“Stupid Robot” Code Samples

Gameboard Draw Code

bool gameBoard::draw()

{

blackImage->drawImage(); // clears the background

int imageWidth = 144; // we know how big our initial images are

int imageHeight = 72;

imageWidth \*= scale; // adjust image size according to scale

imageHeight \*= scale;

int hw = (imageWidth/2); // set half width

int hh = (imageHeight/2); // set half height

int basex = 0; // set our initial values

int basey = 0;

int drawAtX = 0;

int drawAtY = 0;

// this loop here runs through the game board’s x and y positions and draws them at the appropriate location

for(int x = 0; x < Width; x++) // iterate through the board

{

basex = x\*hw; // the base x and y pixel positions are based off of half the image size

basey = x\*hh; // times the position you are trying to draw at

vector<mapTile\*>::iterator itr = mapList[x].begin();

for(int y = 0; y < Height; y++)

{

// here’s the meat and potatoes of what we’re doing, in order to draw an isometric grid, we have to

// take into count the fact that with a map rotated 45 degrees visually it is possible to have

// an x position in the grid that is high but is still drawn to the left of the 0,0 tile position

// due to having a high y position in the grid

// for drawATtX we first find the initial X position which is x\*hw, then we subtract it’s y position \* hw

// this sets the proper pixel X draw position.

drawAtX = mapOffsetX + (x \* hw - (y \* hw) + (hw));

drawAtY = mapOffsetY + (y \* hh) + (x \* hh);

if((\*itr)->getIsActive())

drawTile((\*itr)->getType(), drawAtX, drawAtY, scale, true);

else

drawTile((\*itr)->getType(), drawAtX, drawAtY, scale, false);

itr++;

}

}

return true;

}