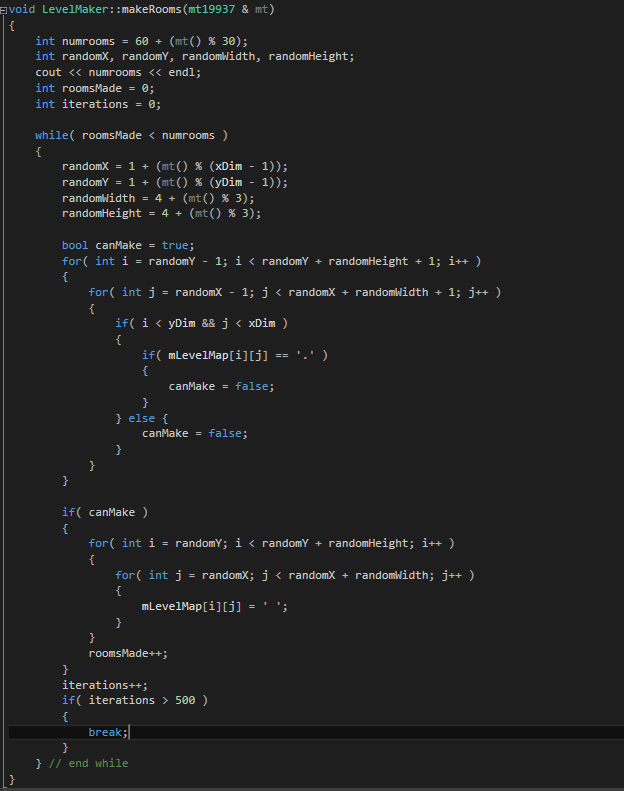
Open-Style Level Design Algorithm

For a Rogue-like Game



**/\* CODE \*/**

void LevelMaker::makeRooms(mt19937 & mt)

{

int numrooms = 60 + (mt() % 30);

int randomX, randomY, randomWidth, randomHeight;

cout << numrooms << endl;

int roomsMade = 0;

int iterations = 0;

while( roomsMade < numrooms )

{

randomX = 1 + (mt() % (xDim - 1));

randomY = 1 + (mt() % (yDim - 1));

randomWidth = 4 + (mt() % 3);

randomHeight = 4 + (mt() % 3);

bool canMake = true;

for( int i = randomY - 1; i < randomY + randomHeight + 1; i++ )

{

for( int j = randomX - 1; j < randomX + randomWidth + 1; j++ )

{

if( i < yDim && j < xDim )

{

if( mLevelMap[i][j] == '.' )

{

canMake = false;

}

} else {

canMake = false;

}

}

}

if( canMake )

{

for( int i = randomY; i < randomY + randomHeight; i++ )

{

for( int j = randomX; j < randomX + randomWidth; j++ )

{

mLevelMap[i][j] = ' ';

}

}

roomsMade++;

}

iterations++;

if( iterations > 500 )

{

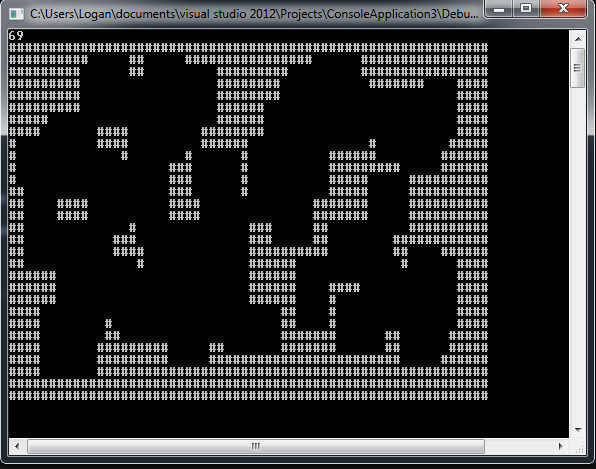
break;

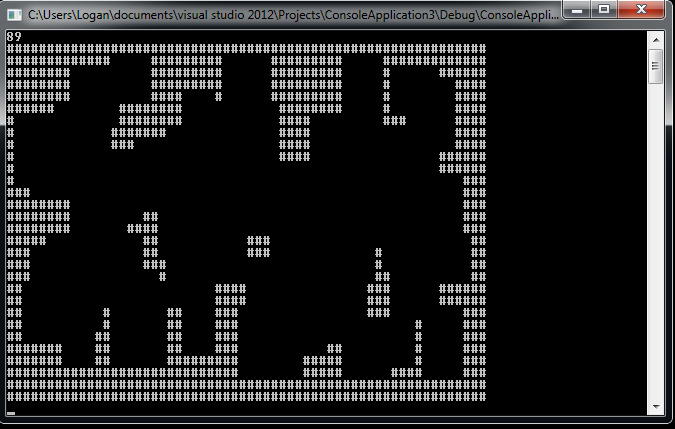
}

} // end while

}

**Sample Output:** The number in upper left is the number of “rooms” constructed.

****

****

****

****