Methodes	Contraintes	Action	Rendu	Classe	Validation
fermer(boolean t)	Matériel fragile, le contact avec un palet peut parfoit décaler la pince	fermer les pinces au maximum, fermer (false)	ferme correctement les pinces	Pinces	
ouvrir(boolean t)		ouvrir les pinces ouvrir(false)	ouvre correctement les pinces		
rotate(double angle,boolean immediateReturn) - boussole initialiser à 0 par defaut	Imprécision des rotates	tourner d'un angle de 0, rotate(0,false)	le robot n'applique pas de rotation	WheelMotor	
		tourner d'un angle de 33, rotate(33,false)	le robot tourne de 33° vers la gauche		
		tourner d'un angle de -33, boussole == 0, rotate(-33, false)	le robot tourne de 33° vers la droite		
rotate(double angle)		tourne de 33°, rotate(33,false)	le robot tourne vers la droite de 33°		
getBoussole()		System.out.print(getBoussole())	retourne la valeur de la boussole		
forward(double distance,boolean immediateReturn)	Variation en fonction du terrain, le robot n'avance pas parfaitement droit	Avancer d'une distance spécifique forward (300,false)	Avance de 300 millimètres		
void backward(double distance)	Variation en fonction du terrain, le robot ne recule pas parfaitement droit	Reculer d'une distance spécifique backward(300,false)	Recule de 300 millimètres		
void boussole_a_0()	Incertitude au niveau des capteurs	réalise un scan face à un mur pour recalibrer la boussole	Le robot est perpendiculaire au mur		
forward() + motor.getMovement(). getDistanceTraveled()		avancer + stocker distance parcouru	avance et renvoie la valeur parcouru		
rotate(double angle,boolean immediateReturn)		rotate(x,true), doit tourner en donnant la main	ne tourne pas		
rotate(double angle,boolean immediateReturn) - boussole initialiser à 0 par defaut	Imprécision des rotates + en cas de choc avec un robot on va dérégler notre boussole	rotate + getBoussole() - boussole<0 && angle>0	la valeur de la boussole vaut la valeur de l'angle + le rotate est correctement réalisé		
		rotate + getBoussole() - boussole>0 && angle>0 et boussole + angle < 180			
		rotate + getBoussole() - boussole>0 && angle>0 et boussole + angle > 180			
		rotate + getBoussole() - boussole<0 && angle<0 et boussole+angle<-180			
		rotate + getBoussole() - boussole<0 && angle<0 et boussole+angle>-180			
		rotate + getBoussole() - boussole>0 && angle<0			
		rotate + getBoussole() - boussole>0 && angle<0			
rotateEnFonctionBoussole(double angleArrivee) - attribut boussoleB et angleB		angleArrivee == this.boussole	le robot ne bouge pas		
		Math.abs(angleArrivee) == Math.abs(this. boussole) && Math.abs(angleArrivee) == 180	le robot ne bouge pas		
		Si angleB > boussoleB && angleB - boussoleB > 180	rotate((angleB-360)-boussoleB)		
		angleB > boussoleB && angleB - boussoleB < 180	rotate(angleB - boussoleB)		
		Si angleB < boussoleB && boussoleB - angleB < 180	rotate(angleB - boussoleB)		
		Si angleB < boussoleB && boussoleB - angleB > 180	rotate((360-boussoleB)+angleB)		
forward() + motor.getMovement(). getDistanceTraveled()		avancer + stocker distance parcouru	avance et renvoie la valeur parcouru		
goTo(largeur,longueur) + getLargeur + getLongueur	Un choc avec un robot dérègle notre	largeur finale < largeur initiale && longueur finale < longueur initiale	se retrouve au bon endroit et renvoi les bonnes longueur et largeur		
		largeur finale > largeur initiale && longueur finale < longueur initiale	se retrouve au bon endroit et renvoi les bonnes longueur et largeur		

		goTo(largeur,longueur) + getLargeur + getLongueur	longueur et notre largeur	largeur finale < largeur initiale && longueur finale > longueur initiale	se retrouve au bon endroit et renvoi les bonnes longueur et largeur		
				largeur finale > largeur initiale && longueur finale > longueur initiale	se retrouve au bon endroit et renvoi les bonnes longueur et largeur		
		Color getColorOnGround()		Récupère la valeur perçus par le capteur de couleur	retourne un Color	Sensor	
		String Color_to_String(int r, int g, int b)	Capteur défaillant pour certaines couleurs	detecter couleur rouge	print "rouge" en passant sur la ligne		
				detecter couleur blanc	print "blanc" en passant sur la ligne		
				detecter couleur vert	print "vert" en passant sur la ligne		
				detecter couleur jaune	print "jaune" en passant sur la ligne		
				detecter couleur bleu	print "bleu" en passant sur la ligne		
				detecter couleur noir	print "noir" en passant sur la ligne		
		float getDistance()	Distance < 1 mètre sinon infinity (pas l'info de distance)	récupère la distance entre son capteur et le mur	renvoie une valeur correct		
		Color getColorOnGround()	Capteur défaillant	scan la couleur au sol	Renvoie le code RGB		
		String Color_to_String(int r, int g, int b)		recupère un code RGB	Renvoie le nom de la couleur		
		fermer() + rotate(double angle, boolean s)	rotation peu précise, présence d'un robot qui peut perturber la rotation	fermer + tourner, fermer(true)/rotate(45, false)	ferme et fait une rotation de 45° en même temps	Pince + WheelMotor	
		ouvrir() + rotate()	rotation peu précise, présence d'un robot qui peut perturber la rotation	ouvrir + tourner, ouvrir(true)/rotate(45, false)	ouvre et fait une rotation de 45° en même temps		
		fermer() + forward()	présence d'un robot qui peut perturber la mise a jour de la longueur et de la largeur	fermer + avancer	avance en fermant ses pinces		
		ouvrir() + forward()	présence d'un robot qui peut perturber la mise a jour de la longueur et de la largeur	ouvrir + avancer	avance en ouvrant ses pinces		
		rotate(double angle) + String Color_to_String(int r, int g, int b)	rotation peu précise, présence d'un robot qui peut perturber la rotation	tourner + capteur de couleur	tourne en captant la couleur au sol	Sensor + WheelMotor	
		rotate(double angle) + getDistance()	rotation peu précise, présence d'un robot qui peut perturber la rotation	tourner + capteur ultrason	tourne en faisant des print des couleurs		
		forward() + getDistance()	présence d'un robot qui peut perturber la mise a jour de la longueur et de la largeur	avancer + capteur ultrason	avance en faisant des prints des distances		
		forward() + havePalet()	présence d'un robot qui peut perturber la mise a jour de la longueur et de la largeur	avancer + capteur de pression	avance puis s'arrete lors d'une pression		
		forward() + String Color_to_String(int r, int g, int b)	présence d'un robot qui peut perturber la mise a jour de la longueur et de la largeur	avancer + capteur de couleur	avance en faisant des prints des couleurs		
		scan(double)	Si un mur est plus proche que tous les palets sur le terrain, le robot va se diriger dessus	scan(20), scan(60), scan(120), scan(180) avec un palet	se positionne vers le palet	FindPalet	
				scan(20), scan(60), scan(120), scan(180) avec plusieurs palets	se positionne vers le bon palet		
		marquerPalet()	Toutes les méthodes du code doivent fonctionner	l'objet capté est trop loin	le robot se repositionne bien et on stop la méthode		
				le robot n'est pas "bien" orienté vers le palet	on refait un scan de vérification		
				le robot a trop avancé par rapport à la distance captée	le robot a fermé les pinces et est revenu vers la ligne blanche et a ouvert les pinces		
				le robot a capté un mur ou un robot en avançant vers l'objet capté	le robot a reculé et a refait un scan		
				le robot a un palet dans ses pinces	le robot est revenu vers la ligne blanche et a ouvert les pinces		
		scanDone() + goToScanPoint()		on a appelé les 2 méthodes	le robot est bien au bon endroit de scan		