

Agent-based Modelling and Simulation 4

Last Time

- Coding agent-based models
- A closer look at the Boids rules
 - Separation
 - Cohesion
 - Alignment
- All of these are just velocity contributions

Today

- Assignment 3
- Crash course in the C++ STL
- What to do when agent-based simulations get slow

Assignment 3

- Flocking simulation
- Download a3.cpp and support.h
- Write the missing code

Crash course in the C++ STL

- First, what is the difference between C / C++?
 - C++ is C with a bit more.
 - Classes
 - STL
 - Some other syntax improvements such as
 - `for (int i = 0; i < 10; i ++)`
 - C code can be compiled as C++ (not vice versa)

STL

- Standard Template Library
- A giant software library shipped with every C++ compiler
- It contains:
 - Algorithms
 - Containers
 - and lots more.

STL Containers are very useful

- A container stores a collection of other elements, which can be ints, floats, or composite objects.
- Examples:
 - vector
 - queue
 - stack
 - priority_queue
 - list
 - set, etc

The STL Vector

- 159.201 covers how these datastructures are implemented, and how their algorithms work.
- We will use vectors to replace the common array
- This will make our lives easier

The STL Vector

- Arrays are great
- We can allocate them dynamically or statically (stack or heap)
- We can modify them as we please
- But what do we do if we run out of space?

Agent agents[50];

- And we add 60 agents during the running of the program

The STL Vector

- We can allocate a new array, and copy the old memory there, and delete the old one
- We can store more things in an separate array
- What if we want to delete one item in the middle?
- Complicated, but we can do it
- We just have to do it manually

The STL Vector

- The STL Vector is designed for this particular purpose.
- Use the STL vector by doing:
`#include <vector>`

The STL Vector

- Declare a vector like this:

```
std::vector<int> vec;
```

The STL Vector

- Add items to a vector like this:

```
vec.push_back(92);
```

```
vec.push_back(21);
```

```
vec.push_back(566);
```

The STL Vector

- Read items from a vector like this:

```
printf ("%d\n", vec[0]);
```

- Sort items in a vector like this:

```
std::sort (vec.begin(), vec.end());
```

Looping through an STL Vector

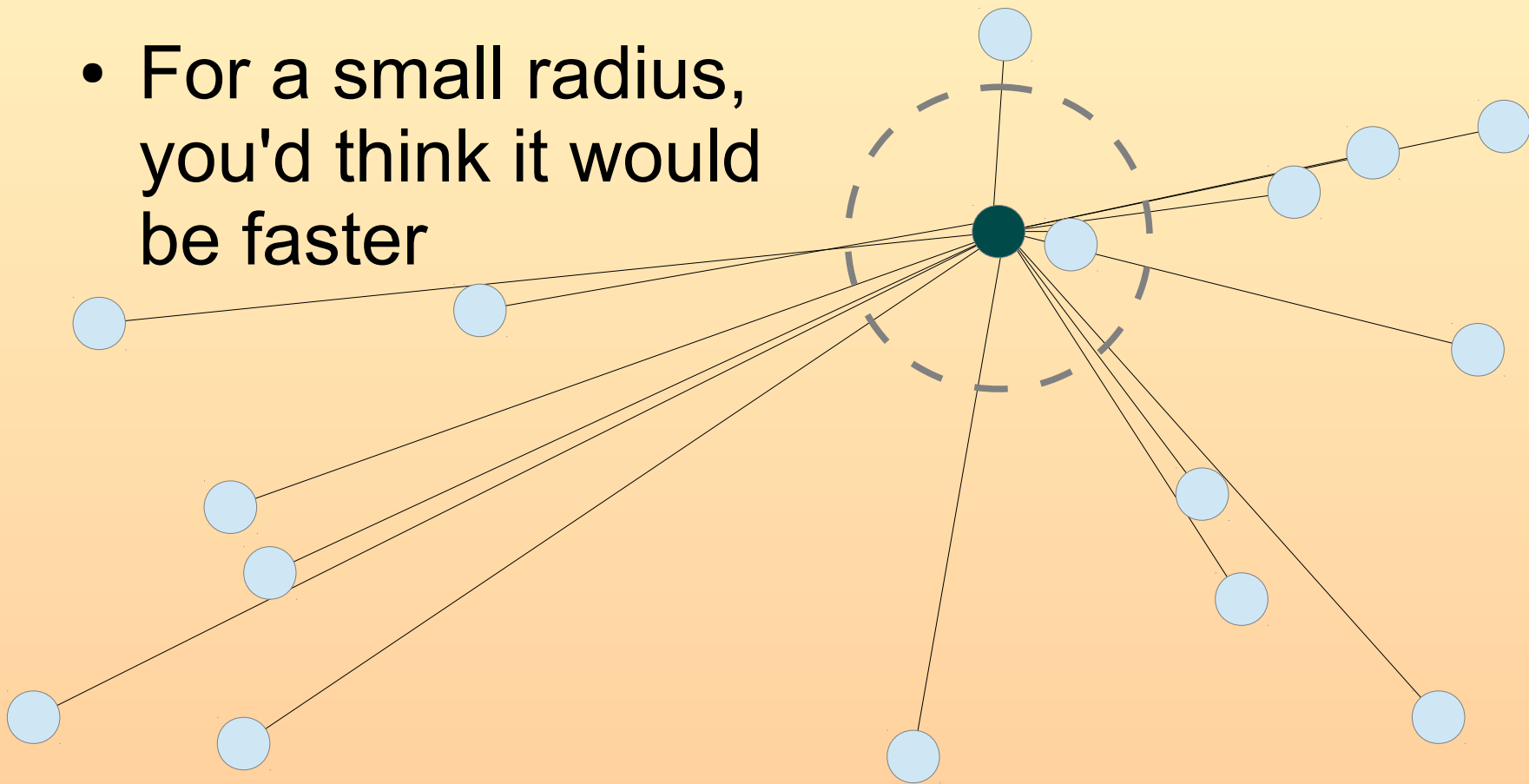
- A bit more complicated, but easier thanks to C++11

```
for (auto it = vec.begin(); it != vec.end(); ++it)  
    printf("%d\n", *it);
```

What to do when ABMs are slow

Agent-based simulations are slow...

- When there are a lot of agents all communicating with each other
- For a small radius, you'd think it would be faster



Complexity

- Agent-based modelling usually has a time complexity of $\mathcal{O}(n^2)$
- We can deal with it using spatial partitioning
 - Uniform Grid Partitioning
 - Octrees / Quadtrees
 - K-D trees

