World of Robots

Master 1 Génie Physiologique et Informatique

Del 4.1 : Basic structure of your classes

Group 14 : Laurène DONG, Marion DOSPITAL, Lauranne LAIR, Thibault NIGGEL, Florian PAILLAS, Ophélie SOCHARD

Robot

- name : String
- xPosition: int
- yPosition: int
- colourBody: String
- numberOfUnnamedRobots: int

- colourBody: String
+ draw(): void
+ getName(): String
+ getNumberOfUnnamedRobots(): int
+ setName(String): void
+ getXPosition(): int
+ getYPosition(): int
+ setXPosition(int): void
+ setYPosition(int): void
+ setColourBody(String) : void
+ getColourBody(): String
+ move(): void

World

- width: int
- height: int
- robots: ArrayList<robot>

+ World (int, int)

Use

Use

CanvasRobot

- canvas: Canvas
- randomGenerator: Random
- redrawable: boolean
- x,y,n,lo, la, plo, pla, tlo, tla, qla, milieu: int
- colourHead: Colour
- colourBoby: Colour
- colourLeg: Colour
- colourEye: Colour
- corps, brasG, brasD, jambeG, jambeD, tete, oeilD, oeilG: Integer

---

+ CanvasRobot(String)
+ drawRobot(int, int): void
+ setColourBody(String): void
+ getColourBody(): String
+ valueOf(String): Colour

Canvas

- canvasSingleton : Canvas
- frame: JFrame
- canvas: CanvasPane
- graphic Graphics2D
- backgroundColor: Color
- canvasImage: Image
- objects: List<Object>
- shapes: HashMap<Object, ShapeDescription>
- shape: Shape
- colour: Colour

+ getCanvas ()
+ Canvas(String, int, int, Color)
+ setVisible(boolean): void
+ draw(Object, Colour, Shape): void
+ erase(Object): void
+ setForegroundColour(Colour): void
+ wait(int): void
+ redraw(): void
+ erase(): void
+ paint(Graphics): void
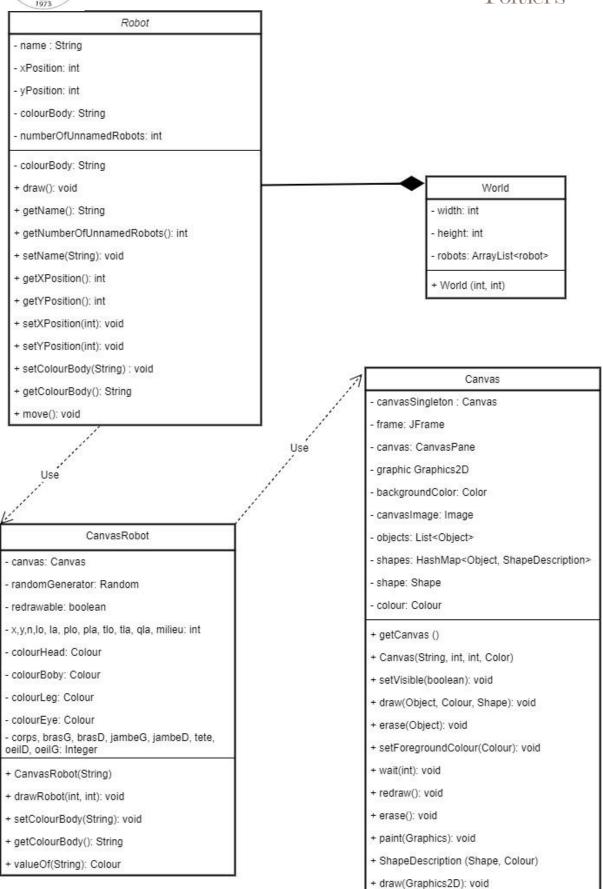+ ShapeDescription (Shape, Colour)
+ draw(Graphics2D): void

Figure 1: Basic structure of Robot and World classes

The relationship between the Robot class and the World class is a composition because the Robot object can't exist without the world and cannot evolve.

The attributes and methods of the robots allow them to be positioned in the world freely.

If a robot needs to evolve in a world, we need to define a class for robot and world. The class 'robot' defines the characteristics of the robot, specifically his position in space. These data allow the robot to position himself, to move around for example. For this purpose, we need a class 'world', to set the size of the world where the robot exists. The class 'robot' uses the class 'CanvasRobot' to visualize the robot, this class himself uses the class 'Canvas' to get a display window where the robots can be seen.