Spatial Simulation

Winter Semester 2023 / 24, MSc Applied Geoinformatics

Movement and action neighbourhoods

Movement

- wander

  Random movement with the given speed

  ```
  do wander speed: 2;
  ```

  Correlated random movement

  ```
  do wander amplitude: 90.0 speed: 2;
  ```

- goto

  Towards a target with the given speed

  ```
  do goto target: {x,y};
  ```

- move

  Moves in the giving speed into the direction of the given heading

  ```
  do move;
  ```

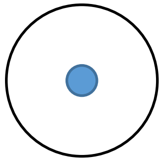
Action neighbourhood

= the space, where an agent potentially can be in the next time step

- wander

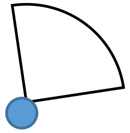
  Random movement with the given speed

  ```
  do wander speed: 2;
  ```

  Correlated random movement

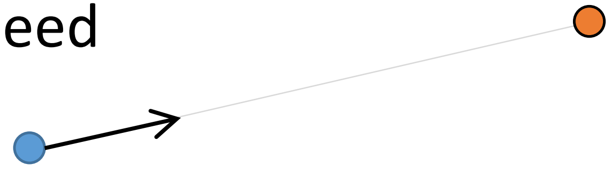
  ```
  do wander amplitude: 90.0 speed: 2;
  ```

- goto

  Towards a target with the given speed

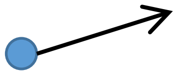
  ```
  do goto target: {x,y};
  ```

- move

  Moves in the giving speed into the direction of the given heading

  ```
  do move;
  ```

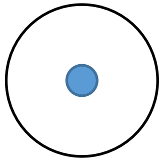
Visualise action neighbourhoods: wander

Random walk

Random movement into all directions (like gas molecules).

The action area is a circular buffer:

```
action_area <- circle(speed);
```

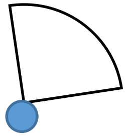


Correlated random walk

Random movement, but can‘t turn more than x degrees left / right of the current heading (like moving animals)

The action area is a cone:

```
action_area <- circle(speed) intersection cone(deg, deg);
```



Visualise action neighbourhoods: move

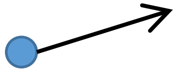
Moving straight ahead

The action area is a line:

```
action_area <- line(self.location, self.location+vector);
```

In GAMA, a vector is written like x,y point coordinates, e.g.:

```
{2,5}
```



Visualise action neighbourhoods: goto

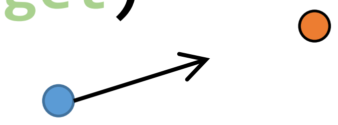
Movement towards a target

The action area is a line:

```
action_area <- line(self.location, location of target);
```

The length of the line is determined by the speed of the agent. So you need to intersect the line with a circular speed buffer:

```
action_area <- line(self.location, location of target)
intersection circle(speed);
```



GAMA works with image coordinates with an origin {0,0} top left. A point 2 steps down and 5 steps to the right is: `{2,5}`

Variables of type geometry

Working with spatial data, we can make use of a new variable type: geometry

```
geometry action_area;

reflex update_actionArea {
   action_area <- circle(speed) intersection cone(heading - 45, heading + 45);
}

aspect action_neighbourhood {
   draw action_area color: #goldenrod;
}
```

Draw layers on top of each other

In the species, define 2 aspects, one for the agent, and one for its action area

In the Experiment, visualise both aspects:

```
species cows aspect: default;
species cows aspect: action-area transparency: 0.5;
```