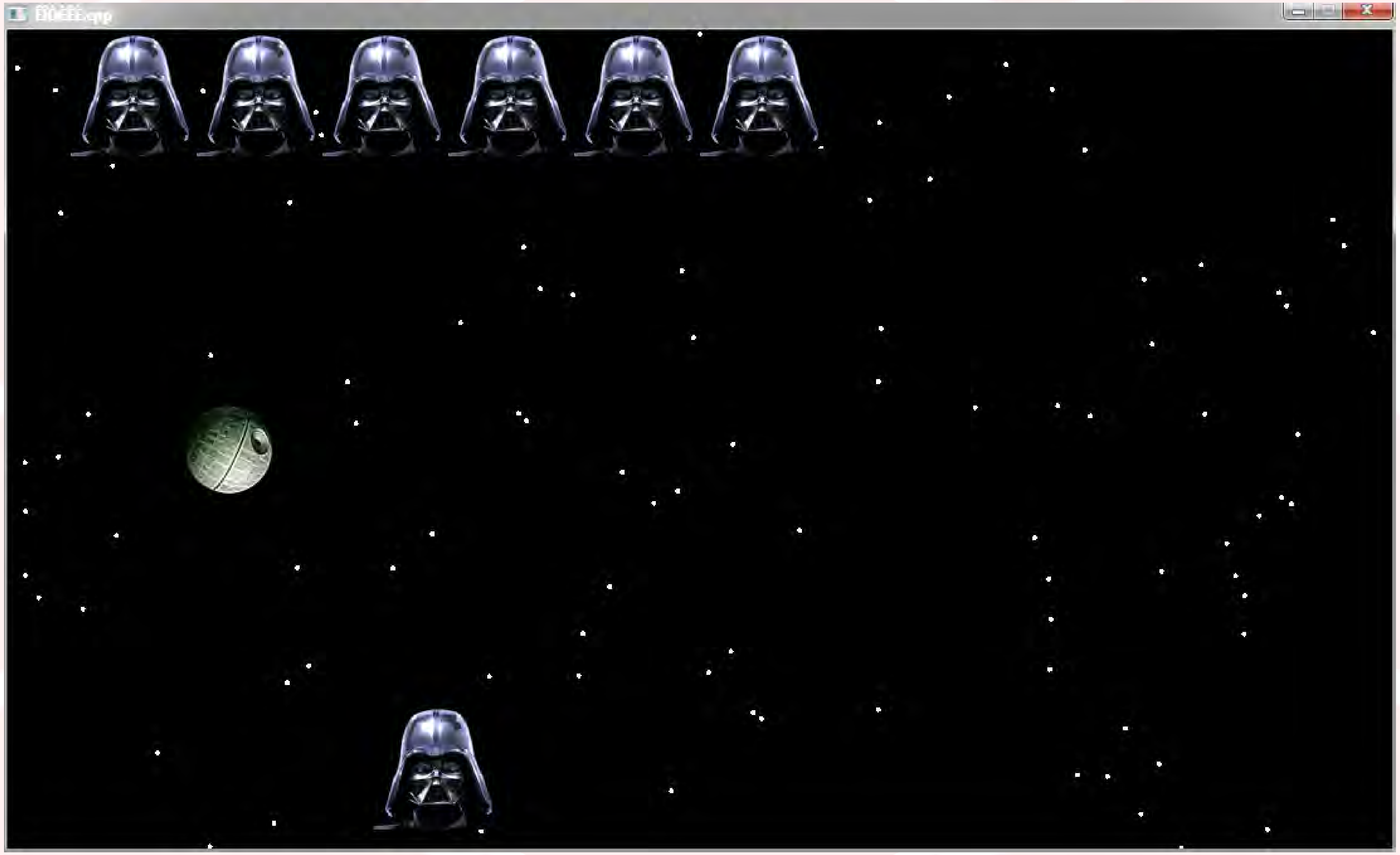
```
bool Hero::doPhysics (Vec sz, float dt)
    {
    // ...

    pos_.x += v_.x * dt;
    pos_.y += v_.y * dt;

    sprite_.setOrigin   (size_.x/2, size_.y/2);
    sprite_.setPosition (pos_.x, pos_.y);
    sprite_.rotate      (rotation_);

    // ...
    }

void Hero::control()
    {
    if (sf::Keyboard::isKeyPressed (sf::Keyboard::Left))  v_.x -= 1 / m_;
    if (sf::Keyboard::isKeyPressed (sf::Keyboard::Right)) v_.x += 1 / m_;
    }
```

Пример программы



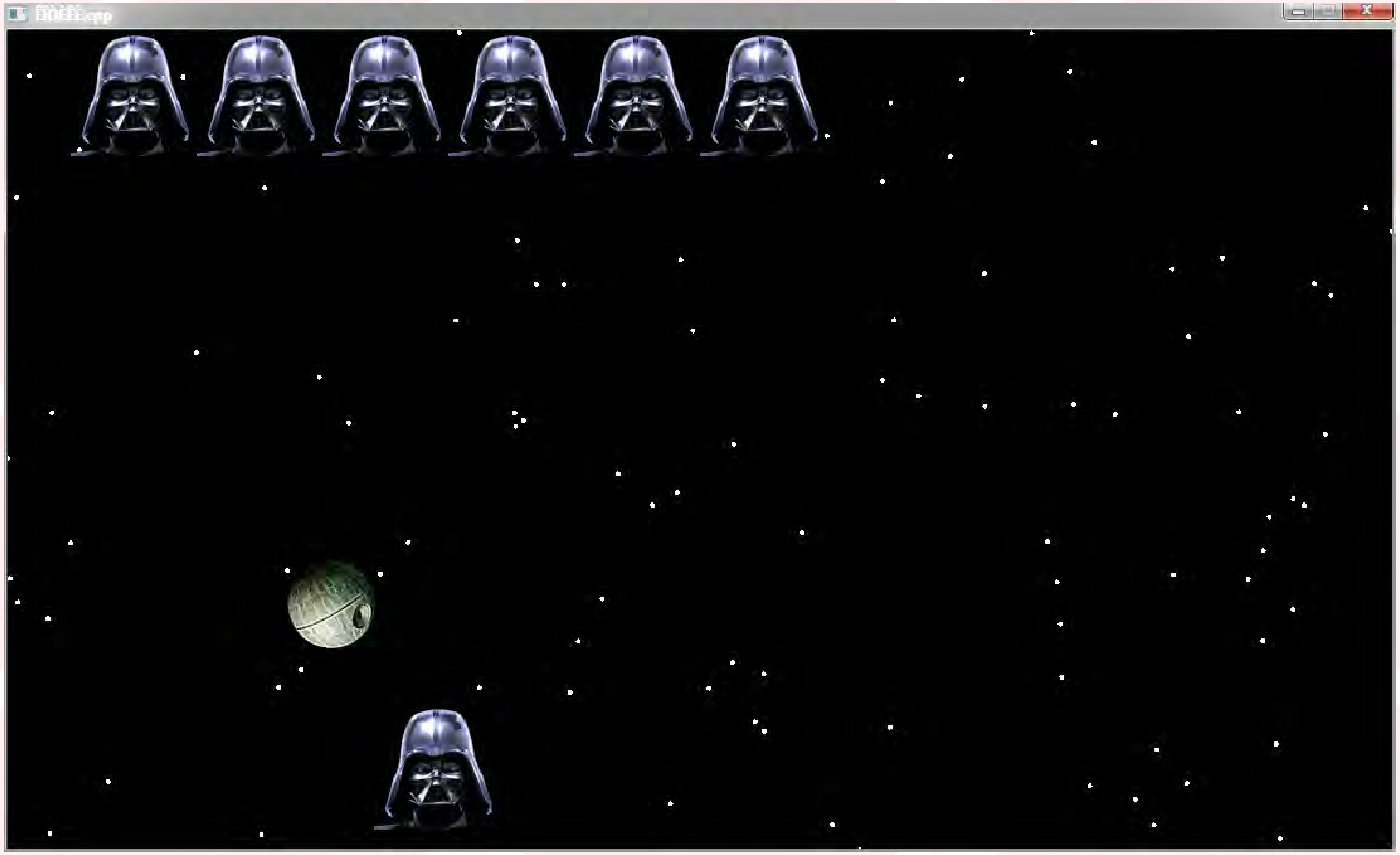
Пример программы

ТЕХНОТРЕК



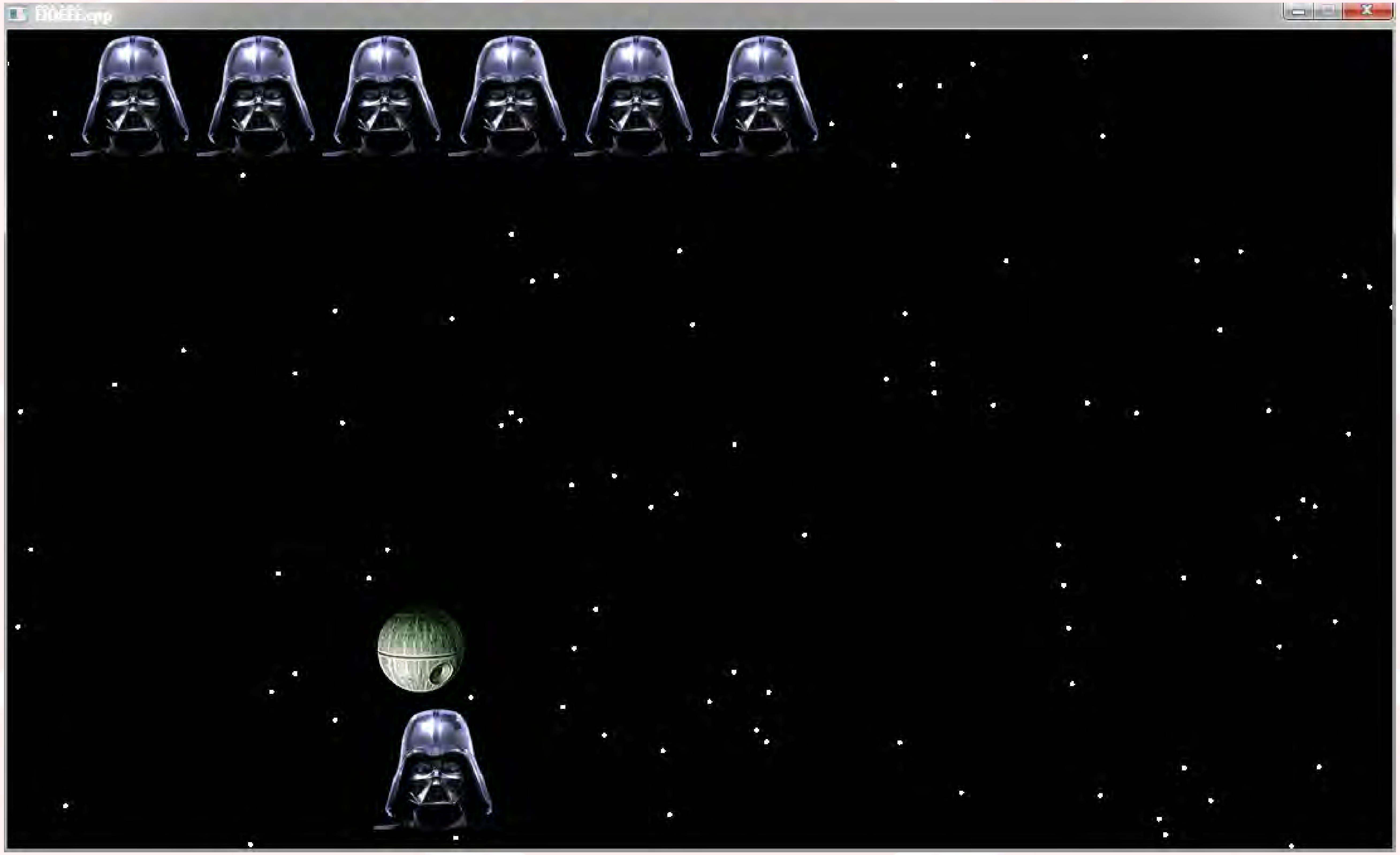
Пример программы

ТЕХНОТРЕК



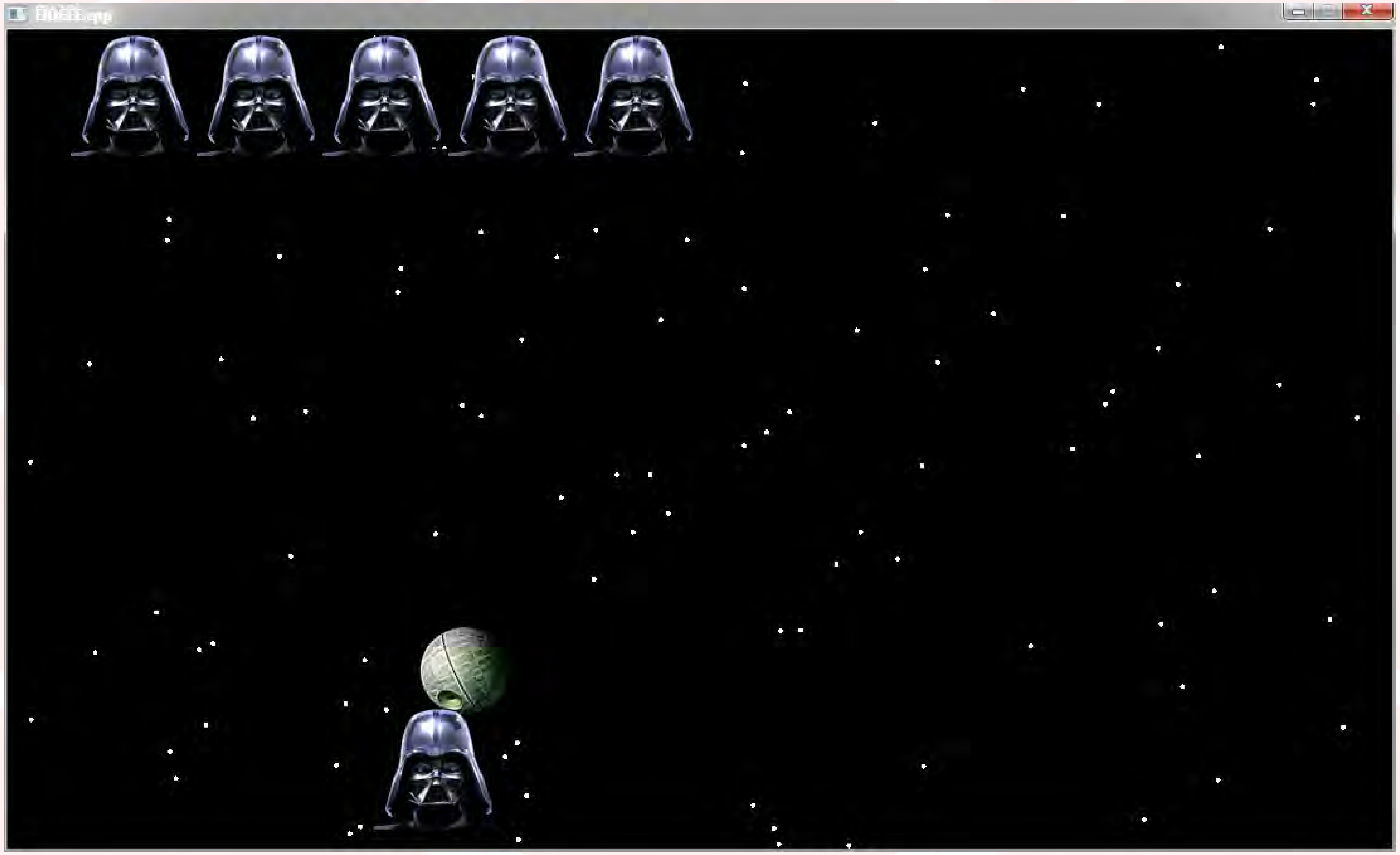
Пример программы

ТЕХНОТРЕК



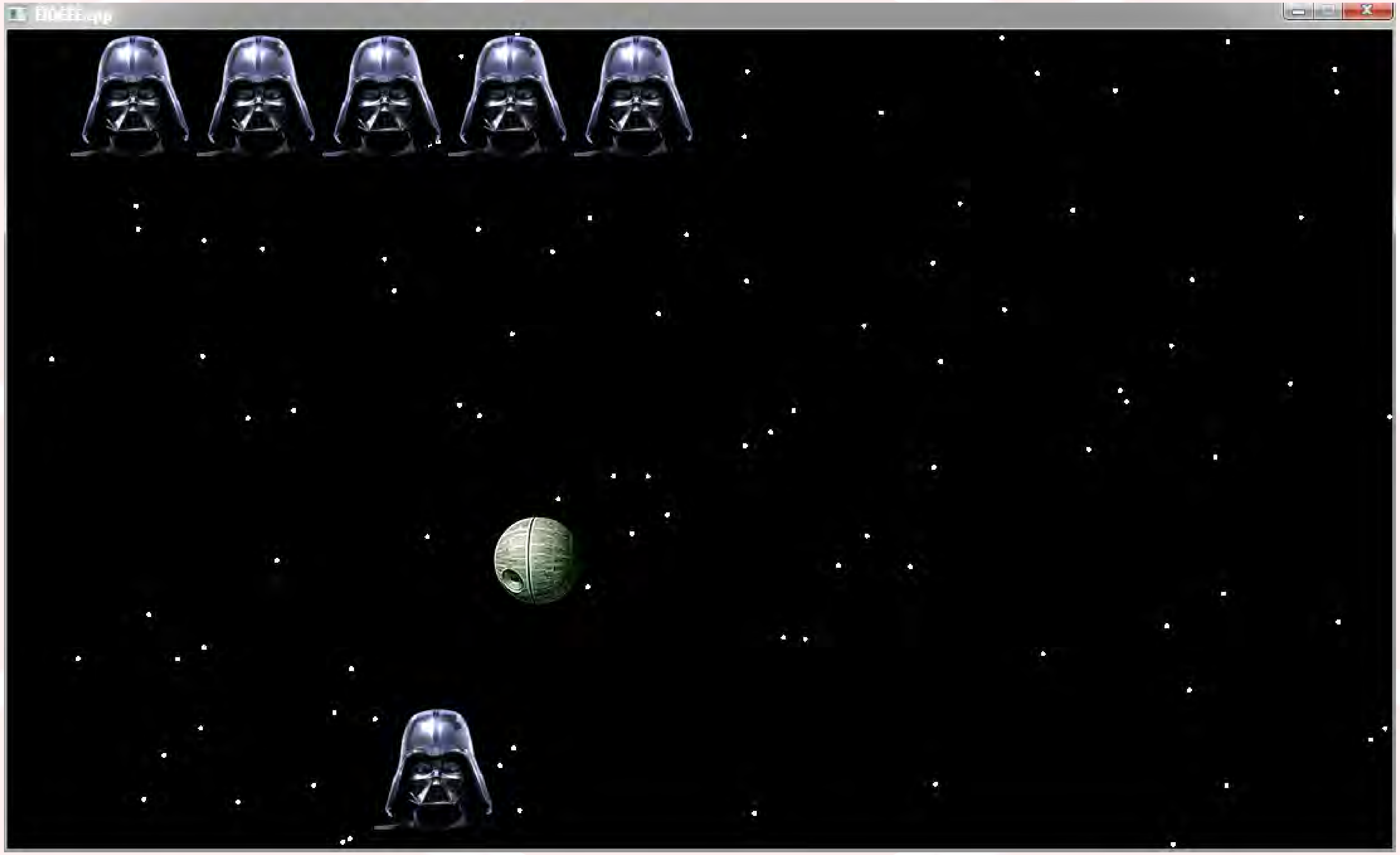
Пример программы

ТЕХНОТРЕК



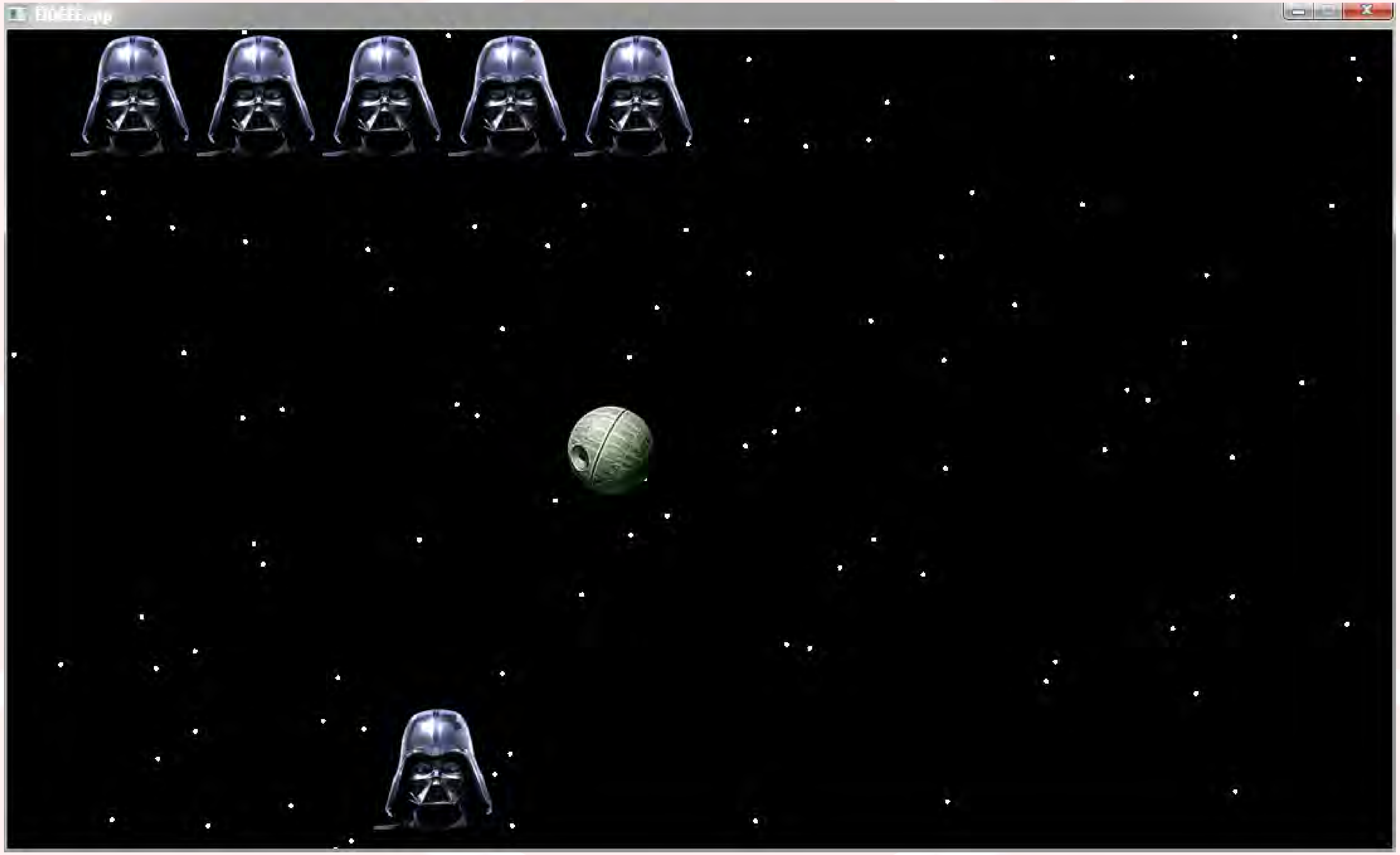
Пример программы

ТЕХНОТРЕК



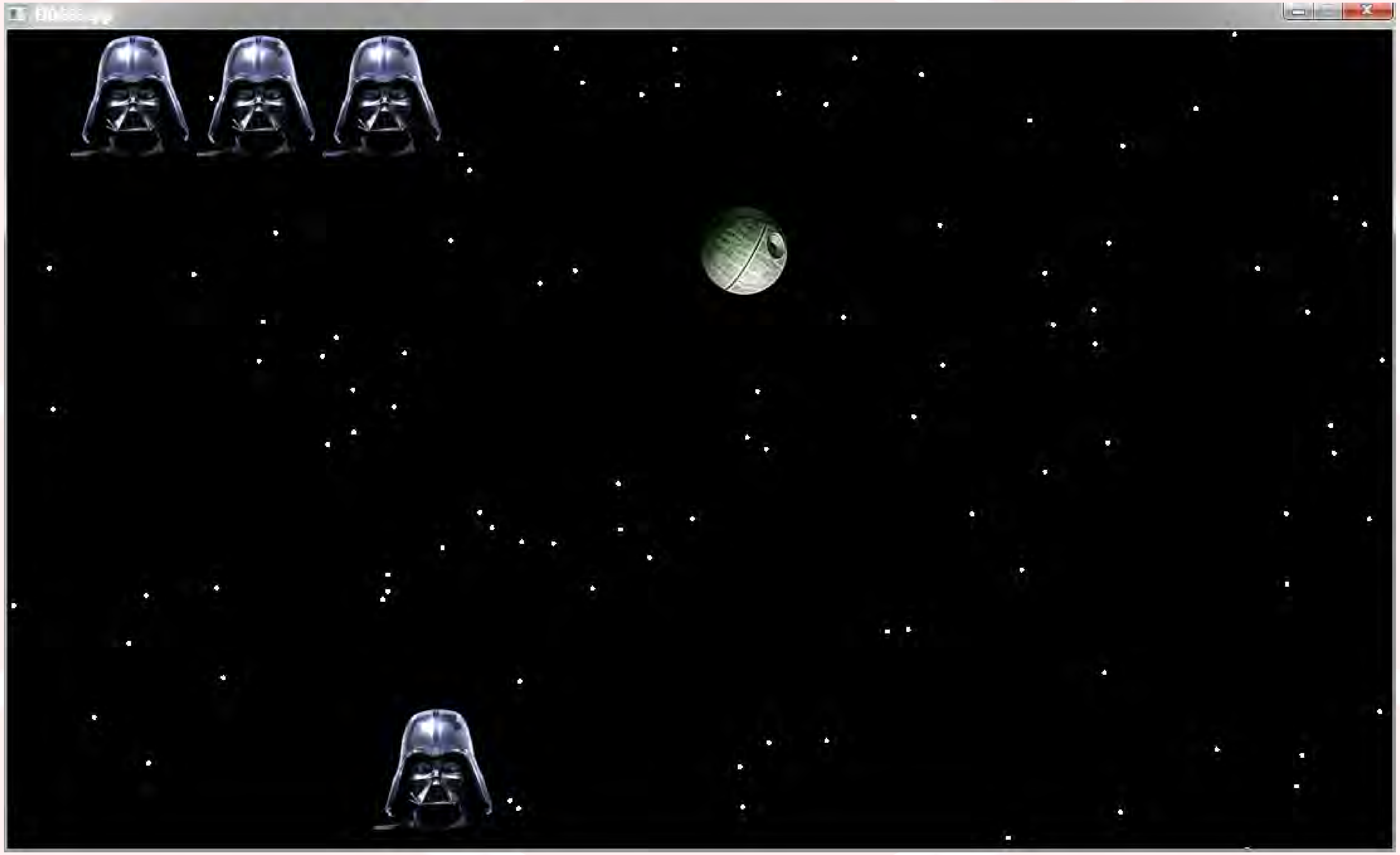
Пример программы

ТЕХНОТРЕК

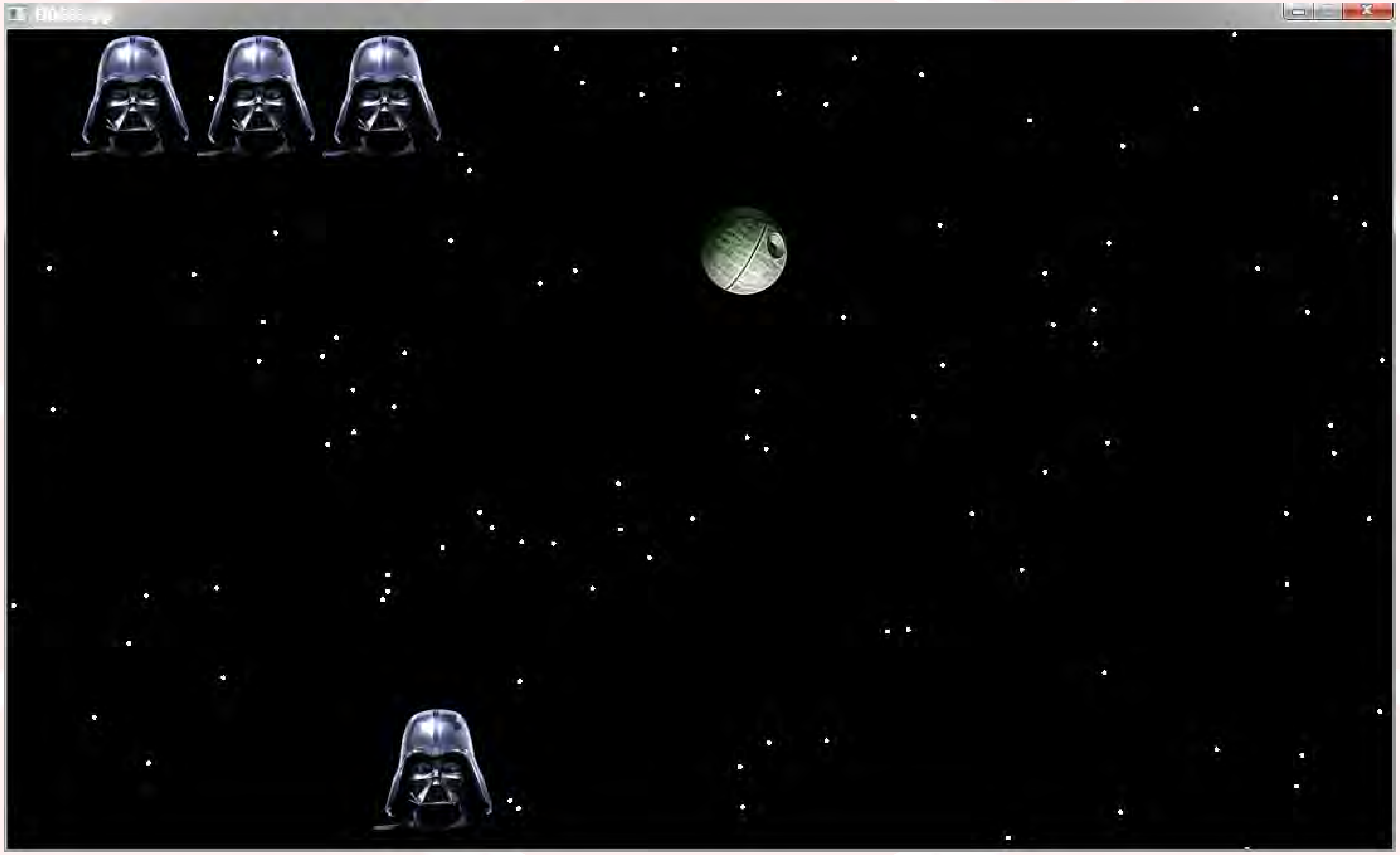


Пример программы

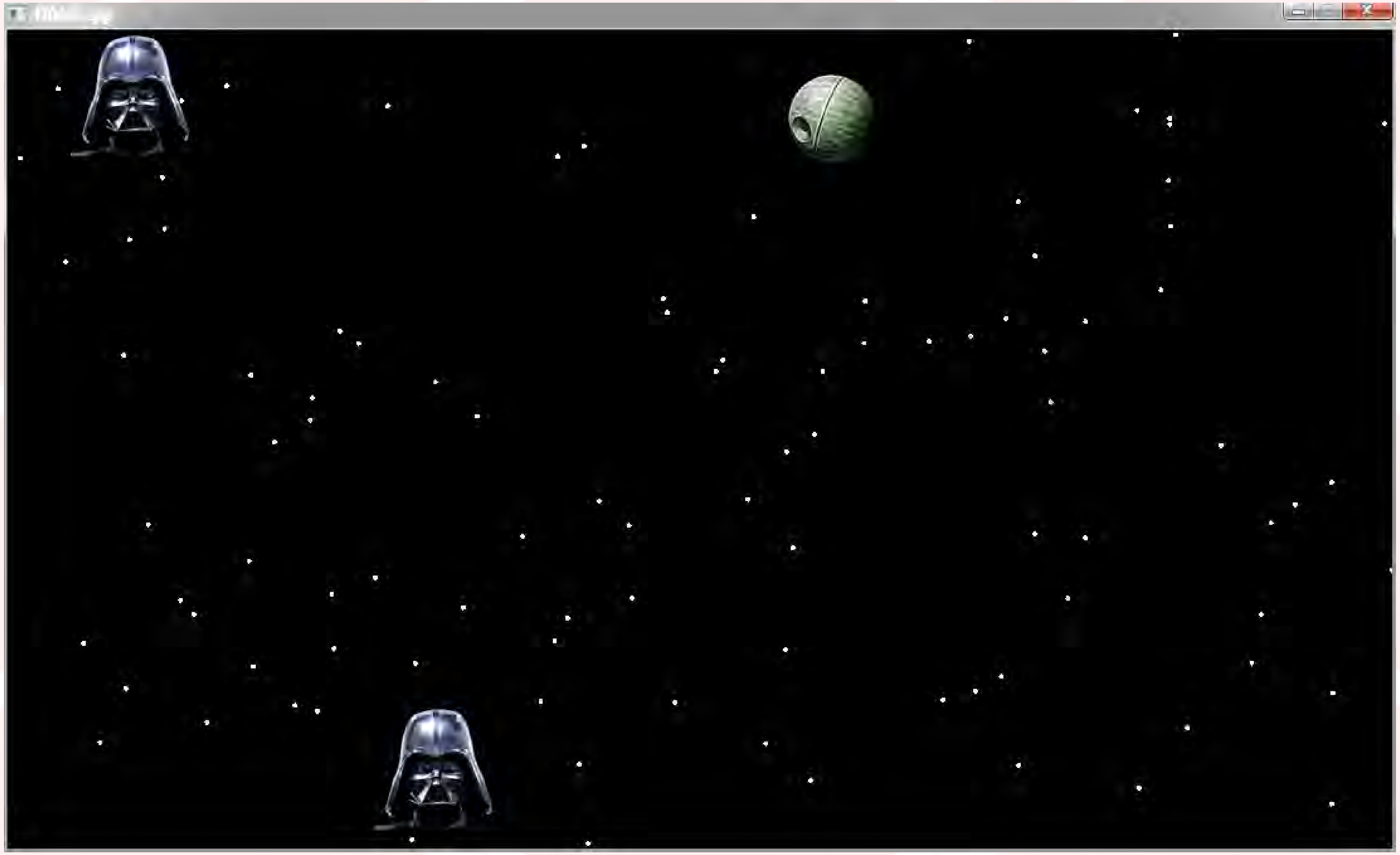
ТЕХНОТРЕК



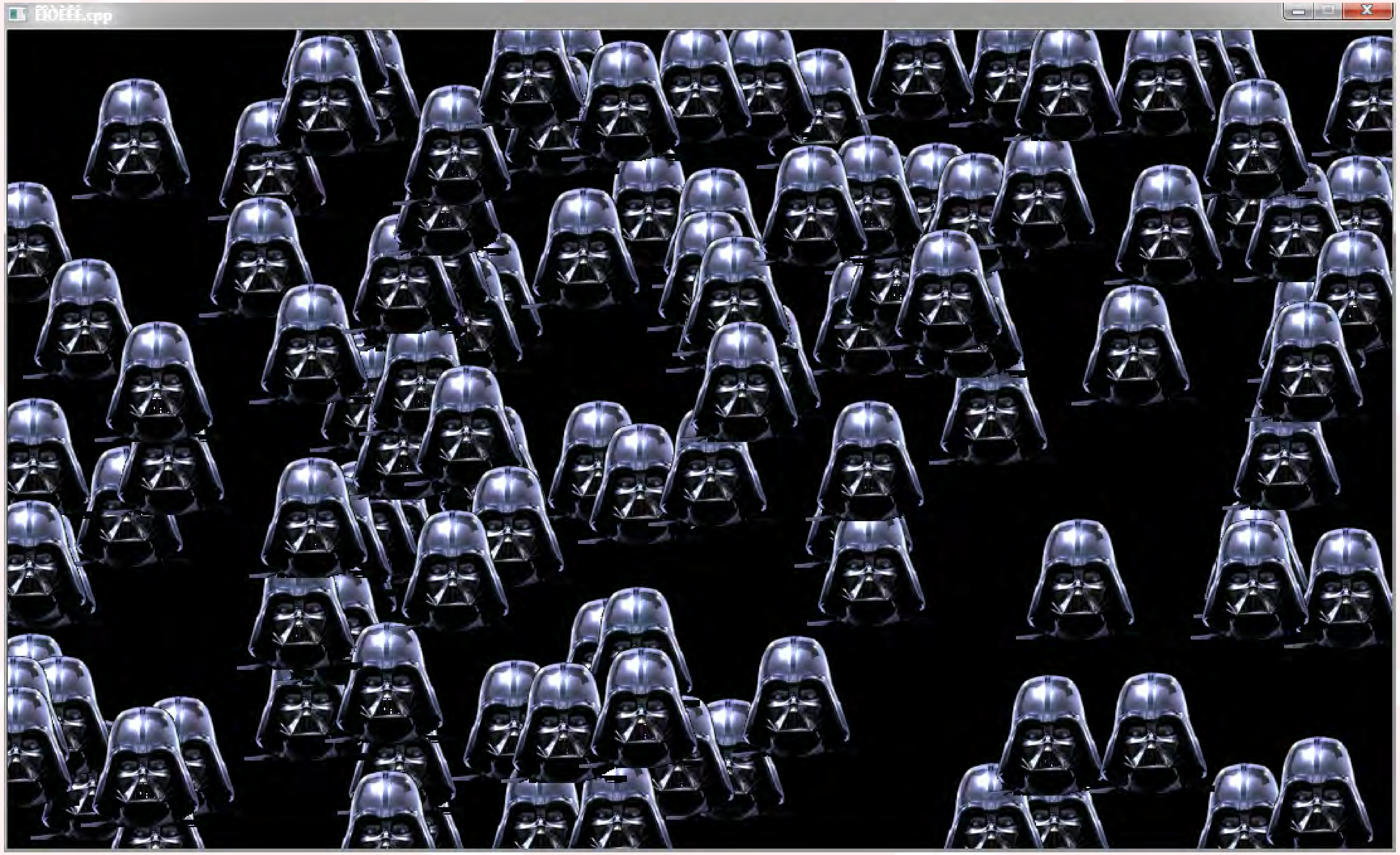
Пример программы



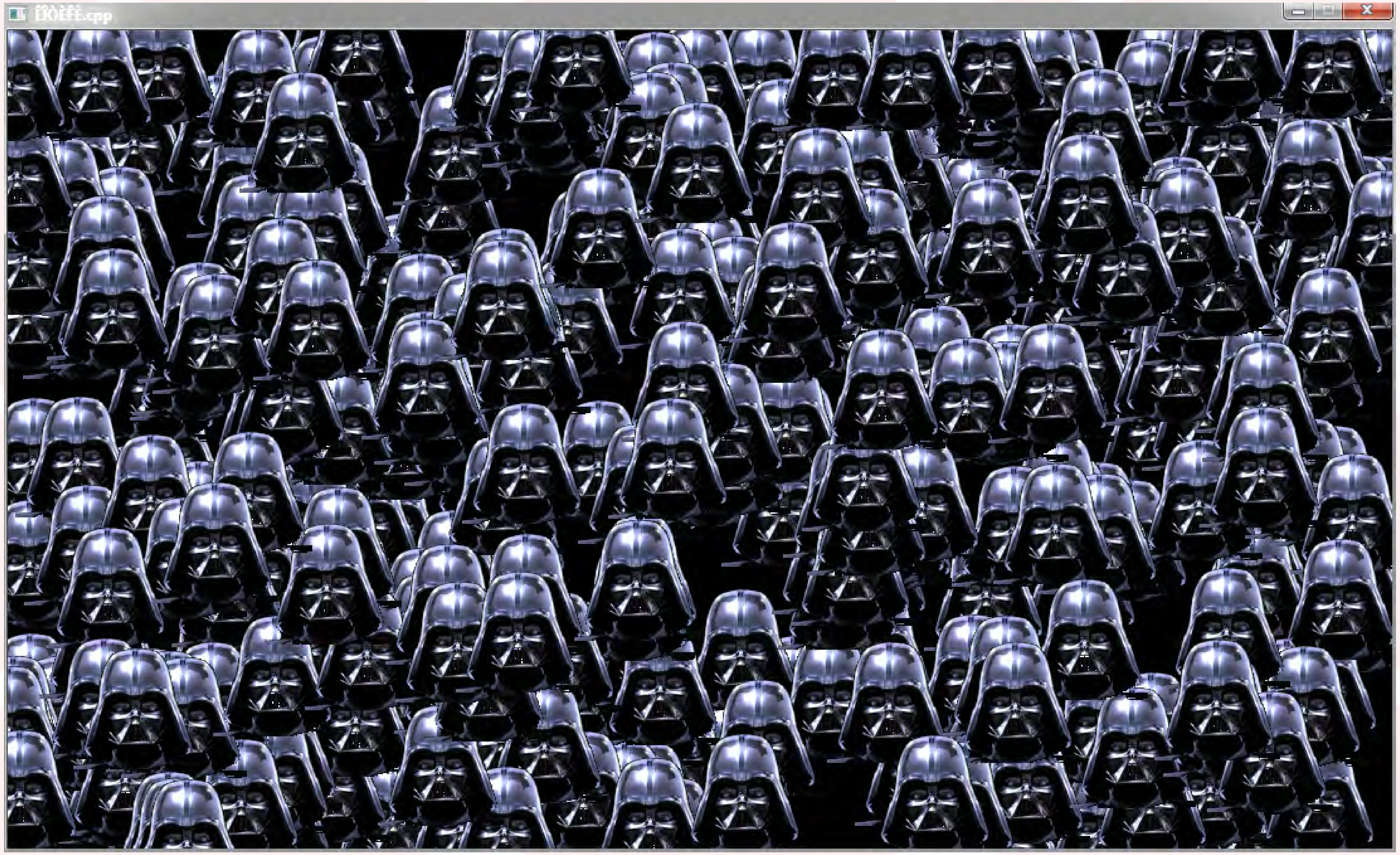
Пример программы



Пример программы

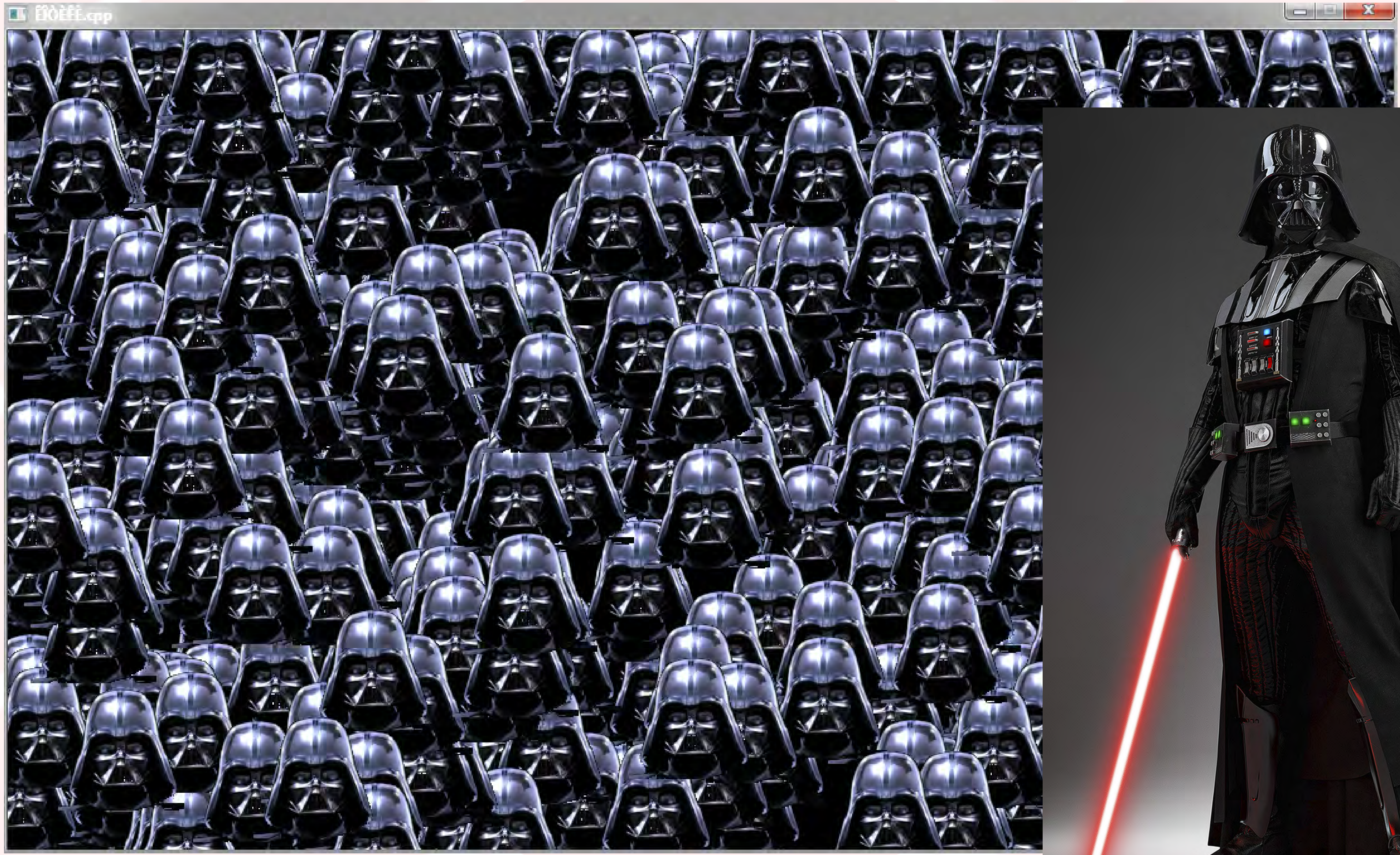


Пример программы



Пример программы

ТЕХНОТРЕК



```
#pragma beware
if (player.failed()) theWorld.~World();
```

// Оставьте фидбек по лекции!

Спасибо за внимание

&&

Задавайте вопросы!

Mail.Ru & МИФИ, 2017