

# cs121 - software development UML

alexandre r.j. françois

visiting associate professor of computer science



# outline

what is uml?

goals

short history

diagram types

use case diagram

state diagram

activity diagram

# what is uml?

uml = unified modeling language

a language for specifying, visualizing, constructing, and documenting the artifacts of software systems

standard sanctioned by the object management group

*[www.omg.org](http://www.omg.org)*

initial 1997, revised 2003

# goals of uml

- Provide users with a ready-to-use, expressive visual modeling language
- Provide extensibility and specialization mechanisms to extend the core concepts
- Be independent of particular programming languages and development processes
- Provide a formal basis for understanding the modeling language
- Encourage the growth of the OO tools market
- Support higher-level development concepts such as collaborations, frameworks, patterns and components
- Integrate best practices

# short history of uml

grady booch, jim rumbaugh and ivar jacobson  
then at *rational software*  
object oriented software engineering

standardization via OMG  
with input from many...

# uml diagram types

use case diagram

state diagram

activity diagram

class diagram

interaction diagrams (objects)

sequence diagram

collaboration diagram

physical diagrams

component diagram

deployment diagram

# use case diagram

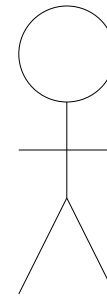
displays the relationship among actors and use cases  
(static)

use to gather and explore requirements for purpose of  
design or testing

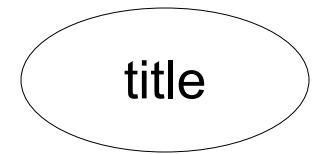
involves actors and use cases  
relationships

actor / use case

use case / use case: extends, includes, generalizes

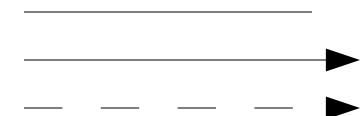


actor

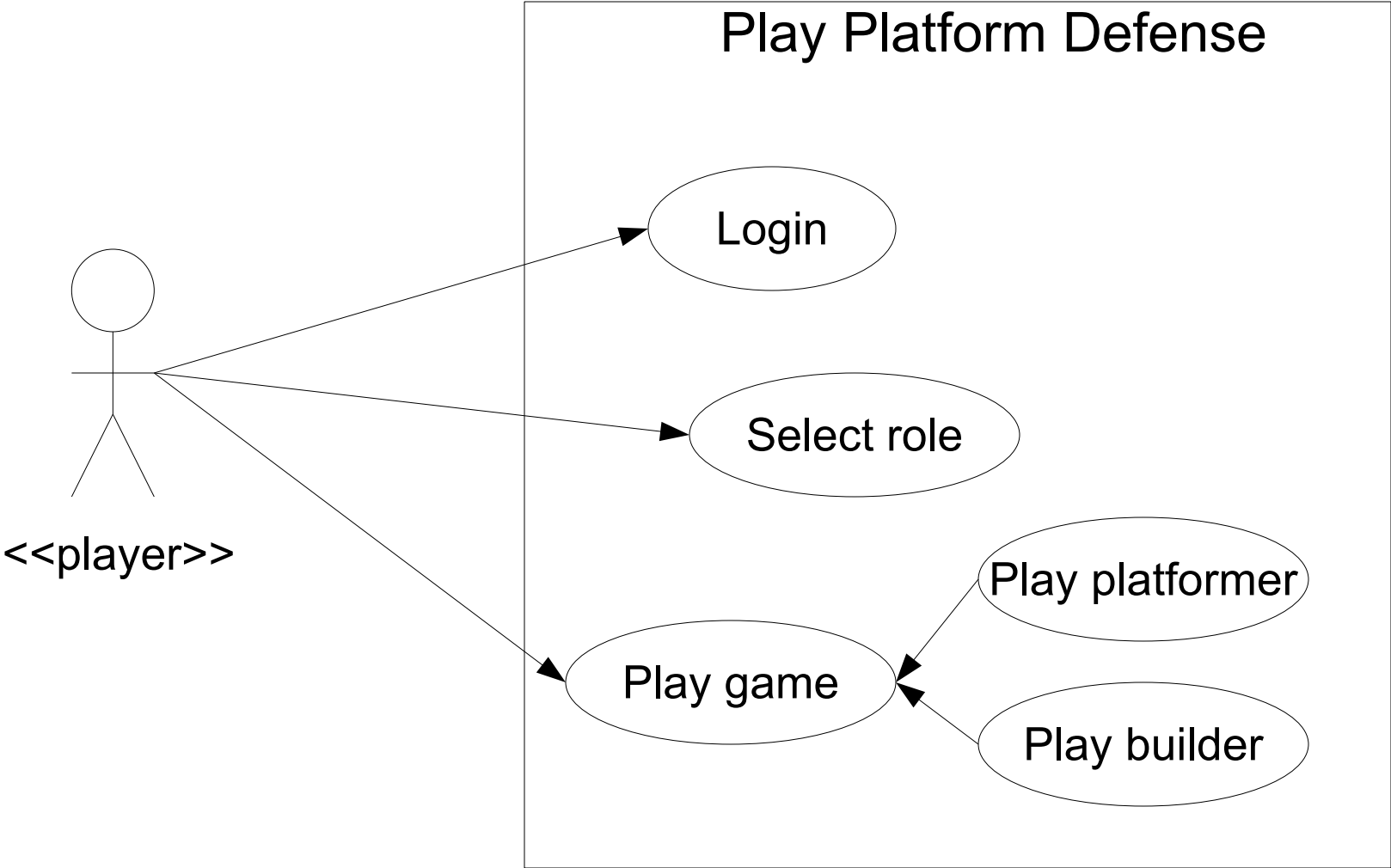


use case

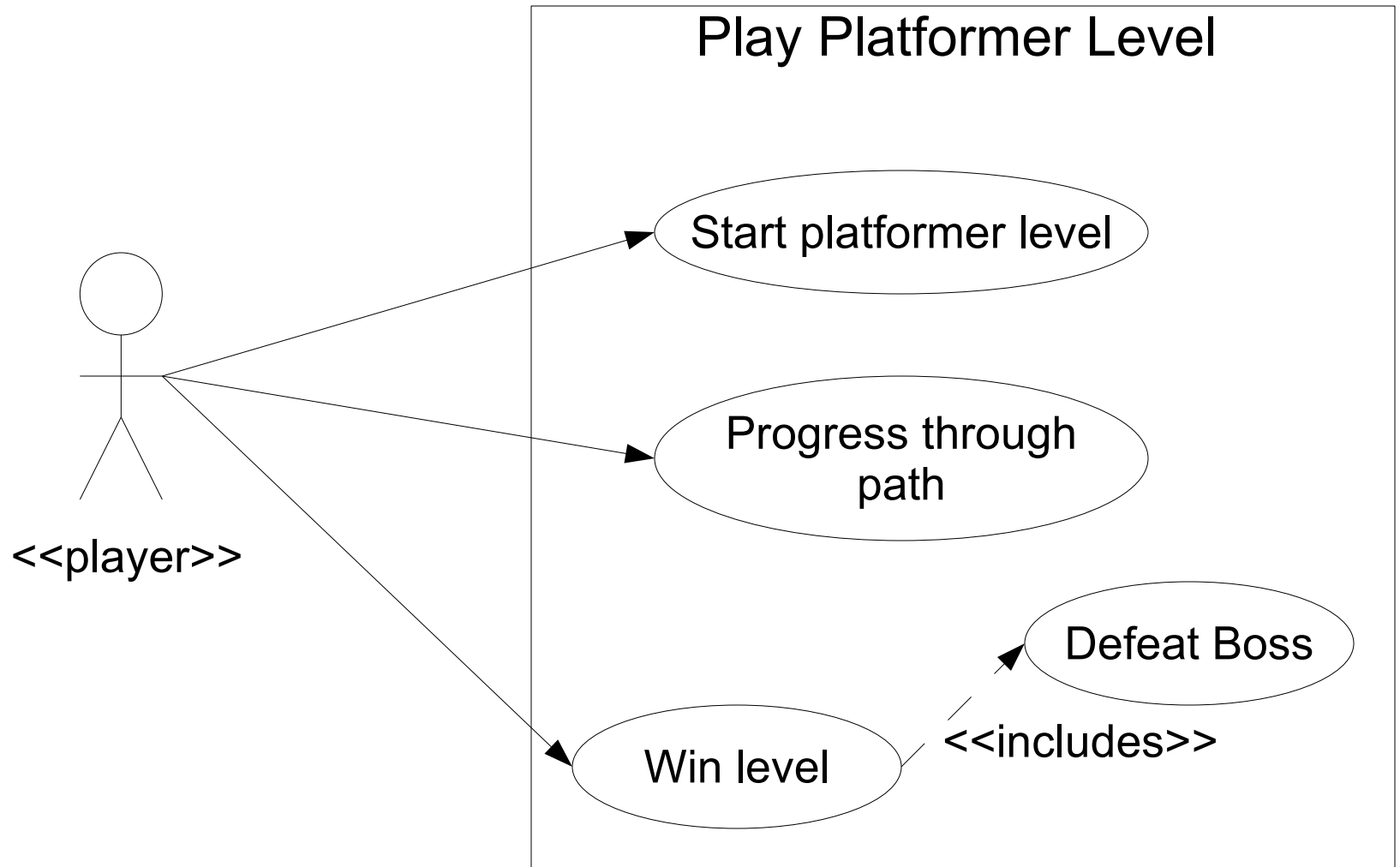
system



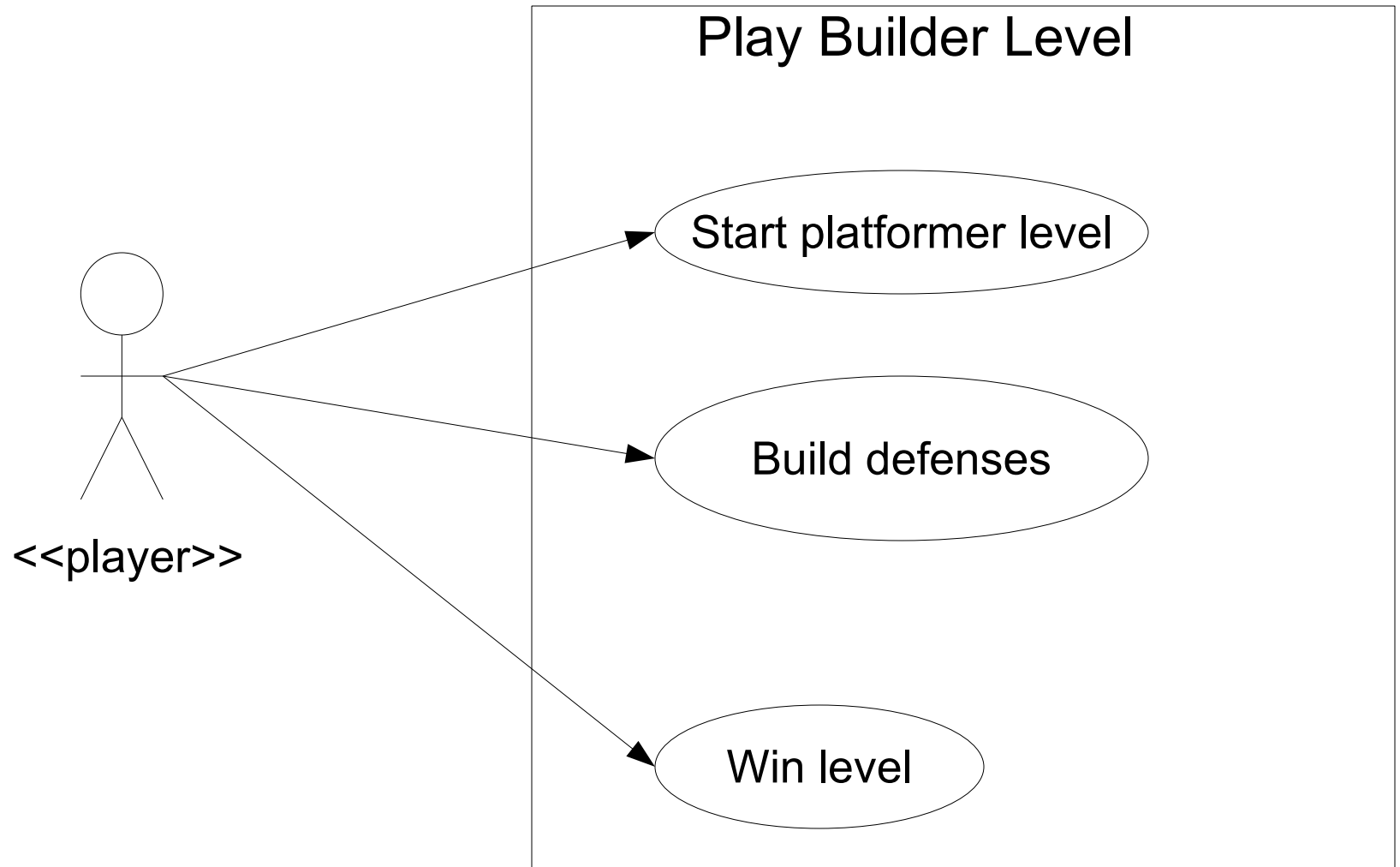
# use case diagram: example1



# use case diagram: example2



# use case diagram: example3



# state diagram

displays the sequences of states that an object goes through during its life in response to received stimuli, together with its responses and actions

(dynamic)

start / end states



transition

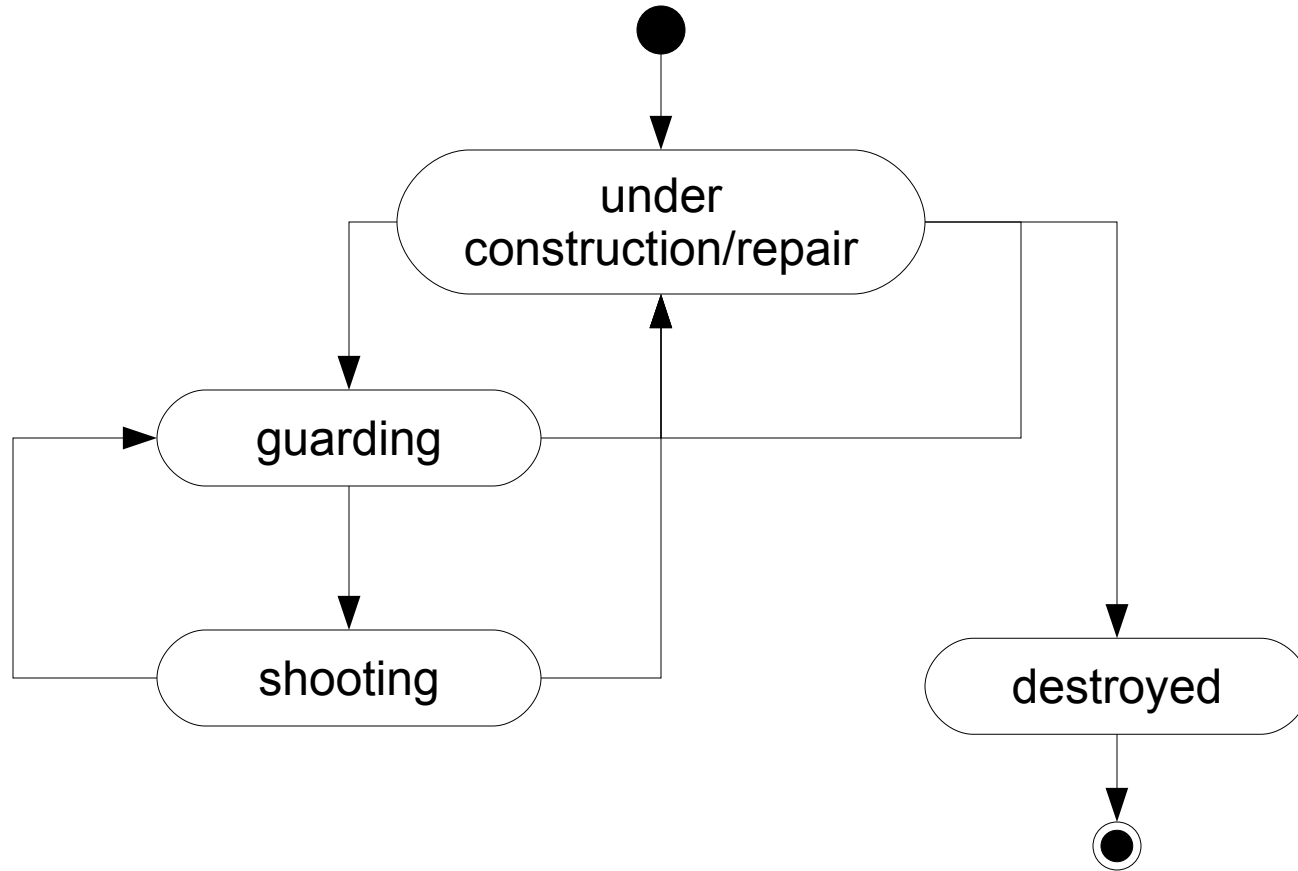


+ event / message

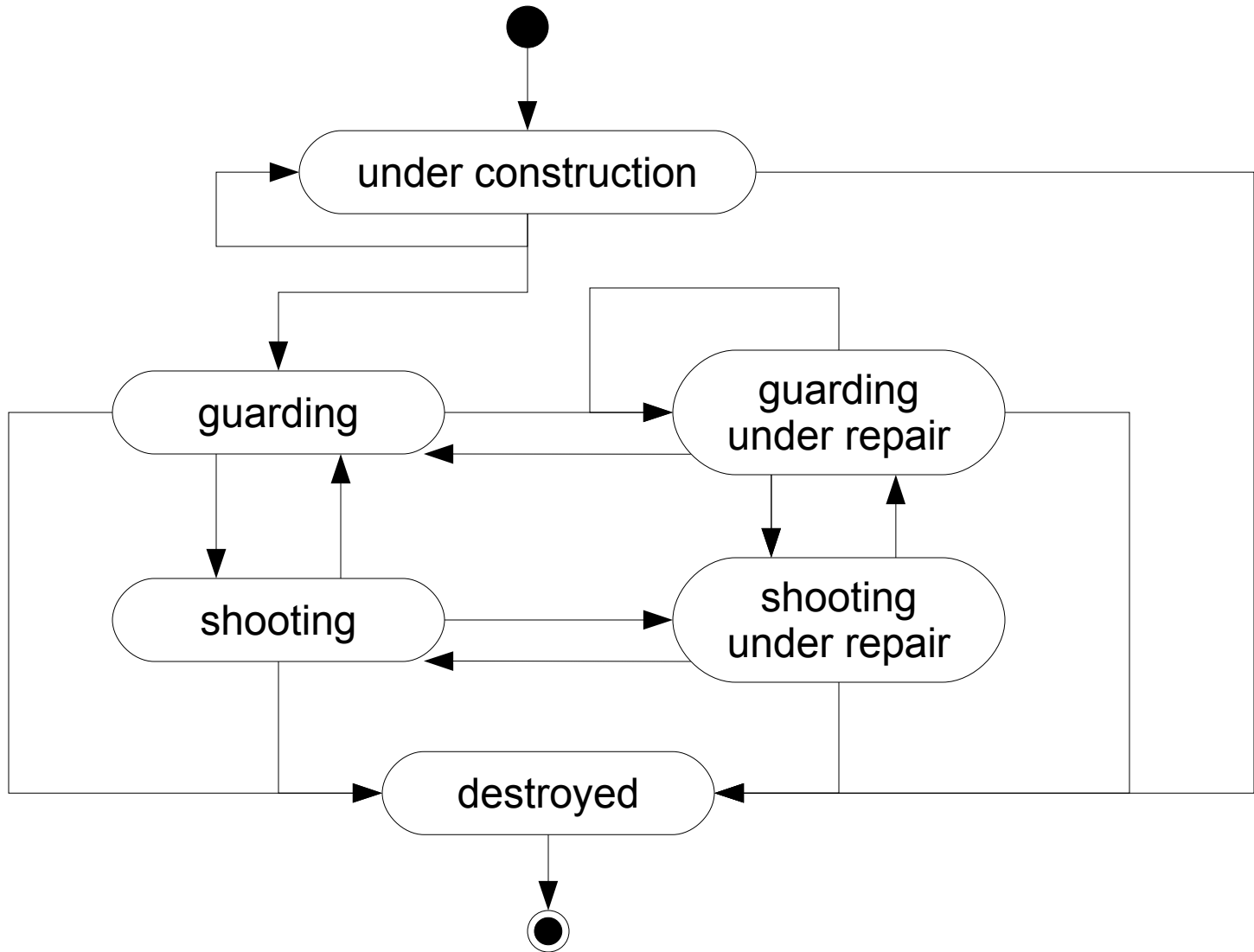
state



# state diagram: example1 tower



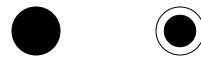
# state diagram: example2 tower



# activity diagram

special state diagram where most of the states are  
action states and most of the transitions are triggered  
by completion of the actions in the source states  
focuses on flows driven by internal processing (dynamic)

start / end of flow



transition



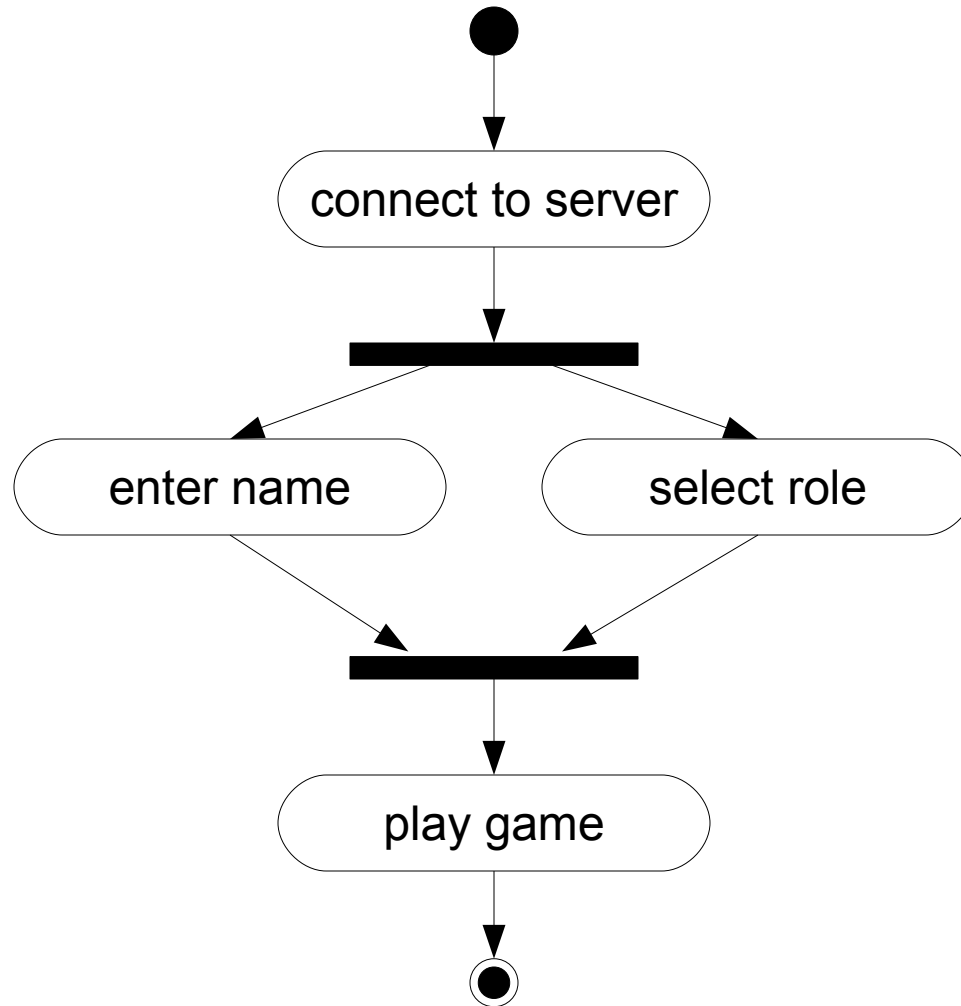
activity



synchronization



# activity diagram: example play platform defense



# summary

uml is a unified language

all that (so many diagram types!)

and only that...

language does not magically solve all problems!

use diagrams when they help in design and  
requirements modeling, code design, etc.

class diagram and interaction diagrams will be described  
when relevant...