

Разработка приложения «Alien attack»

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

Костин Данила

9 июня 2021

Цель проекта - реализовать игру «Alien attack» - используя языки C++ и Qt.

Задачи проекта:

1. Разработать графический интерфейс приложения
2. Реализовать функции, случайным образом генерирующие координаты на экраны, поворот объекта в градусах, нормирующую и вычисляющую расстояние между точками.

Все модули были поделены на две группы:

1. Реализация на Qml

- 1.1 main - основной модуль окна
- 1.2 Plane - модуль самолёта и его движения
- 1.3 Enemy - модуль НЛО и взаимодействия с ним
- 1.4 GameFuncs - модуль функций меню и начала/конца игры
- 1.5 Rocket - модуль ракеты

2. Реализация на C++

- 2.1 helper - основные функции поворота, нормирования и других вычислений

Описание игрового процесса



Рис. 1: Игровое окно

Пользователь управляет коричневым самолётом, его задача не врезаться в НЛО и сбить его путём пуска ракеты. Коротко перечислим функционал, доступный пользователю:

1. Самолёт летит в место клика
2. При нажатии клавиши «Space» самолёт выпускает ракету
3. В меню имеются кнопки начала новой игры или выхода
4. В меню выводится количество сбитых НЛО и секунды до перезарядки ракеты

Данный модуль - основа всех модулей на qml. Он задаёт размеры окна, его фон, обрабатывает нажатие на область окна. При нажатии происходит вызов функции `vector()` на языке C++, которая нормирует направляющий вектор в место клика мыши, чтобы самолёт полетел к месту клика без ускорения, а также функция `atang()` поворота самолёта в направлении клика.

В данном модуле происходит прибавление к координатам самолёта координат некоторого вектора, вычисленного после клика мыши. Также здесь присутствует проверка того, чтобы самолёт не улетел за предел видимой части окна.

Данный модуль содержит функции перемещения НЛО к некоторой точке в центре экрана по таймеру. Другая функция - это присваивание случайных координат НЛО, если он был сбит или улетел за пределы окна. Также в данном модуле происходит проверка того, что НЛО столкнулось с выпущенной ракетой. Если столкновение произошло, то ракета убирается из видимой части окна, счёт увеличивается, и происходит генерация нового положения НЛО функцией `rand()`.

Модуль основных функций меню снизу окна, а также отображения счёта, обработки перезарядки, вывода оставшихся секунд, а также содержит функцию открытия диалога с концом игры (при столкновении самолёта с НЛО).

Задаёт кнопки новой игры, при нажатии которой самолёт смещается в начальную точку расположения на взлётной полосе, и кнопку выхода.

При выстреле ракеты задаётся флаг (`ready_strike`) того, что начата перезарядка, и пока количество секунд не дойдет до 0, функция `check` каждую секунду будет уменьшать время перезарядки.

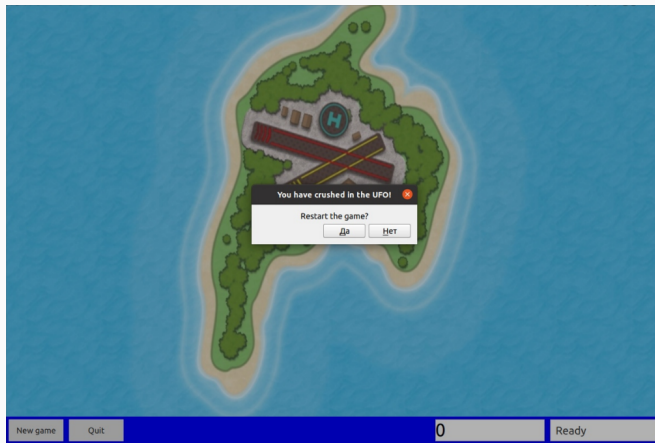


Рис. 2: Диалог о проигрыше

Далее будут перечислены все функции модуля `helper`:

1. `rast`
2. `rand`
3. `atang`
4. `reject`
5. `vector`

Данная функция находит расстояние между двумя точками по следующей формуле:

$$length = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2};$$

Она возвращает `true`, если расстояние между точками меньше 70 пикселей (это расстояние было выбрано экспериментально, оно удовлетворяет размерам объектов)

Данная функция случайным образом генерирует число, получает от него остаток от деления на 4. Полученное число - есть сторона, с которой прилетит НЛО. Так, если это число будет 0, то НЛО прилетит сверху, 1 - справа, и так далее по часовой стрелке.

Данная функция получает координаты вектора, направляющего полёт самолёта, получает из них тангенс, и через формулу арктангенса выражает угол поворота самолёта в градусах и возвращает его:

$$rotation = \frac{\arctg \operatorname{tg} \frac{y}{x} * 180}{\pi};$$

Данная функция получает координаты вектора и нормирует их - то есть делит каждую координату на длину вектора:

$$norma_x = \frac{x}{\sqrt{x^2 + y^2}}; norma_y = \frac{y}{\sqrt{x^2 + y^2}};$$

Данная функция получает поворот самолёта в градусах, и чтобы найти угол, который нужно будет присвоить самолёту при его столкновении с границей окна, из 180 вычитается текущий поворот. Таким образом будет найден угол - разность 180 и угла падения, он же совпадает с углом отражения.

В заключение можно сказать, что была создано функционирующее приложение с графическим интерфейсом. Данное приложение получилось достаточно объемным в связи с большим количеством взаимодействий объектов таких, как ракета и НЛО, самолёт и НЛО, самолёт и граница окна и так далее. В проекте использовался как язык qml, так и C++, без которого не возможна была бы реализация большинства функций приложения.

Спасибо за внимание !