**Quad Tree**

**Application2D.h**

1. Put a pointer to quadtree in game.

**Application2D.cpp**

1. Quad tree is created in Game’s constructor and passed top left and bottom right of level.
2. When enemies are created in the constructor, each one needs to be added to the Quad Tree.
3. Quad tree should be deleted in destructor.
4. Draw should call the Quad Tree’s draw function.

**QuadTree**

1. Constructor should create the root node, giving it the top left and bottom right that were passed in.
2. Destructor should delete the root node.

void QuadTree::AddEnemy(Enemy\* pEnemy)

{

Add enemy to root node.

}

void QuadTree::Draw()

{

Draw root node.

}

**Node**

Node::Node(float fLeft, float fTop, float fRight, float fBottom)

{

Store the left, top, right and bottom in the node’s topLeft and bottomRight vectors

LOOP through all child nodes

Set child node to null

END LOOP

Set Enemy pointer to null

}

Node::~Node()

{

LOOP through all child nodes

Delete child node

END LOOP

}

void Node::Draw()

{

Draw this node’s square (see DrawSquare() in HelperFunction.h)

LOOP through all child nodes

Draw child node

END LOOP

}

bool Node::IsInside(Vector2 v2Pos)

{

IF (v2Pos is inside this node’s square)

Return true

ELSE

Return false

}

void Node::AddEnemy(Enemy\* pEnemy)

{

//Child quads already exist, already subdivided.

IF child nodes exist

//Loops through all children and find which one contains the enemy.

LOOP through child nodes

IF enemy is inside child node

Add enemy to that child node by recursively calling this function on it.

END IF

END LOOP

ELSE

//No child quads, not subdivided.

//Check if this child already has an enemy.

IF this node doesn’t have an enemy already

Store enemy in this node

ELSE

//The node already has an enemy so we need to subdivide to make room.

Create 4 child nodes and store them in pointers in this node.

//These two things can both be done with a recursive call to this function

Relocate this node's enemy to one of its 4 children.

Find a place for the actual enemy we are trying to add.

Set the enemy pointer in this node to null since it has now been moved to one

of its children

END IF

END IF

}