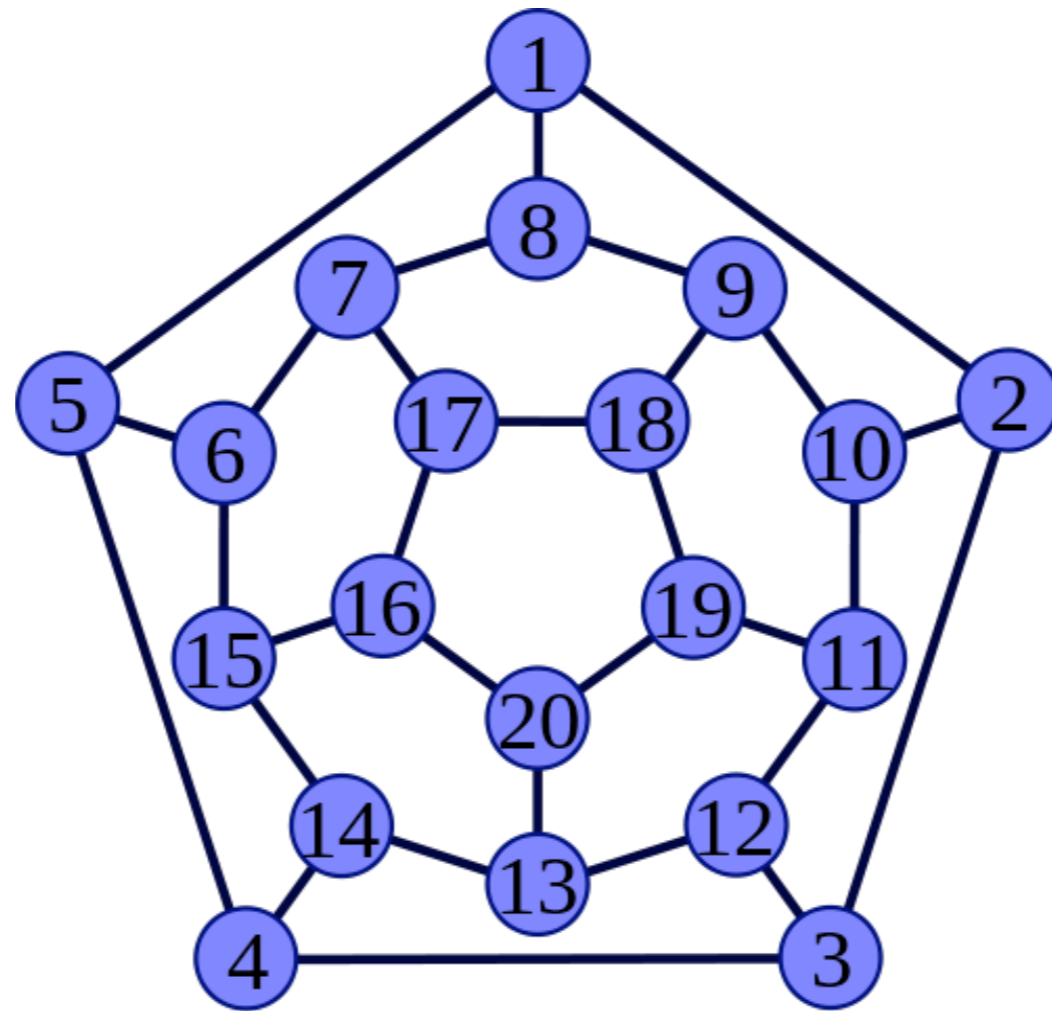


# Object Oriented Design

COMPI400 - Week 10

# Hunt the Wumpus

[http://en.wikipedia.org/wiki/Humpus](http://en.wikipedia.org/wiki/Hunt_the_Wumpus)



# Hunt the Wumpus

1. Start in a random room.
2. Move from room to room.
3. Avoid hazards:
  - Bats move you randomly,
  - The Pit kills you,
  - The Wumpus eats you.

# Hunt the Wumpus

4. Warning messages show nearby hazards.
5. Shoot arrows into nearby rooms.
6. Hit the wumpus to win.
7. You lose if you die or run out of arrows.

# Data

We to represent the following data:

- The connectivity of the map
- The positions of the bats, pit and wumpus
- The player's position
- The number of arrows left

# Design principles

## Abstraction:

- Divide code and data into **independent chunks**.
- **One idea** per object or method

## Encapsulation:

- Keep code close to the data it uses.

# Classes

Player

myMap

myRoom

myArrows

Map

myRooms

Input

Room

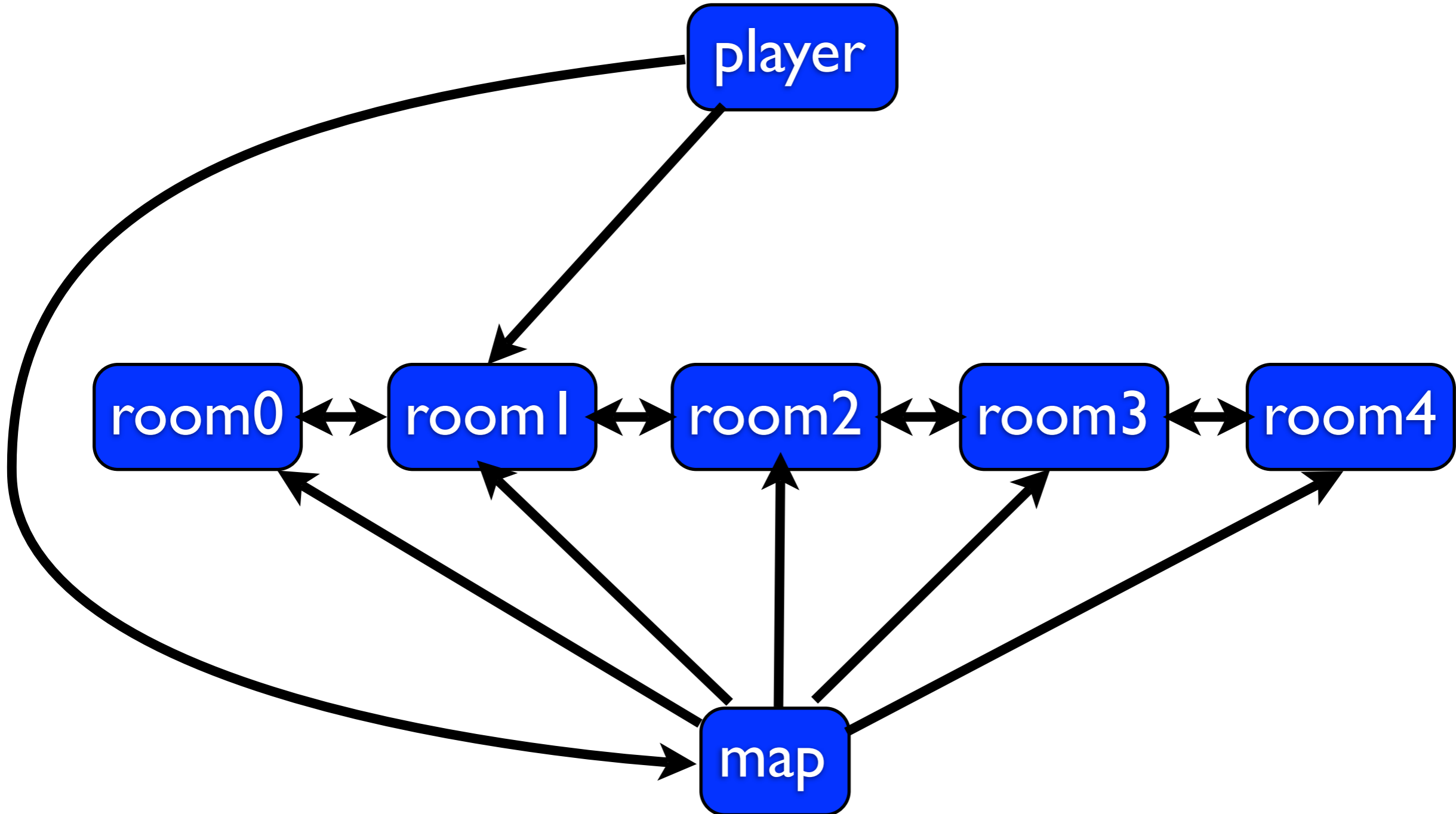
myNeighbours

myHasBats

myHasPit

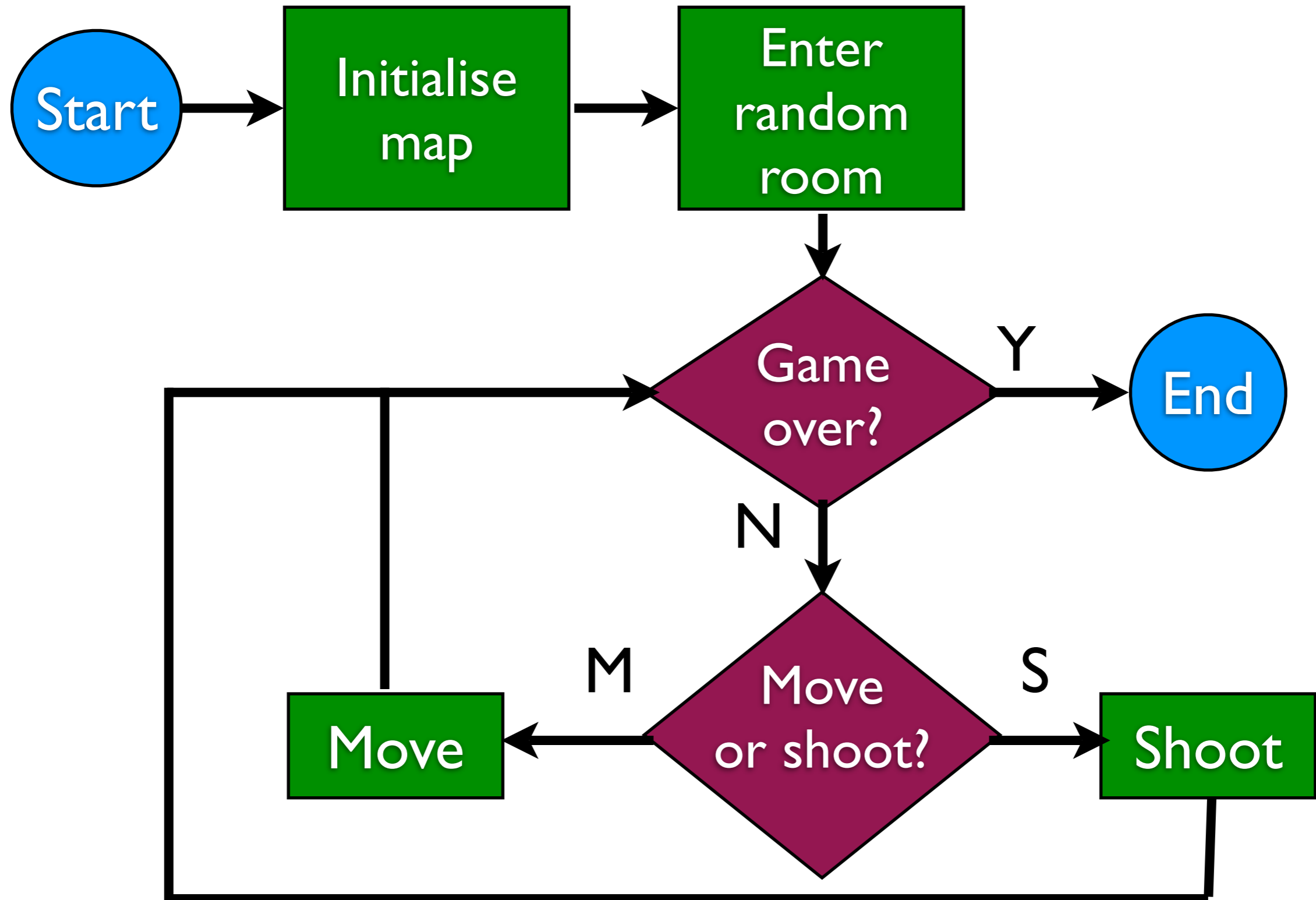
myHasWumpus

# Data Structure

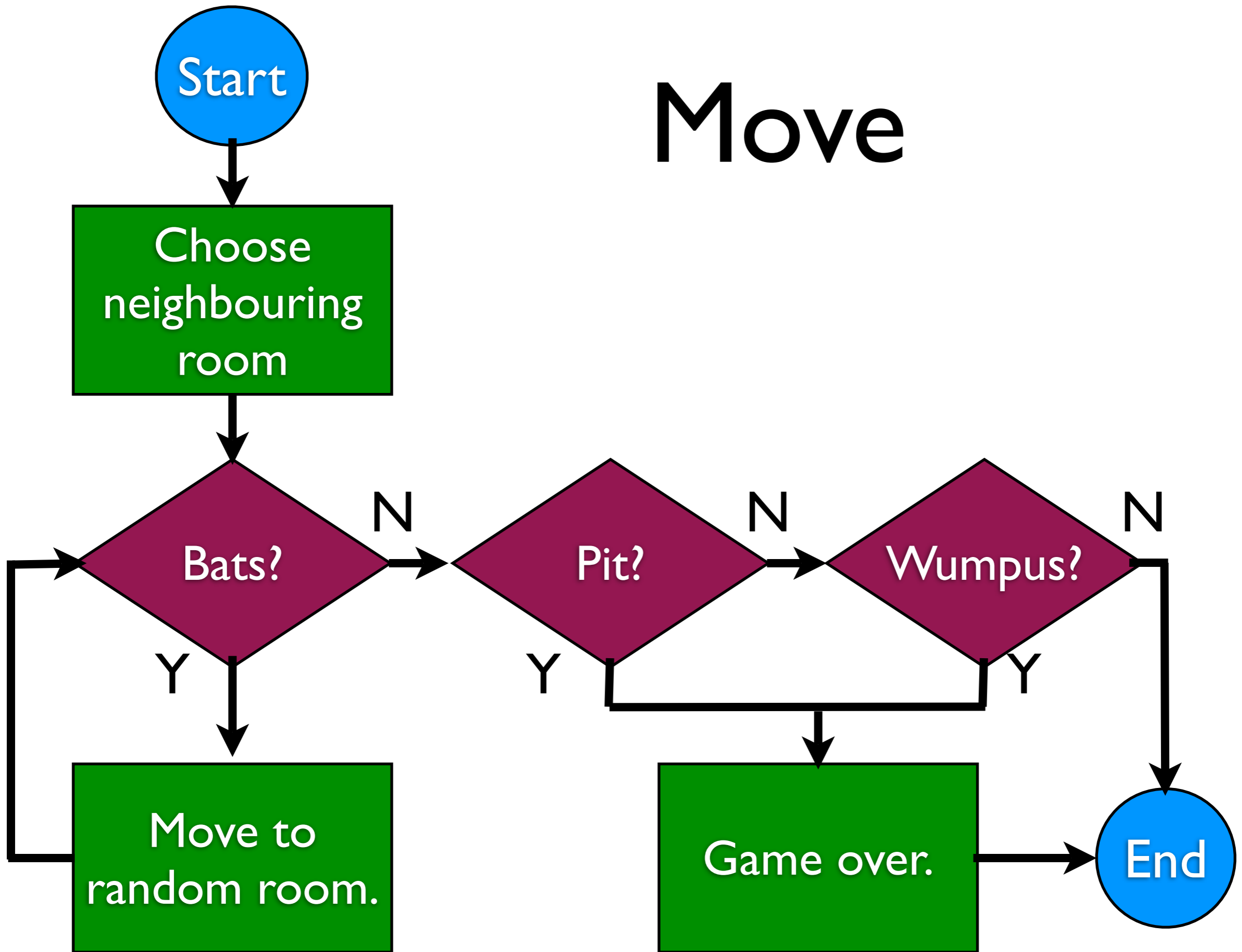




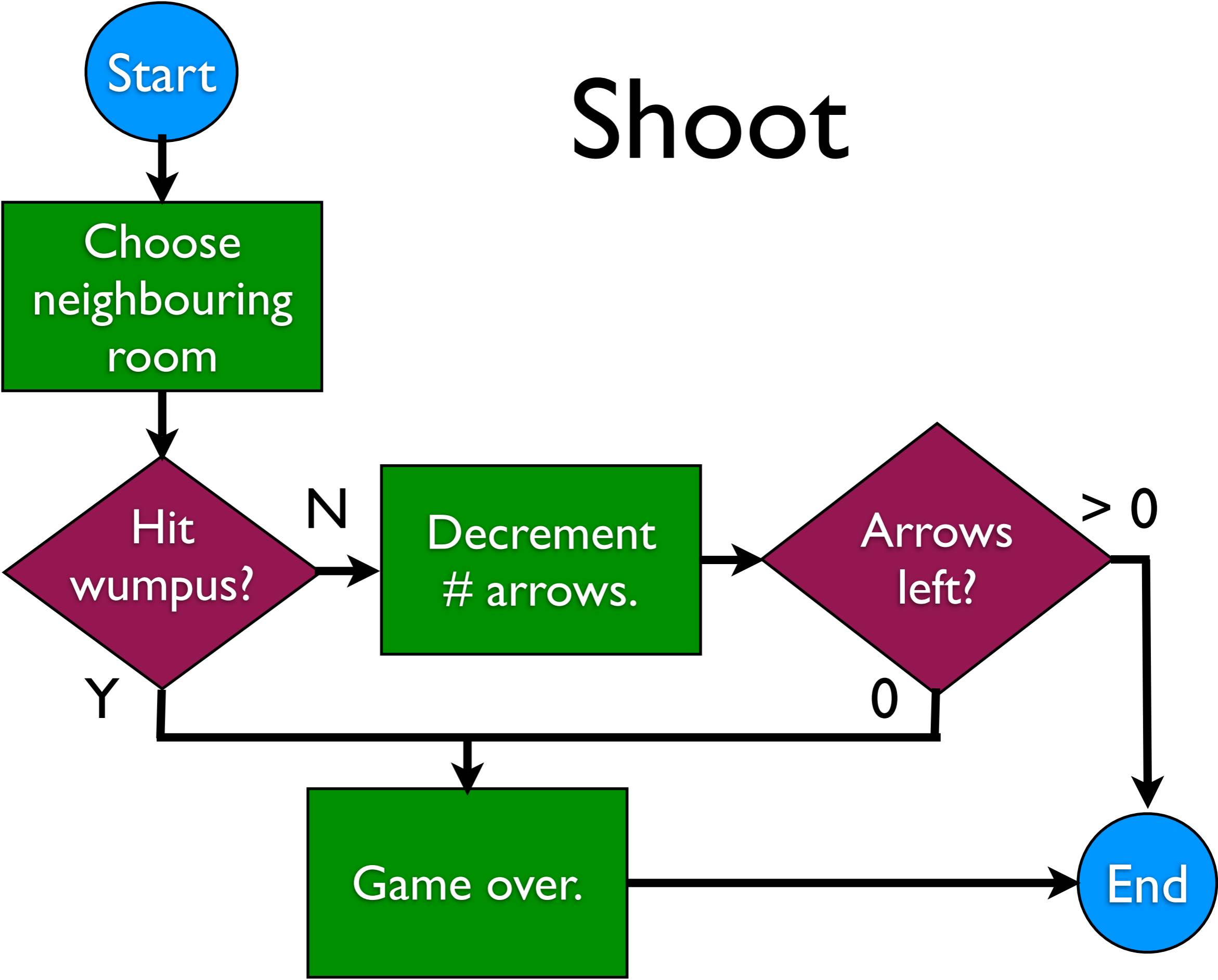
# Flow chart



# Move



# Shoot



# The main method

When you write a stand-alone java program you need to create a **main method**.

The main method is the starting point for executing a program.

# The main method

The main method has a standard syntax:

```
public static void main(  
    String args[]) {  
    // start here  
}
```

# The main method

Things to note:

- main is public, so it can be externally accessed.
- main is static, so it can be run without creating an object (although it must belong to a class)
- main has return type void

# Running a java program

From the command line:

- Create a JAR file
- Select the main method
- Run from terminal:

```
% java -jar Wumpus.jar
```

# The main method

The `args` parameter to `main` is an array of strings.

Each string is one **command line argument** to the program:

```
% java -jar Wumpus.jar Malcolm 27
```

```
// args[0] == "Malcolm"
```

```
// args[1] == "27"
```