What does it mean to be OO?

Comp-304 : What do is mean to be Object Oriented?
Lecture 5

Alexandre Denault
Original notes by Hans Vangheluwe
Computer Science
McGill University
Fall 2006

Late Policy

Forgot to mention my late policy for assignment.

For each day an assignment is late, your maximum grade is reduce by 15%.

Thus, if an assignment is one day late, your maximum grade is 85%. If it is 3 days late, your maximum grade is 55%.

After 7 days, it's not worth handing in the assignment anymore.

What does it mean to be OO?

- What are the characteristic of Object Oriented programming?
- What does Object Oriented programming add (as opposed to structure programming?)

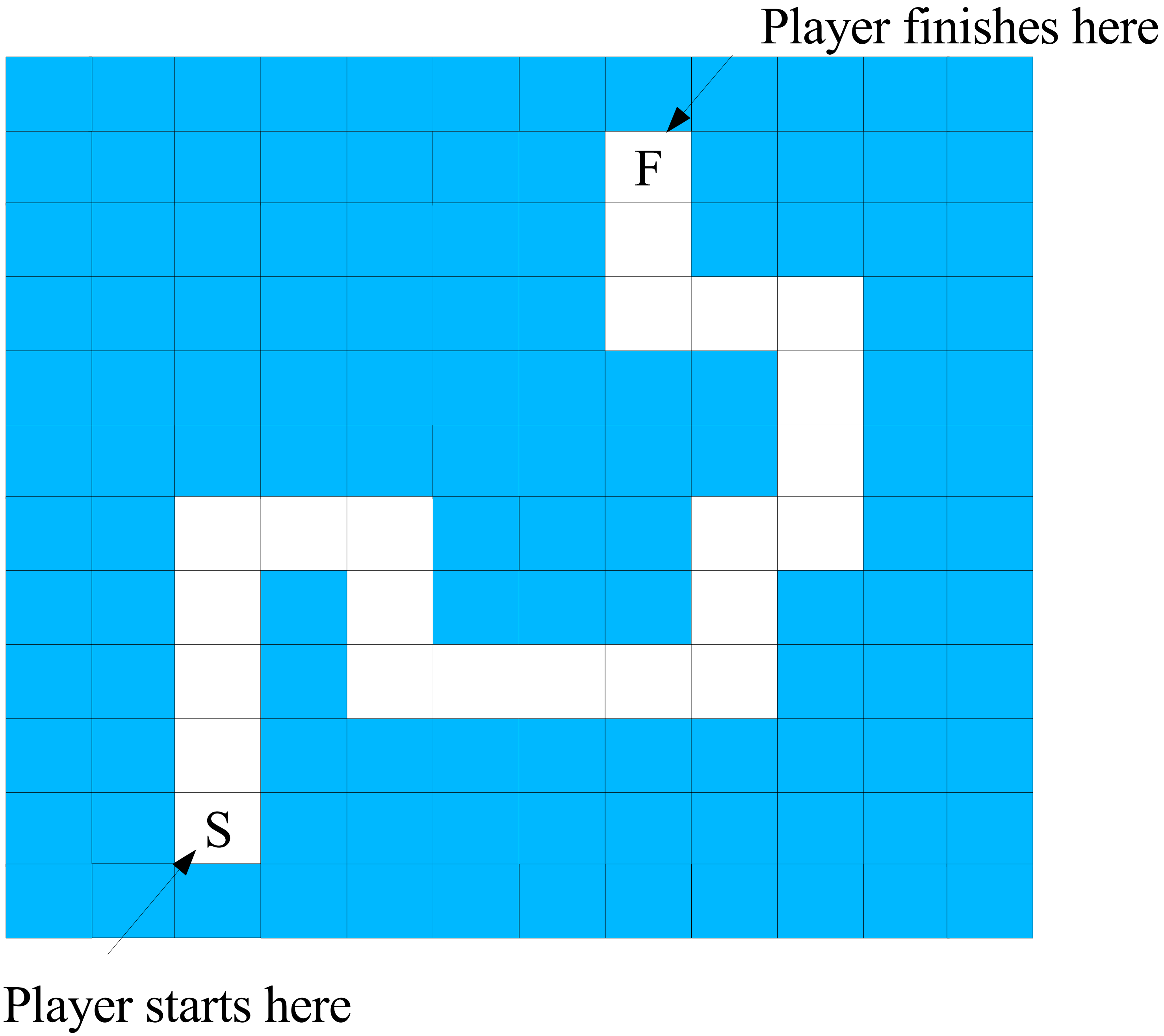
What does it mean to be OO?

1) Encapsulated
2) State Retention
3) Implementation / Information Hiding
4) Object Identity
5) Messages
6) Classes
7) Inheritance
8) Polymorphism
9) Generacity

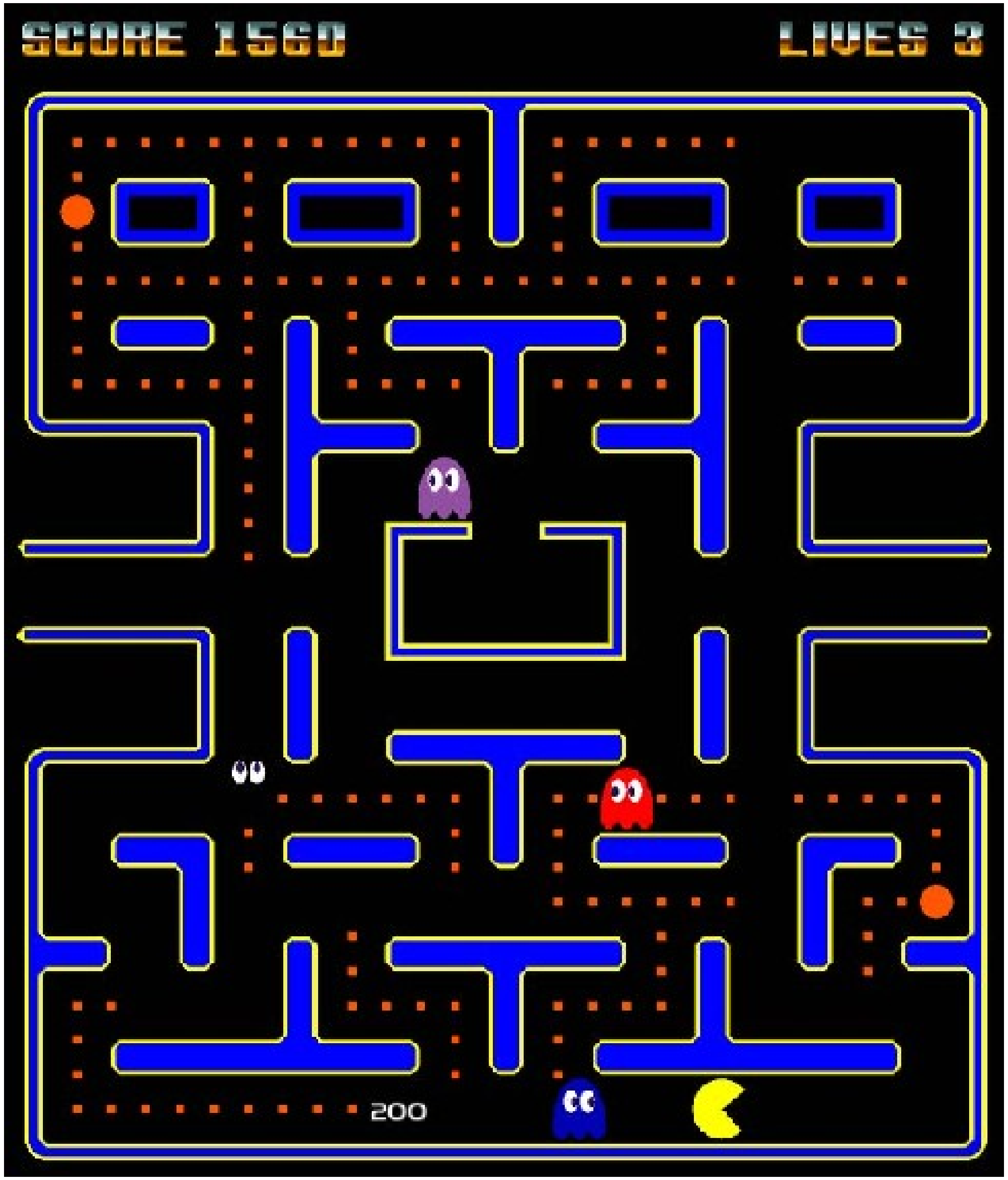
Object Structured, Based, Oriented

- If we exhibit 1 – 3, we are object – structured
- If we exhibit 1 – 4, we are object – based
- If we exhibit 1 – 7, we are class – based
- If we exhibit 1 – 9, we are object – oriented

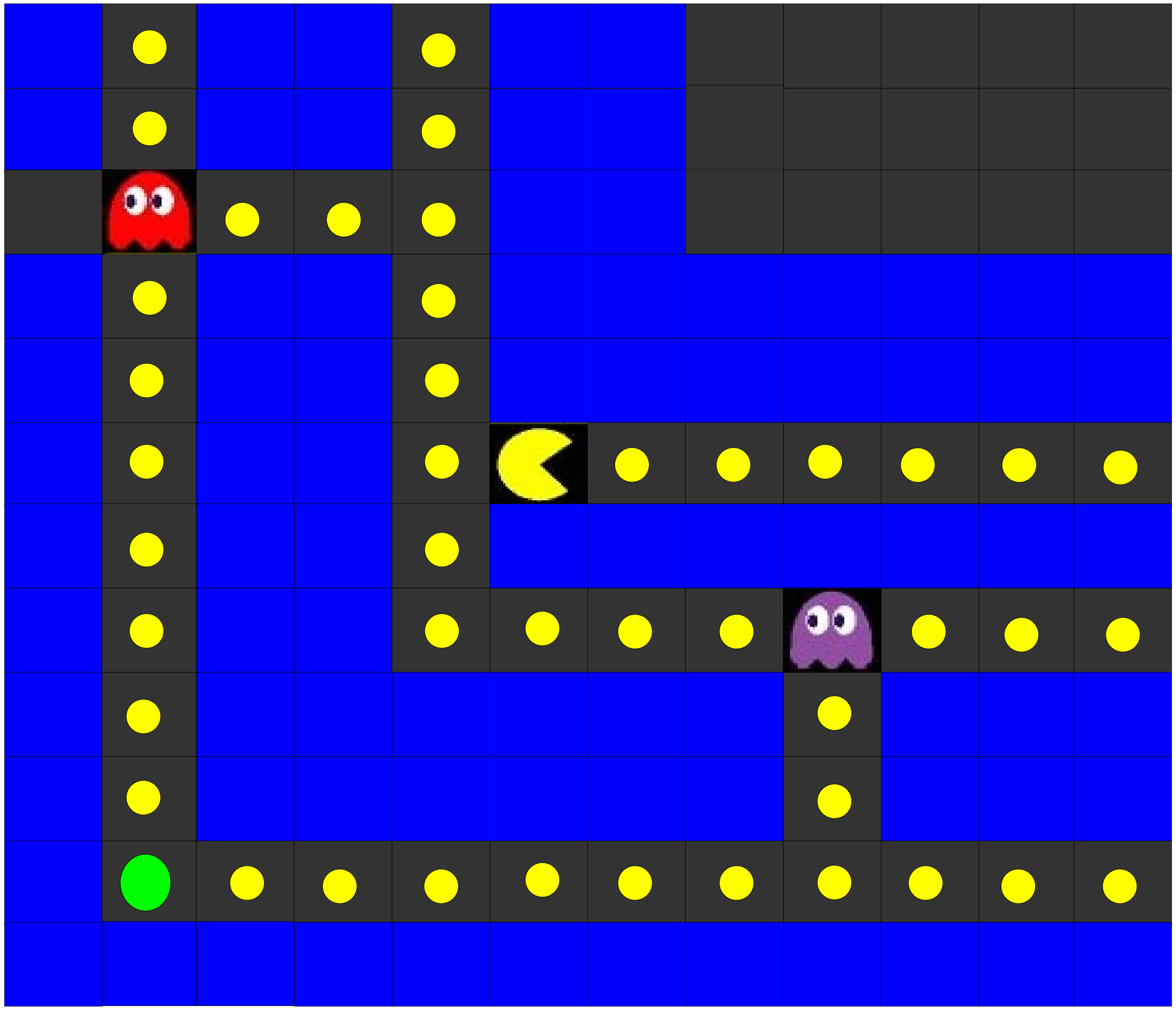
Case Study, Hominoid



Case Study, Pacman



PacMan Grid



Lets design

- What class do we need?
- What attributes should they have?
- What methods should they have?

Encapsulation

- Grouping of related ideas into a single unit referred to by a single name
- Different levels of encapsulation:
  - Level 0 : line of code
  - Level 1 : lines of code, procedural
  - Level 2 : set of procedures, class
  - Level 3 : set of classes of the same domain,
    - Horizontal, package like

Or

  - Level 3 : set of classes of different domains performing a common job
    - vertical, component like

So what is OO encapsulation?

- Object – Oriented (referred to as OO hereafter) encapsulation is the grouping of methods and attributes representing state, into an object so that the state is accessible and modifiable via the interface provided by the encapsulation

Encapsulation in PacMan

- Level 1 : Lines of code -> Actions
  - Up, Down, Left, Right
  - Add player to maze
- Level 2 : Actions + State -> Key objects :
  - Player
  - Game Area
  - Etc
- Level 3 : Key objects -> (mini?)Game
- Level 4 ?

State retention

- The attributes of an object represents what is remembers.
- The state of an object is the set of current values for those attributes.
- If an attributes changes, so does the state of the object.
  - An object composed of 4 booleans has 16 possible states.
  - An object composed of 2 integers has 18 446 744 073 709 551 616 possible states.
- State of an object may differ before and after a method call.
  - objects don't die after “execution”.

State in Pacman

- How many states can a player have?
- A player has the following attributes:
  - Location : Square
  - Direction : Cardinal Direction

PacMan

- How do you store the direction the player is facing?
  - North, South, Est, West

Info. / Implementation hiding

- When observing an encapsulation, we can have two point of view:
  - From the outside (public view)
  - From the inside (private view)
- The advantages of a good encapsulation is the separation of the private and public views.
- To access elements in the private view, users must go through the public interface.
  - Use of encapsulation to restrict internal workings of software from external user view

Pacman : Player

- How do I store the direction a player is facing?
  - An integer ?
    - 4 possble values : 1=North, etc
    - Values from 0 to 99.9 ?
    - Values from 0 to 360 ?
  - A character ? n,s,e and w
  - 4 booleans ? north, south ?
- How do I hide this from the user?
  - IsFacingNorth() : boolean
  - IsFacingSouth() : boolean
  - IsFacingEst() : boolean
  - IsFacingWest() : boolean