

# Terrarium

A simulator for ants

# Observer/Singleton Design Pattern

Subject (Engine)

```
class Engine {
public:
    ~Engine() {}
    bool overlay;
    float ph;
    float th;
    int w;
    int h;
    float zoom;
    float dzoom;
    float dph;
    float dth;
    int pixlight;
    int window;
    SDL_Window* context;
    SDL_GLContext* passive;

    #handleEventSO
    +physics()
    +display()
    +setPaused()
    +getQuit()
    +setOverlay()
    +moveCamera()
    +start()
    +Render()
    +Render()
    +reshape()
    +addObject()
    +keyStateUpdated()
    +setW()
    +setH()
    +setZoom()
    +getZoom()
    +setDzoom()
    +getDzoom()
    +setTh()
    +setPh()
    +setDth()
    +getDth()
    +setDph()
    +getDph()
    +Pause()
    +Render()
}
```

```
class Renderable {
public:
    +animate()
    +render()
}
```

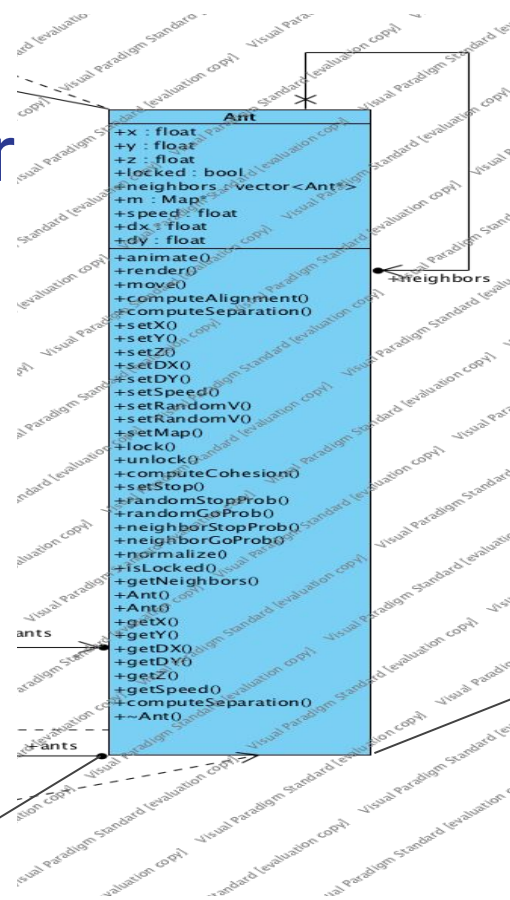
```
class Ant {
public:
    float vx;
    float vz;
    float locked;
    vector<Ant*> neighbors;
    float m;
    float speed;
    float dx;
    float dy;
    +animate()
    +render()
    +move()
    +computeAlignment()
    +computeSeparation()
    +setX()
    +setY()
    +setZ()
    +setDX()
    +setDY()
    +setSpeed()
    +setRandomV()
    +setRandomV()
    +setMap()
    +lock()
    +unlock()
    +computeCohesion()
    +setStop()
    +setRandomStopProb()
    +randomGoProb()
    +neighborStopProb()
    +neighborGoProb()
    +normalize()
    +isLocked()
    +getNeighbors()
    +Ant()
    +Ant()
    +getX()
    +getY()
    +getDX()
    +getDY()
    +getSpeed()
    +computeSeparation()
    +Ant()
}
```

Observers

```
class Map {
public:
    ~Map() {}
    int width;
    int height;
    float* heightmap;
    vector<Ant*> ants;
    vector<Tile*> tiles;
    +getState()
    +animate()
    +render()
    +generate()
    +Map()
    +getW()
    +getH()
    +setW()
    +setH()
    +addAnt()
    +getNeighbors()
    +Map()
    +getHeight()
    +getState()
    +getNeighbors()
    +Map()
    +setNeighbors()
}
```

```
class Tile {
public:
    ~Tile() {}
    int state;
    +animate()
    +render()
    +getState()
    +setState()
    +Tile()
    +Tile()
}
```

# State-like Behavior



# Blob?

