**SDL Tower Defense Code Highlight**

**Pathfinding**

**Pathfinding algorithm in pathfinding.cpp lines 23-125.**

Algorithm uses A\* and the Manhattan heuristic to find the shortest path in a grid of map tiles. It also does not allow diagonal movement across obstacles. It uses a heap to store the open set so that the lowest cost tile can stay at the top of the heap. The game was built around practicing with the A\* algorithm.

[First tests with pathfinding](https://puu.sh/yf1zR/9192c3dbbc.png), program randomly created blocked tiles, user could click on 2 positions to draw a path. Red is in the open set, green in the closed set, and blue is the path.

[Testing with enemies finding paths](https://puu.sh/ykdqj/16be8abc15.gif). At the time diagonal movement was still allowed across corners and had enemies cutting through.

Finally, in the [current version of the game](https://puu.sh/yuakq/d4c16a75b8.mp4), enemies now only do diagonal movement when it does not cross over an occupied tile.

**Collisions**

**Collision algorithms found in quadtree.cpp lines 32 – 178.**

Collisions were done using a quad tree containing entities in the game with their own bounding boxes. If tree reaches max capacity of 4 items, it will subdivide and try to insert those items into the sub trees.

[First collisions tests](https://puu.sh/ylreg/2cdd48456e.gif) were done by inserting points into the tree, and using SDL to render some basic rectangles around the bounding boxes.

The game was reduced to only drawing bounding boxes and the path, enemies were added to the quad tree, and towers would query it with the bounding box of their range to [see what enemies are in range](https://puu.sh/ylv6W/6ba37a370f.gif).

Towers then were updated to choose a [random enemy in range to attack](https://puu.sh/ymplu/bc0bfd7d2f.gif) then updated to [fire projectiles](https://puu.sh/ylAzO/88f08dd848.gif) that queried the tree with their own bounding boxes. Tests here and here.

Finally, in the [current version of the game](https://puu.sh/yuaTs/f76f1949c5.mp4) towers, projectiles and the mouse check enemies for collisions.