Event-driven Programming

Usefulness

- Up til now, most of your coding has been serial:
  - Stop everything when you reach a wall
  - Start driving when the button is pushed
  - Turn 90 degrees when the intersection is crossed

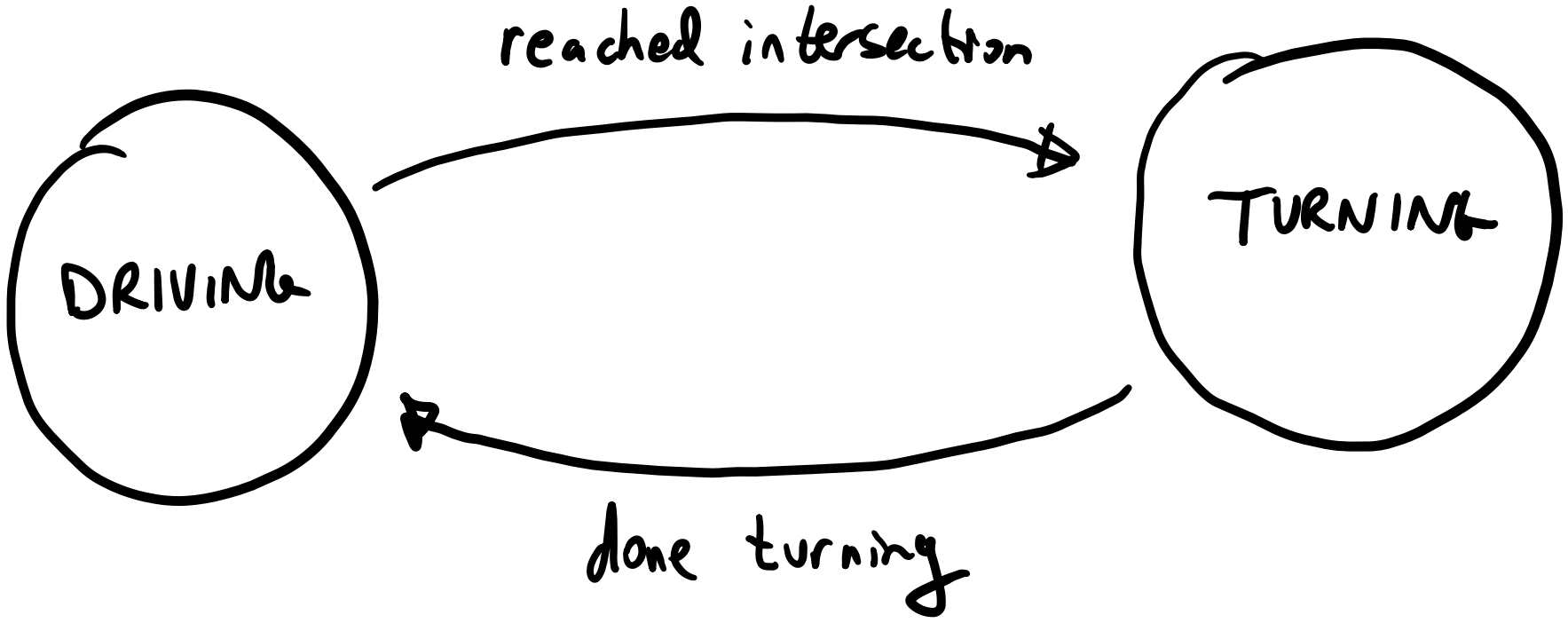
Usefulness

- Ultimately, you need to be able to do the same things over and over
- Imagine delivering multiple pizzas:

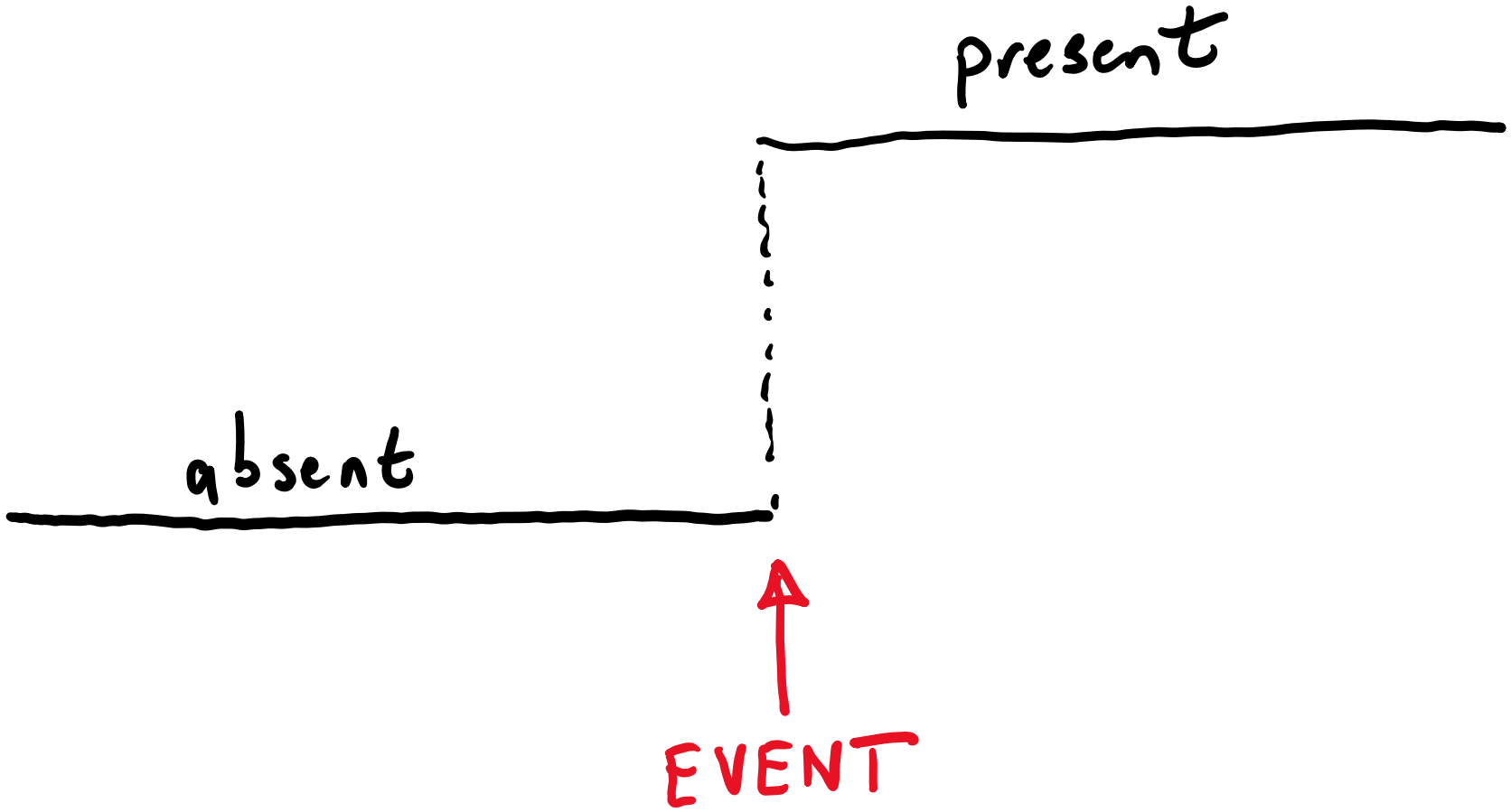
```
while(!intersection) {Drive();}
TurnLeft();
while(!intersection) {Drive();}
LiftPizza();
.
.
.
```

A better way

- A better way would be to define a set of *states* – what the robot is doing – and develop a way to manage transitions from state to state



An event is a *change* of a condition



Truth table

		Previous condition	
		present	absent
Current condition	present	F	arrival
	absent	F	F

Important points

- Checkers and handlers must be fast

```
bool DetectObjectEvent(int pin)
{
  bool returnValue = false;

  bool currObject = digitalRead(pin);

  delay(1000); // Bad code! No biscuit!

  if(!prevObject && currObject) returnValue = true;

  prevObject = currObject;

  return returnValue;
}
```

Important points

- Maintain a “record” of the previous state

```
bool prevObject = false;

bool DetectObjectEvent(int pin)
{
  bool returnValue = false;

  bool currObject = digitalRead(pin);

  if(!prevObject && currObject) returnValue = true;

  prevObject = currObject;

  return returnValue;
}
```

Important points

- Read a sensor only once in a checker

```
bool DetectObjectEvent(int pin)
{
  bool returnValue = false;

  bool currObject = digitalRead(pin);

  if(!prevObject && currObject) returnValue = true;

  prevObject = digitalRead(pin); //What if pin changed?

  return returnValue;
}
```

Important points

- It’s better to use static variables

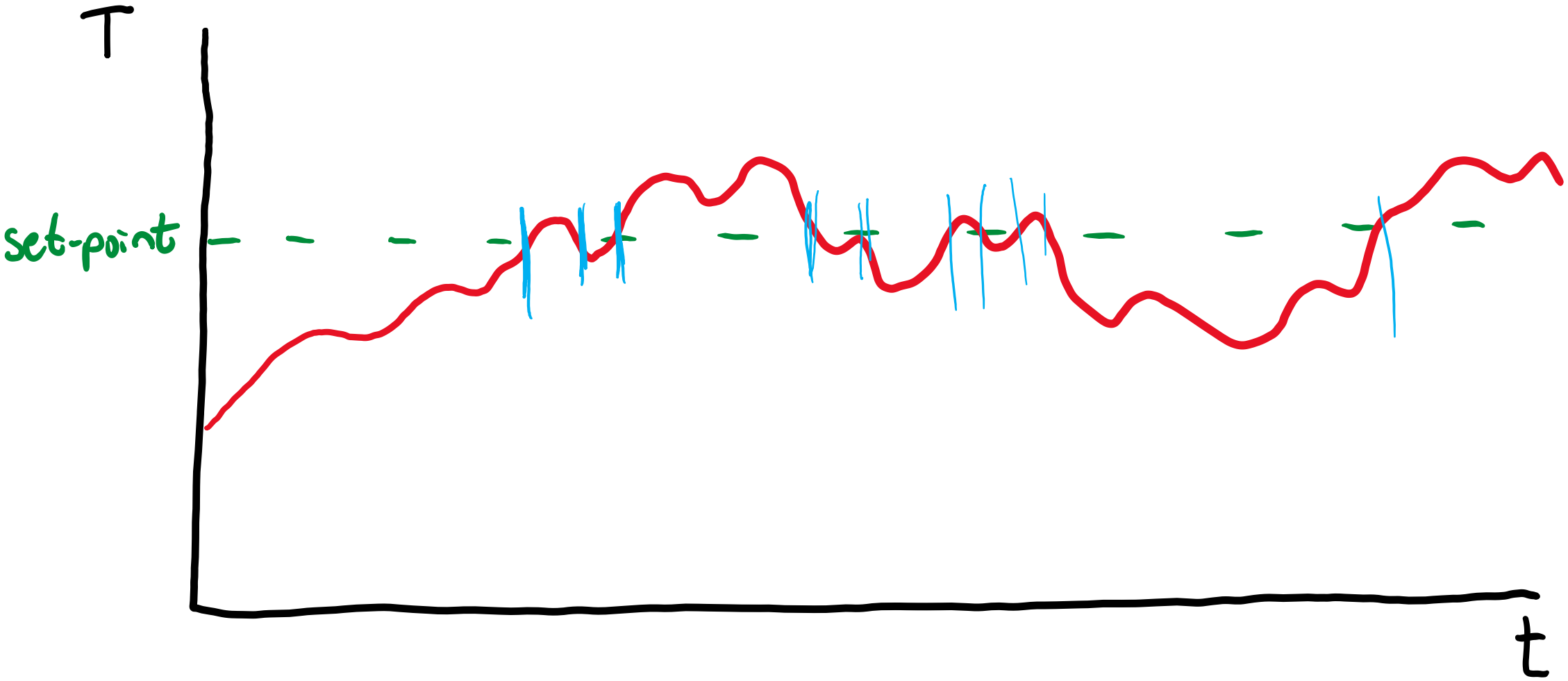
```
bool DetectObjectEvent(int pin)
{
  static bool prevObject = false; //static gives global persistence

  bool returnValue = false;

  bool currObject = digitalRead(pin);

  if(!prevObject && currObject) returnValue = true;

  prevObject = currObject;

  return returnValue;
}
```

Important points

- Declare a return value and carry it through the whole code

```
bool DetectObjectEvent(int pin)
{
  static bool prevObject = false; //static makes it global

  bool returnValue = false;

  bool currObject = digitalRead(pin);

  if(!prevObject && currObject) returnValue = true;

  prevObject = currObject; //need to be sure to get to this line

  return returnValue;
}
```

Hysteresis is used to smooth noisy events



Hysteresis can be used to smooth noisy events