Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

”Петрозаводский Государственный Университет”

Физико-технический институт

Отчёт по командному заданию по курсу

«Технология программирования»

Выполнили:

студенты группы 21312

Физико-технического института

Кунильский Н. В.

Белюсенко В. Н.

Тароев Д. С.

Преподаватель:

Бульба Артем Владимирович

Петрозаводск

2017

Цель работы:

Разработать программу (игру) с использованием наследования, контейнеров, итераторов и раздельной компиляции. Промежуточные результаты работы необходимо сохранять на GitHub (адрес репозитория: <http://github.com/NikitaKunilsky/GameProject> ).

Используемые средства программной реализации:

Среда разработки программы: Microsoft Visual Studio 2015 Community

Язык программирования: С++ с подключенной библиотекой SFML

Созданные файл:

game.cpp - главный код, которые обращается ко всем заголовочным файлам. Описание всего функционала игры;

Entity.h ( +Entity.cpp) - заголовочный файл с объявлением класса родитель для классов пуля, игрок и противник и файл с определением класса родитель;

bullet.h ( +bullet.cpp) - заголовочный файл с объявлением класса пуля и файл с определением класса пуля;

player.h ( +player.cpp) - заголовочный файл с объявлением класса игрок и файл с определением класса игрок;

Enemy.h ( +Enemy.cpp) - заголовочный файл с объявлением класса противник и файл с определением класса противник;

Global.h - заголовочный файл с объявлением массива строк и функции активации ловушек;

map.h - заголовочный файл с объявлением и инициализацией переменных для ширины и высоты карты;

map.cpp - файл с инициализацией массива строк и определением функции активации ловушек;

Game.obj - объектный код исходного текста игры;

Entity.obj - объектный код модуля исходного кода класса родитель;

bullet.obj - объектный код модуля исходного кода класса пуля;

player.obj - объектный код модуля исходного кода класса игрок;

Enemy.obj - объектный код модуля исходного кода класса противник;

map.obj - объектный код модуля исходного кода с массивом строк и функцией активации ловушек.

Среда разработки диаграмм: WhiteStarUML

Система контроля версий: Git

Сюжеты игры:

Игроку предстоит пройти путь по лабиринту, который представляет из себя несколько комнат, соединенных между собой узкими коридорами. В комнатах и коридорах будут встречаться противники и ловушки, которые будут мешать пройти путь игроку. На поле будут расставлены пули, подобрав которые, можно будет применить против противников, чтобы облегчить себе путь. У игрока, как и противников, есть здоровье, которые он может пополнить с помощью расставленных на поле специальных поинтов (health points). Один контакт с противником или с ловушкой не убивает игрока, но наносит большой урон, аналогично, если игрок атакует противников, он наносит урон, а не убивает моментально! Цель игры - остаться в живых и дойти до выхода.

Функционал главного исходного кода.

Прежде всего создаются экземпляр класса игрок и 3 экземпляра класса противник с привязкой изображения, начальных координат, ширины, высоты и имени. Затем, при запуске окна происходит объявление 4 таймеров, которые отвечают за игровое время (время между началом игры и концом - смерть или победа игрока), время обновления состояний объектов, время срабатывания ловушки и время бессмертия при получении урона. Таймер со временем срабатывания ловушки длится 2 секунды, затем вызывает функцию активацию (либо деактивацию ловушки).Таймер со временем бессмертия длится 2 секунды и обновляется каждый раз, при получении урона. Стрельба пулями игроком осуществляется при нажатии “p” с проверкой на наличие этих пуль у игрока. Пуля помещаются в лист и количество пуль у игрока уменьшается на один. Так как пуля, как и противник, является потомком класса родитель, то она также имеет ширину и длину, таким образом осуществляется проверка столкновения пули с противником. В случае, если пуля пересекается с противником, второму наносится урон, а сама пуля исчезает. Противник же также исчезнет, если будет иметь здоровье равное или меньше 0. Если противник или пуля исчезают (т.е. life = false), то их необходимо удалить, а вместе с ними и итераторы из списка, которые указывают на сами объекты. Когда игрок сталкивается с противником, также как и с пулями, проверяется пересечение между ними, только если таймер со временем бессмертия не работает (т.е. >2 сек.). Если они пересекаются, то игроку наносится урон и происходит обновления таймера бессмертия. В отсутствие этого таймера игроку моментально бы наносился огромный урон из-за того, что время обновления состояний объектов очень короткое. За счет этого времени происходит прорисовка всех объектов на игровом поле и обновление их состояния, которое определяется взаимодействием с окружением. В случае, если игрок дошел до назначенной точки и остался в живых, ему присваивается статус победителя и на экран выводится надпись “YOU WON”, а если игрок потерял все свое здоровье, не дойдя до конца, на экран выводится надпись “YOU DIED”. Помимо надписей с победой и поражением, на экране всегда отображается время игры, здоровье и количество пуль игрока.

Заключение:

В среде Visual Studio на языки С++ с подключенной библиотекой SFML была разработана программа, соответствующая описанию сюжета и поставленной цели. Были использованы наследование, конструкторы, итераторы и раздельное компилирование. Была реализована очистка памяти. В коде присутствуют неиспользуемая переменная moveTimer, а также в классе bullet не используется переменная health, наследуемая от Entity . В ходе разработки программы применялась система контроля версий Git. Весь листинг программы, диаграммы и скриншоты из git прилагаются.

