Министерство образования и науки РФ
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра «Информационные и управляющие системы»

Курсовая работа

по дисциплине «Технологии программирования»

Выполнил Стойкоски Н.С.
студент гр.23534/1

Руководитель Александрова О.В.

« 15 » декабря 2017 г

Санкт-Петербург
2017

Оглавление

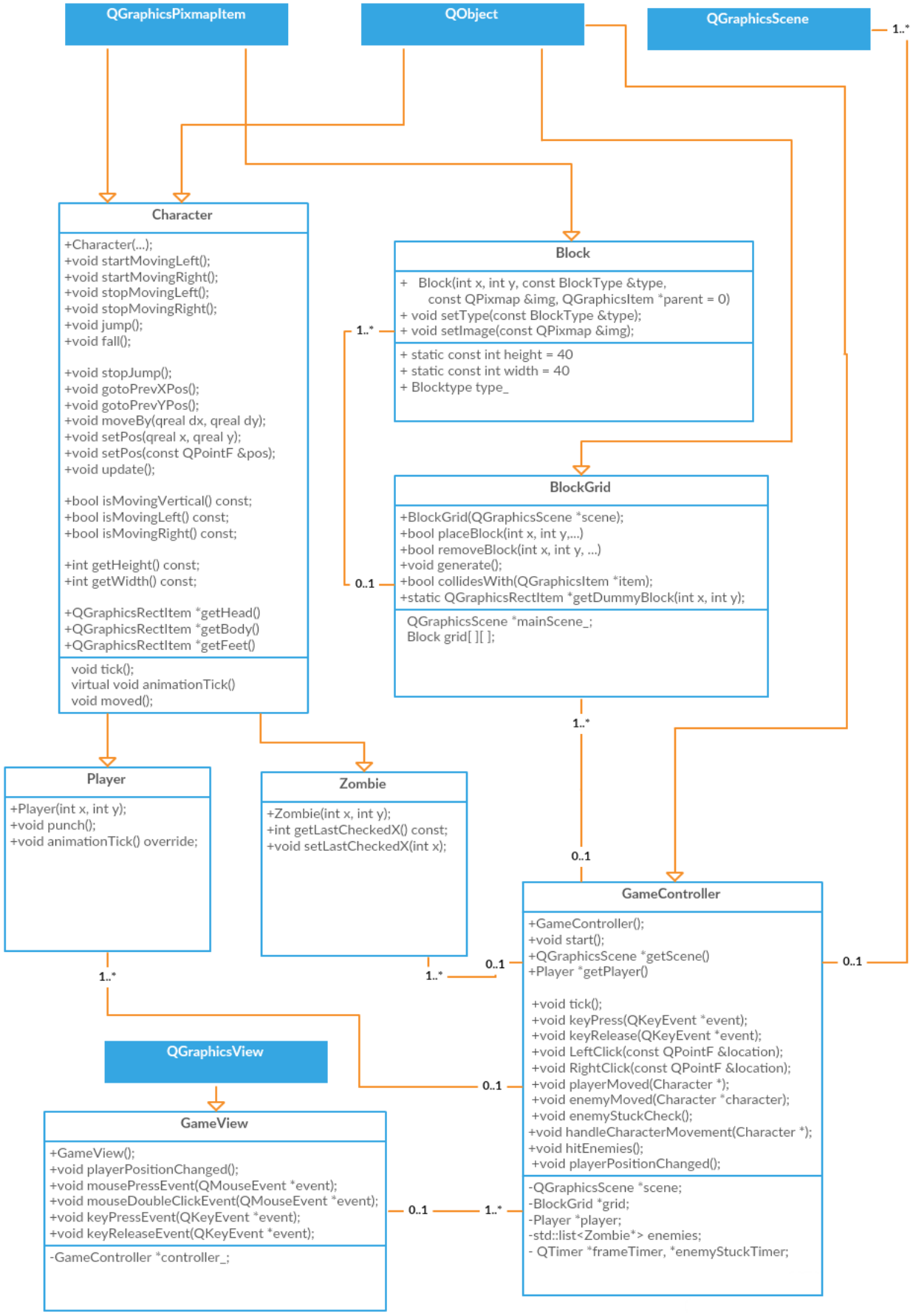
Техническое задание

Создать небольшая 2Д игра, где пользователь управляет персонажем игрока, который находится в рандомно-сгенерированная среда, состоящая из различных типов прямоугольных блоков. Движение игрока должно быть реализовано в отношении окружения - игра имитирует физику движения и столкновения реального мира. Карта игры так же содержит врагов, которые можно убить, либо от них убежать. Также возможно интерактивно взаимодействовать с окружающую среду - можно добавлять или удалять блоки. Управление движением персонажа осуществляется за счет нажатия стрелок на клавиатуре, либо кнопки WASD. Ударить врага - клавижа Space. Добавить/удалить блок в место положения указателя мыши - щелчок правой/левой кнопкой мыши.

Описание основных модулей

Программа написана на языке C++ в среде QT Creator 4.4.1. Применены принципы объектно-ориентированного программирования. В реализации данного приложения использованы класс Block, BlockGrid, Character, Player, Zombie, GameController, GameView.
Класс Block описывает модель базового строительного блока.
Класс BlockGrid – представляет окружение игры, составленное из блоков, т.е. содержит все блоки. Обеспечивает возможность добавлять и вставлять блоки. Предоставляет функция – проверка на столкновение. Рандомно генерирует первоначаьное окружение.
Класс Character – базовый клас - модель для игрок и врагов. Предоставляет множество функции с помощю которых можно контролировать движение карактера.
Класс Player – наследник класса Character, представляет персонаж игрока, предоставляет дополнительную возможность – ударение/атака врагов.
Класс Zombie – наследник класса Character, описывает вражеский объект.
Класс GameController – обеспечивает связь между все обьекты, и их представление в QGraphicsScene. Принимает сигналы из класса GameView в связи с нажатия разных кнопок. Контролирует модели классов Player и Zombie.
Класс GameView – обеспечивает отображение сцены на экране, воспринимает нажатия кнопок пользователя, и отправляет соответствующие сигналы обьекту GameController.
В программе преимущественно используются методы, содержащиеся в библиотеке QT.

Диаграмма классов



Алгоритмы

Основная часть алгоритма, отвечающий за управление движением карактеров. Выполняется при получение сигнала moved() из одного из карактеров (character). grid – экземпляр класса BlockGrid. Этот блок кода выполняется как для персонажа игрока, так и для врагов.

```
if(!grid->collidesWith(character->getFeet()))
{
  if(!character->isMovingVertical())
  {
    character->fall();
  }
}
else
{
  character->stopJump();
}
if(grid->collidesWith(character->getBody()))
{
  character->gotoPrevXPos();
}
if(grid->collidesWith(character->getHead()))
{
  character->gotoPrevYPos();
  character->fall();
}
```

BlockGrid::generate() – генерирует рандомый рельеф окружности, используя блоков типа Blocktype::dirt (почва). Следует псведо-код алгоритма который раскрашивает сетку BlockGrid путем измененения типа блоков в Blocktype::stone (камень). Использовалась модифицированная версия алгоритма BFS (Breadth First Search), который на каждом шаге меняет тип блока, и перестает развлетвление только когда радиус станет ноль.

```
struct QueueNode
{
  int x, y, radius;
};
queue<QueueNode> q;
bool visited[][] = {false};

/* добавление первоначаьные блоки */
for(x=0; x < maxHorizontalBlocks; x+=20)
  for(y=groundPos; y < maxVerticalBlocks; j+=20)
    q.push(x, y, rand() % 7);
```

```
while(!q.empty())
{
  QueueNode cur = q.front();
  q.pop();

  if(cur.radius > 0 && !visited[cur.x][cur.y])
  {
    visited[cur.x][cur.y] = true;
    setToStone( grid_[cur.x][cur.y] );

    /* развлетвление */
    q.push(cur.x+1, cur.y, cur.radius-rand()%3);
    q.push(cur.x-1, cur.y, cur.radius-rand()%3);
    q.push(cur.x, cur.y+1, cur.radius-rand()%3);
    q.push(cur.x, cur.y-1, cur.radius-rand()%3);
  }
}
```

Инструкции пользователя

После запуска программы открывается окно игры. Движение персонажа игрока обеспечивается через нажатия клавиш со стрелками (влево, вправо, вверх), либо клавиши WASD. Навстречу игрока могут подойти враги, которые необходимо убыть. Удар производится путем нажатия клавиши space. Добавить/удалить блок в место положения указателя мыши - щелчок правой/левой кнопкой мыши.

Вывод

В результате проведенной работы был реализован проект игры с использованием методов объектно-ориентированного программирования, с применением библиотеки QT. Был реализован способ управление персонажем игрока и взаимодействие с окружающая среда, которая рандомно генерируется, а так же содержит множество врагов. Корректно реализована физика движения и столкновения.

Приложение (скриншоты)