

Méthode :	Classe :	Action	retour	erreurs possible	validation	solutions	fonctionne
	MotorWheels					ajout d'un Coeff *1.5=pour les mouvement* ajoute d'un Coeff *1.3=pour les angles**	
angle()		calcul l'angle parcouru	double	imprécision de calcul		l'erreur diminue plus l'angle est grand	correction erreurs
backward(double)		recule d'une distance exprimé en double	void	différence entre la longueur rentrer et la longueur réel		*	
backward(double, boolean)		recule d'une distance exprimé en double en continuant à exécuter le programme	void	différence entre la longueur rentrer et la longueur réel et exécution du programme car il peut y avoir des contradiction avec un forward		*	fonctionne pas
boussole_a_0()		permet de réorienter le robot face a la zone d'en but de l'adversaire	void	différence entre l'angle rentrer et l'angle réel		due réglé le coefficient car le robot avec le palet ne tournait pas assez (multiplication par 12% l'angle ) le problème a été réglé	
forward()		avancer jusqu'a l'appelle de stop()	void	faire attention a stopper le motor avant l'appel d'une autre méthode qui utilise les motor		faire attention de bien appeler la méthode stop() après l'utilisation	
forward(double)		avancer d'une distance exprimé en double	void	différence entre la longueur rentrer et la longueur réel		*	
forward(double, boolean)		recule d'une distance exprimé en double en continuant à exécuter le programme	void	différence entre la longueur rentrer et la longueur réel et exécution du programme car il peut y avoir des contradiction avec un forward		*	
goTo(double, double)		amène le robot a un point donné sur le plan de jeu	void	imprécision de calcul et prendre en compte les variations de longueur , largeur et boussole		réduction des erreurs après avoir marqué	
rotate(double)		tourne d'un angle donné exprimé en double	void	différence entre l'angle rentrer et l'angle réel		**	
rotate(double, boolean)		tourne d'un angle donné exprimé en double en continuant à exécuter le programme	void	différence entre l'angle rentrer et l'angle réel		**	
rotateneg()		fait 1 tour sur lui-même par la gauche	void	différence entre l'angle rentrer et l'angle réel		*	
	Pinces		void				
fermer(boolean)		ferme les pinces	void	fermer es pinces alors qu'elle est fermé			
ouvrir(boolean)		ouvre les pinces	void	ouvrir les pinces alors qu'elle sont déjà ouverte			
reglagepinces(Int)		règle les pinces d'un angle exprimé en Int	void	/			
	Robot		void				
alleraupalet(Int)		appelle goTo en fonction du tableau de palet	void	imprécision de calcul et prendre en compte les variations de longueur , largeur et boussole		utilise goTo donc même résolution	
AllerAZone(Int)		appelle goTo en fonction du tableau de researchArea	void	imprécision de calcul et prendre en compte les variations de longueur , largeur et boussole		utilise goTo donc même résolution	
aunpalet()		permet de savoir si l'élément devant le robot est bien un palet	boolean	faire attention au faux positif et au faux négatif		ajustement de la longueur capté	
allerjusqua(String)		va jusqu'a la ligne blanche	void	faire attention si pendant le trajet il y a un obstacle		ajout d'une méthode pour éviter un obstacle	

catchTarget(float)		permet d'attraper un palet	void	faire attention si pendant le trajet il y a un obstacle ,et erreur sur les distance en avançant		ajout d'une méthode pour éviter un obstacle
distance()		retourne la distance	double	différence entre la distance capté et la distance réel		
erreurs_boussole()		permet de corriger l'erreur après un but	void	/		
getColor()		capte la couleur dessous la capteur couleur	void	le capteur ne capte pas la ligne		
goal(boolean)		marque un but(true fait erreurs_boussole())	void	/		ajout de la méthode erreurs_boussole() qui permet de réduire les imprécisions sur l'orientation ,corrige après un but
research()		permet de faire une recherche autour du robot	void	l'objet capté est ou non un palet		ajout d'un contrôle de mise a jour, cela permet de réduire l'erreur, calcul du temps entre le démarrage et la réussite de la trouver un palet le problème a été réglé
	UltrasonicSensor		void			
objectDetectedInRange(Int)		Détecte si un objet se situe à une distance inférieur a la range(en mètre) du capteur	void	différence entre la distance capté et la distance réel		pas besoin de correction, les résultats sont cohérents
	TouchSensor		void			
isPressed()		renvoie true si le capteur touch est enfoncé	boolean	/		pas utilisés
	ColorSensor		void			
color_String(Int, Int, Int)		retourne la couleur après 3 string qui sont RVB	void	différence entre la couleur capté et la couleur réel		aucune différence
getColor()		retourne la couleur capte la couleur dessous la capteur couleur de type couleur	void	différence entre la couleur capté et la couleur réel		aucune différence
getColorint()		retourne la couleur capte la couleur dessous la capteur couleur en tableau en Int	void	différence entre la couleur capté et la couleur réel		aucune différence