

Разработка приложения «Игра Змейка»

Отчет о проектной работе по курсу «Основы информатики и программирования»

Кулаков А.М.

11 июня 2021

Цель работы

Игра должна соответствовать по всем канонам "Змейки" и должна завлечь на пару минут.

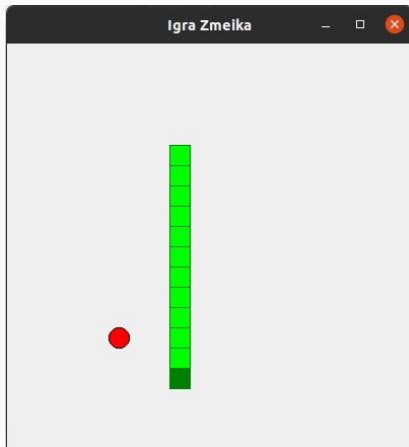


Рис. 1: Snakee!

В этом приложении будет 2 модуля

- main.cpp
- game.cpp

Задачи

- разработать функцию отрисовки змейки и яблока;
- разработать функцию врезаний в препятствия;
- разработать функцию которая предотвращает разворот 180;
- разработать функцию рандомного спавна яблок;
- разработать функцию возможности поедания яблок;
- разработать функцию движения змейки;

Функция 1

```
void Game::doDrawing()
{
    QPainter qp(this);

    if (m_inGame){

        qp.setBrush(Qt::red);
        qp.setPen(Qt::black);
        qp.drawEllipse(m_apple.x() * DOT_WIDTH, m_apple.y() * DOT_HEIGHT, DOT_WIDTH, DOT_HEIGHT);

        for(int i = 0; i < m_dots.size(); ++i)
            if (i == 0) {
                qp.setBrush(Qt::darkGreen);
                qp.setPen(Qt::darkGreen);
                qp.drawRect(m_dots[i].x() * DOT_WIDTH, m_dots[i].y() * DOT_HEIGHT, DOT_WIDTH, DOT_HEIGHT);
            } else {
                qp.setBrush(Qt::green);
                qp.setPen(Qt::darkGreen);
                qp.drawRect(m_dots[i].x() * DOT_WIDTH, m_dots[i].y() * DOT_HEIGHT, DOT_WIDTH, DOT_HEIGHT);
            }
            } else {
                gameOver();
            }
    }
}
```

Рис. 2: отрисовка змейки и яблока

Функция 2

```
void Game::check_field()
{
    if (m_dots.size() > 4){
        for (int i = 1; i < m_dots.size(); ++i) {
            if (m_dots[0] == m_dots[i]) {
                m_inGame = false;
            }
        }
    }
    if(m_dots[0].x() >= FIELD_WIDTH) { m_inGame = false; }
    if(m_dots[0].x() < 0) { m_inGame = false; }
    if(m_dots[0].y() >= FIELD_HEIGHT){ m_inGame = false; }
    if(m_dots[0].x() < 0) { m_inGame = false; }

    if(!m_inGame) {
        killTimer(timeId);
    }
}
```

Рис. 3: врезание в препятствия

```
void Game::keyPressEvent(QKeyEvent *event)
{
    int key = event->key();
    if (key == Qt::Key_Left && m_dir != Directions::right) { m_dir = Directions::left; }
    if (key == Qt::Key_Right && m_dir != Directions::left) { m_dir = Directions::right; }
    if (key == Qt::Key_Up && m_dir != Directions::down) { m_dir = Directions::up; }
    if (key == Qt::Key_Down && m_dir != Directions::up) { m_dir = Directions::down; }
}
```

Рис. 4: предотвращение на 180

Функция 4,5

```
void Game::localApple()
{
    QTime time = QTime::currentTime();
    qsrand((uint) time.msec());
    m_apple.rx() = qrand() % DOT_WIDTH;
    m_apple.ry() = qrand() % DOT_HEIGHT;
}

void Game::check_apple()
{
    if (m_apple == m_dots[0]) {
        m_dots.push_back(QPoint(0,0));
        localApple();
    }
}
```

Рис. 5: случайное яблоко и поедание

Функция 6

```
void Game::move()
{
    for(int i = m_dots.size() - 1; i > 0; --i) {
        m_dots[i] = m_dots[i-1];
    }

    switch (m_dir){
    case left:  {m_dots[0].rx() -= 1; break;}
    case right: {m_dots[0].rx() += 1; break;}
    case up:    {m_dots[0].ry() -= 1; break;}
    case down:  {m_dots[0].ry() += 1; break;}
    }
}
```

Рис. 6: движение

Спасибо за внимание !