

Modeling: Some Concrete Cases

By MiMaCh System – *Course of Intelligent Systems* – a.a. 2021/2022

Contents

- Introduction
- The world of **two coins**
- The world of a **robot**
- The world of **mean-max game**
- Conclusion

Introduction

What is Modeling?

- **Generally**, the process of representing a real-world object or phenomenon.
- The **representation**, often mathematical, of a process, concept, or operation of a system, often implemented by a computer program.

Why is it important?

- It is important to **model a problem** or the input of a problem in order to find an easy way to **understand** and **solve** it.
- Model the whole universe to understand it.
- Basic of **ontological foundations** of cognitive and communicative processes, **knowledge engineering**, conceptual modeling and software engineering.

The world of two coins

Description:

- Two coin with two sides
 - $C1 = \{T\} \{H\}$
 - $C2 = \{T\} \{H\}$
- Toss both the coins one or more time
- Win 10\$ each turn if the following purposes occur.

Purposes:

- Never two tails
- At least one tail
- Consider both the purposes (in conjunction)

Formal definition:

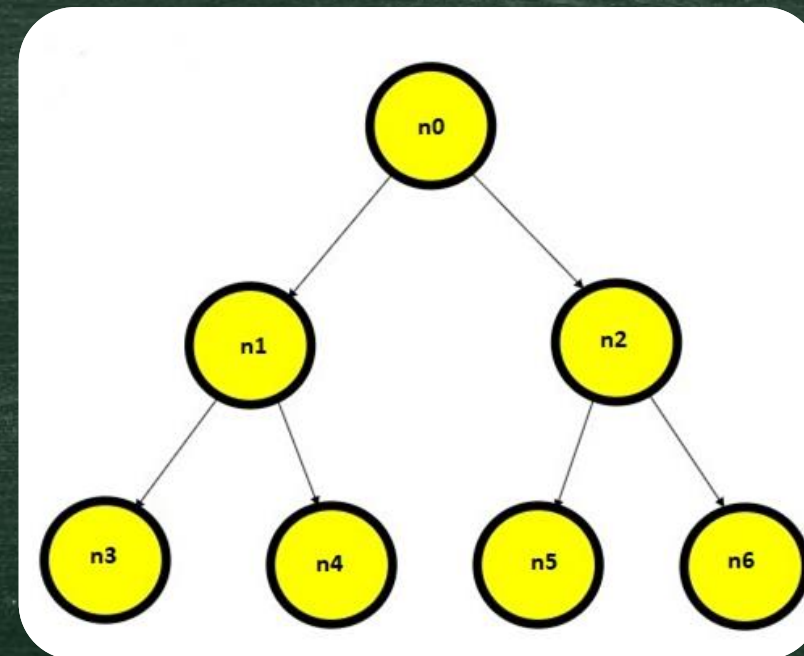
- The set of all the possible outcomes without $\{T, T\}$
- The set of all the possible outcomes with at least a T
- Conjunction of the first two

The world of two coins

Structure and Modelization:

- The binary tree explores every possible outcomes:
 - **n0** = no coins tossed
 - **n1** = head for the first coin
 - **n2** = tail for the first coin
 - **n3** = head after head
 - **n4** = tail after head
 - **n5** = head after tail
 - **n6** = tail after tail

Model the outcomes as a Tree

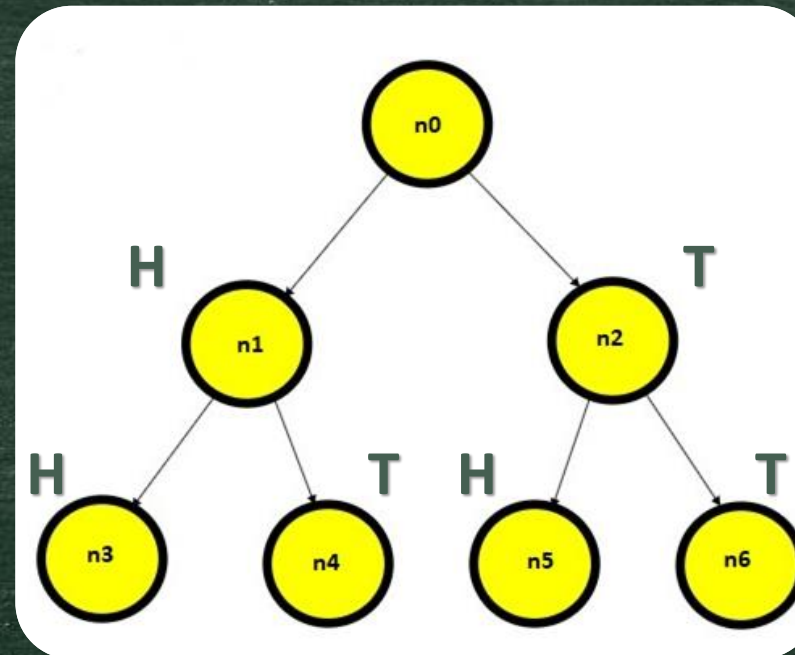


The world of two coins

Structure and Modelization:

- Focus on the first purpose:
 - Never two tail
 - $n3=\{H,H\}$; $n4=\{H,T\}$; $n5=\{T,H\}$;
- Focus on the second purpose:
 - At least one tail
 - $n4=\{H,T\}$; $n5=\{T,H\}$; $n6=\{T,T\}$;
- Focus on the third purpose:
 - First purpose AND second purpose
 - $n4=\{H,T\}$; $n5=\{T,H\}$;

Model the outcomes as a Tree



$\Phi: n4 \wedge n5$

T1: $n4$

T2: $n5$

$n4$ and $n5$ are models

The world of a robot

Description:

- There is a robot in a **room**
- The robot wants to exit **avoiding obstacles**

Purpose:

- Reach the **exit**

Formal definition:

- Find a **«safe» path** from the initial position of the robot to the position of the exit.

The world of a robot

- The map is a **2-D array** of size **NxM**. In this case there are 4 rows and 5 columns.
- Each object is in a cell with **position (i, j)**.
- Three kind of objects:
 - The **Robot** (Dynamic)
 - k **Bombs** (Static)
 - 1 single **Exit** (Static)
- Given the objects and an **ID** that identifies them in the **matrix**, knowing their position, it will be easy to **construct a path** using an algorithm that allows the robot to reach the exit cell without die by reaching the cell with bombs.

0,0 	0,1	0,2	0,3	0,4 
1,0	1,1 	1,2	1,3	1,4
2,0	2,1	2,2	2,3 	2,4
3,0	3,1	3,2	3,3	3,4 

The world of «Mean-Max» Game

▪ Description:



The map:

- Model the “mean-max” game as a **circular map** in a Cartesian plane, with the center equal to $C(0,0)$ and **radius $r=6000$** .
- Water Town is the core of the game and it is in the center of the map and has a radius of 3000.

The Entities:

- Map **each object** in the map as an Entity.
- There are different types of players and autonomous objects:

- **Reapers**
- **Destroyers**
- **Doofs**
- **Tankers**
- **Wrecks**

Every Entity has some **attributes** and **properties**

The world of «Mean-Max» Game

The Moves:

As **players** we can:

- **Move** to a position (X, Y) with a THROTTLE ≥ 0 and ≤ 300
- Use a **SKILL** in position (X, Y)
 - *Reaper's Skill* = Tar Pool
 - *Destroyer's Skill* = Nitro Grenade
 - *Doof's Skill* = Oil Pool
- **WAIT**






Purpose:

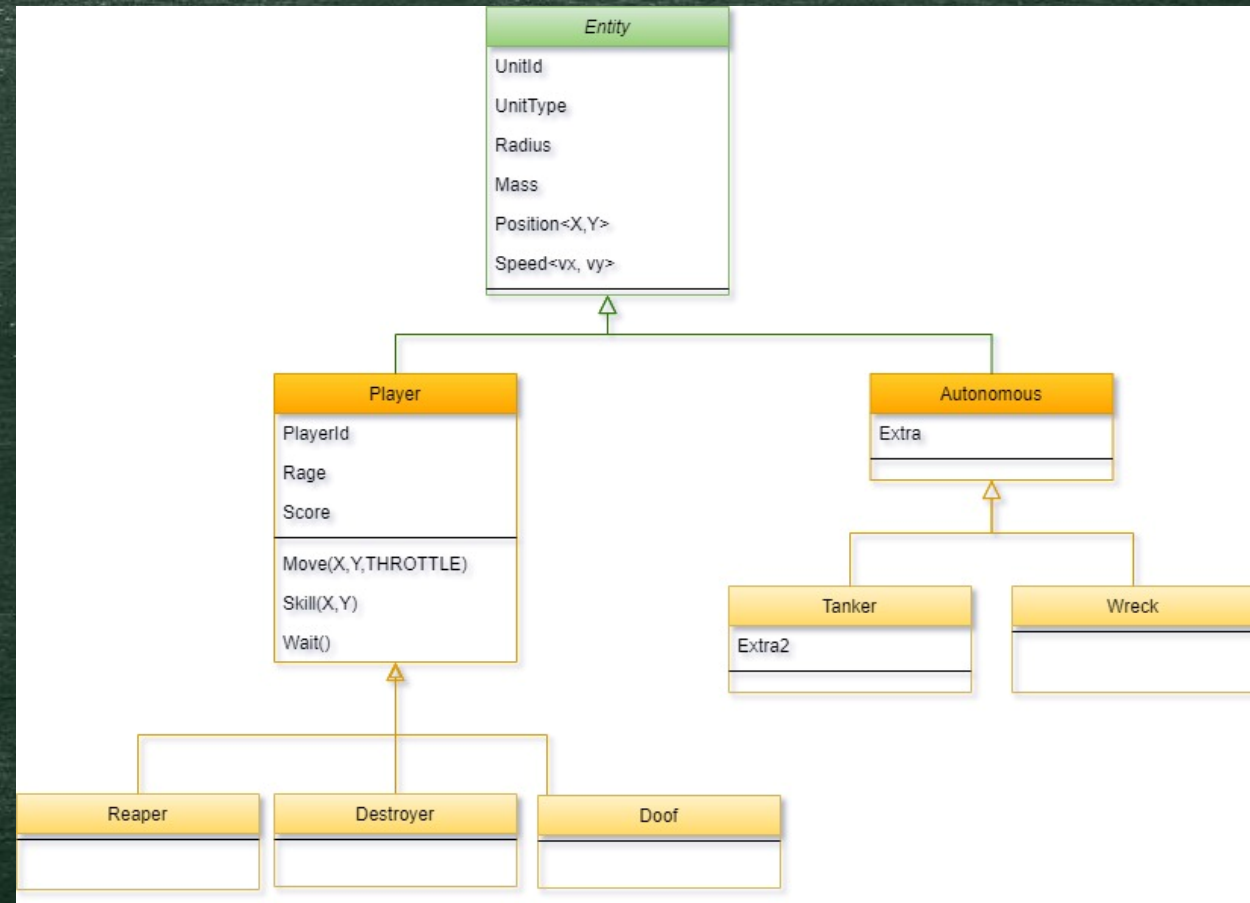
- Win the **play** by gathering more water than the other players.

Formal definition:

- Have an higher «**score**» than the other reapers.

The world of «Mean-Max» Game – Class Diagram

		Name	Type
Looters = Player controlled		Reaper	0
		Destroyer	1
		Doof	2
Autonomous		Tanker	3
		Wreck	4



Resources

- Text Book (Mainly Chapter 10):
[Intelligenza Artificiale. Un Approccio Moderno. Stuart J Russell, Peter Norvig. Pearson, 2021.](#)
- Modeling definitions:
[dictionary.com](#)
- Mean-Max Game:
[codingame.com](#)
- Class Diagram UML:
[visual-paradigm.com](#)

THANKS FOR YOUR ATTENTION

MiMaCh System

- Canonaco Martina [231874]
- Morello Michele [223953]
- Passarelli Chiara [223971]