TODO 1:

fog\_layer = new FOG\_TYPE[App->map->data.width \* App->map->data.height];

for (uint y = 0; y < App->map->data.height; y++)

{

for (uint x = 0; x < App->map->data.width; x++)

{

fog\_layer[y \* App->map->data.width + x] = DARK\_FOG;

}

}

TODO 2:

return fog\_layer[y \* App->map->data.width + x];

if (App->fog\_of\_war->GetFogID(tiles\_in\_view[k].x,tiles\_in\_view[k].y) == DARK\_FOG)continue;

TODO 3:

if (type == ENEMY && fog\_type != NO\_FOG)return; /\* Enemy entities are only drawn when there's no fog \*/

else if (type == NEUTRAL && fog\_type == DARK\_FOG)return; /\* Neutral entities aren't drawn under dark fog \*/

TODO 4:

std::vector<AlphaCell\*> in\_view\_cells; uint size = fog\_quadtree.CollectCandidates(in\_view\_cells, App->render->camera\_viewport);

for (uint k = 0; k < size; k++)

{

App->render->FogBlit(in\_view\_cells[k]->position, alpha\_cell\_size, in\_view\_cells[k]->alpha);

}

TODO 5:

else if (type == ALLY)

{

App->fog\_of\_war->ClearAlphaLayer(vision\_area, 0);

}

TODO 6:

std::vector<MyEntity\*> entities\_vec;

uint size = App->entities\_manager>entities\_quadtree.CollectCandidates(entities\_vec, vision\_area);

TODO 7:

if (entities\_vec[k]->type == NEUTRAL)

{

App->fog\_of\_war->ClearAlphaLayer(entities\_vec[k]->vision\_area, MID\_ALPHA);

App->fog\_of\_war->ClearFogLayer(entities\_vec[k]->render\_area, GRAY\_FOG);

}

TODO 8:

if (in\_view\_cells[k]->alpha < MID\_ALPHA)in\_view\_cells[k]->alpha = MID\_ALPHA;