

The logo for COLOBOT is displayed in a stylized, metallic font. Each letter is composed of mechanical parts, including gears, bolts, and cylindrical components, giving it a robotic and industrial appearance. The letters are arranged horizontally and are set against a background of a control panel with various buttons, lights, and cables.

1 Introduction

COLOBOT est un mélange d'un jeu de stratégie en temps réel et d'une initiation à la programmation. Le scénario vous place à la tête d'une expédition spatiale, composée d'un seul humain (vous) et de quelques robots. Vous devrez explorer et coloniser différentes planètes, tout en cherchant des matières premières nécessaires à votre survie.

1.1 Histoire

La vie sur terre est menacée par un cataclysme dévastateur. Il faut la quitter.

Une première expédition composée uniquement de robots est partie pour trouver une autre planète colonisable. Elle a connu échecs sur échecs, pour des raisons encore obscures. Relevez le défi de réussir en tant qu'humain là où les robots ont échoué ...

Vous allez atterrir sur des planètes inconnues, avec quelques robots comme seuls compagnons. Vous devrez construire l'infrastructure nécessaire pour garantir l'approvisionnement en matières premières et en énergie.

Mais hélas, plusieurs planètes sont habitées par des insectes géants, qui n'apprécient pas du tout votre présence. Il faudra produire les moyens indispensables pour vous défendre, et combattre pour survivre.

Prenez les commandes d'un robot, et essayez d'être le plus rapide, de ne pas vous laisser surprendre. Mais parfois, c'est impossible. Les insectes sont trop nombreux, tout explose, et c'est la fin.

Le seul moyen consiste à déléguer certaines tâches à vos robots en les programmant. Ainsi, vous pourrez envoyer plusieurs robots programmés à l'assaut d'un essaim de guêpes géantes qui vous attaquent, et prendre les commandes d'un autre pour attraper celles qui passent au travers du filet.

Heureusement que Houston, le centre de contrôle sur terre, ne vous laisse pas tomber. Ses ingénieurs vous fournissent des programmes tout faits. Mais Houston est bien lointain, et les programmes sont parfois mal adaptés.

L'apprentissage du langage de programmation utilisé, proche dans la structure et la syntaxe de C et de Java™, sera de toute manière un atout dans votre carrière professionnelle, après votre retour dans la vie civile.

La tâche semble ardue. Pourtant, Houston a choisi de ne pas sélectionner un spécialiste en informatique, mais un homme d'action doué d'une bonne dose de bon sens, et surtout de beaucoup de courage. Ces qualités sont essentielles pour réussir dans les missions. En effet, votre formation se fait sur le tas ...

1.2 Configuration requise

- Processeur 300 MHz
- 64 Mb RAM
- Carte graphique 3D avec 16 Mb RAM
- Les cartes Voodoo ne sont pas supportées
- 100 Mb d'espace libre sur le disque dur
- Windows[®] 95, Windows[®] 98, Windows[®] ME ou Windows[®] 2000

1.3 Installation

- Insérez le CD-ROM COLOBOT dans votre lecteur
- Double-cliquez sur **Poste de travail**.
- Double-cliquez sur **(D:)** ou sur la lettre correspondant au lecteur de CD-ROM.
- Double-cliquez sur **install.exe**.
- Si ce n'est pas déjà fait, le programme vous demande d'installer DirectX[®] 8a.

Si vous avez déjà installé une version précédente de COLOBOT, une nouvelle installation dans le même dossier détruira toutes vos sauvegardes.

Par la suite, la présence du CD est nécessaire pour jouer.

1.4 Désinstallation

- Double-cliquez sur **Poste de travail**.
- Double-cliquez sur **Panneau de configuration**.
- Double-cliquez sur **Ajout/Suppression de programmes**.
- Double-cliquez sur **COLOBOT** dans la liste.

2 Menu principal



2.1 Missions

Les missions contiennent la partie aventure de COLOBOT. Elles sont au nombre de 36, réparties sur 9 planètes différentes. Elles doivent être faites dans l'ordre imposé.

2.2 Jeu libre

Le mode « jeu libre » vous permet d'agir librement sur une planète, sans but déterminé. Seuls les planètes déjà visitées et les recherches effectuées sont disponibles.

2.3 Programmation

Cette partie de COLOBOT permet d'apprendre à programmer, même si vous n'avez aucune notion dans ce domaine. Différents chapitres présentent des sujets de plus en plus compliqués. Vous pouvez effectuer n'importe quel exercice, bien qu'il soit conseillé de commencer par les plus simples.

2.4 Défis

Les défis vous demandent une bonne connaissance de la programmation. Ils permettent de vérifier si les notions apprises sous « programmation » sont effectivement assimilées.

2.5 Suppl.

Les missions supplémentaires peuvent être téléchargées sur le web, à l'adresse www.colobot.com. Cliquez ensuite sur « forum ».

2.6 Options

2.6.1 Affichage



Pilotes :

Il faut choisir un pilote qui porte la mention HAL (Hardware Abstraction Layer). Evitez les pilotes ayant la mention « Emulation » ou « T&L ». Il arrive fréquemment que des pilotes avec des noms différents aient des comportements identiques.

Résolutions :

Les premier et deuxième chiffres indiquent le nombre de pixels en largeur et en hauteur dans l'écran. Le troisième chiffre indique le nombre de couleurs affichables (16 pour 65'000 couleurs et 32 pour 4 millions de couleurs).

Plus la résolution est grande et plus le jeu est beau. Mais il risque de devenir lent. Commencez par mettre 640 x 480 x 16. La plupart des cartes graphiques modernes supportent 1024 x 768 x 16. A vous d'essayer le meilleur compromis.

✓ Plein écran

Normalement, COLOBOT occupe tout l'écran, quelle que soit la résolution. Si vous enlevez la coche, COLOBOT tournera dans une fenêtre fixe ayant approximativement 640 x 480 pixels.

[Appliquer les changements]

Il faut cliquer ce bouton pour que les changements effectués dans cette page prennent effet.

2.6.2 Graphique



✓ Ombres

Normalement, tous les objets (robots, bâtiments, titane, etc.) projettent une ombre sur le sol. Avec certaines cartes graphiques anciennes, un carré gris peut apparaître, en plus de l'ombre. Si cela vous semble inesthétique, supprimez cette option.

✓ Marques sur le sol

Les marques accentuent les berges et donnent des teintes différentes à certains endroits du terrain. Si cette option est déclenchée, il n'est pas possible de montrer les zones plates avec le cosmonaute (voir chapitre 4.1.1).

✓ Salissures

Les salissures donnent un effet vieux et sale aux robots et aux bâtiments.

✓ Ciel

Avec cette option, le ciel de certaines planètes contient des nuages poussés par le vent. Sans cela, le ciel est généralement rempli par un simple dégradé de couleurs.

✓ Rayons du soleil

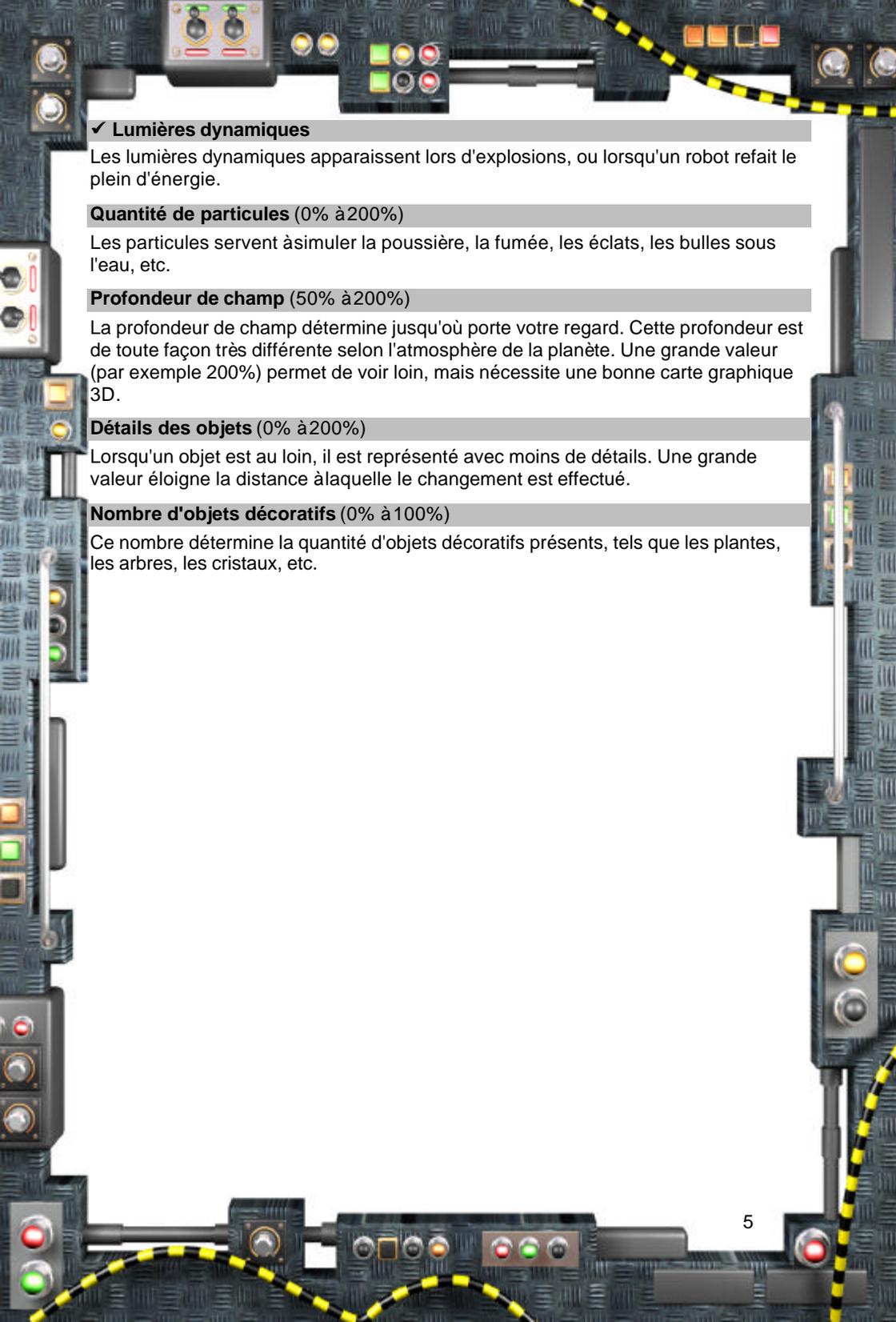
Lorsque vous vous tournez en direction du soleil, un reflet apparaît.

✓ Planètes et étoiles

Sur certaines planètes, vous voyez dans le ciel des planètes proches qui bougent lentement.

✓ Brouillard

Cette option correspond aux nappes de brouillard horizontales qui sont généralement très proches du sol.



✓ Lumières dynamiques

Les lumières dynamiques apparaissent lors d'explosions, ou lorsqu'un robot refait le plein d'énergie.

Quantité de particules (0% à 200%)

Les particules servent à simuler la poussière, la fumée, les éclats, les bulles sous l'eau, etc.

Profondeur de champ (50% à 200%)

La profondeur de champ détermine jusqu'où porte votre regard. Cette profondeur est de toute façon très différente selon l'atmosphère de la planète. Une grande valeur (par exemple 200%) permet de voir loin, mais nécessite une bonne carte graphique 3D.

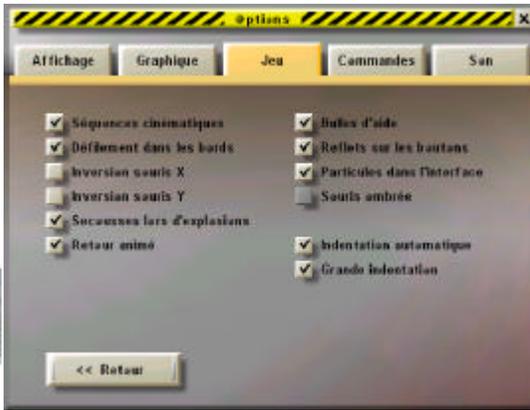
Détails des objets (0% à 200%)

Lorsqu'un objet est au loin, il est représenté avec moins de détails. Une grande valeur éloigne la distance à laquelle le changement est effectué.

Nombre d'objets décoratifs (0% à 100%)

Ce nombre détermine la quantité d'objets décoratifs présents, tels que les plantes, les arbres, les cristaux, etc.

2.6.3 Jeu



✓ Séquences cinématiques

Certaines missions commencent ou finissent par un petit film montrant l'atterrissage ou le décollage du vaisseau. La touche « Esc » permet toujours de stopper ces séquences. Si cette option est inactive, ces films sont sautés.

✓ Défilement dans les bords

Lorsque la caméra est derrière le cosmonaute ou derrière un robot, une rotation est effectuée si la souris s'approche du bord de l'écran.

✓ Inversion souris X

Inverse le sens de la rotation lorsque la souris touche le bord gauche ou le bord droite de l'écran.

✓ Inversion souris Y

Inverse le sens du mouvement lorsque la souris touche le bord supérieur ou le bord inférieur de l'écran, dans l'éditeur de programmes.

✓ Secousses lors d'explosions

Lors d'une explosion, ou lorsque vous effectuez un atterrissage brusque, la caméra subit un choc qui se manifeste par une secousse plus ou moins prononcée. Supprimez cette option pour que la caméra soit toujours parfaitement stable.

✓ Retour animé

Indique comment la situation est réinitialisée dans les exercices de programmation et dans les défis.

✓ Bulles d'aide

Les bulles d'aide vous donnent un petit texte explicatif lorsque la souris s'arrête sur un bouton ou un objet.

✓ Reflets sur les boutons

Les reflets sont visibles lorsque la souris survole un bouton.

✓ Particules dans l'interface

Pluie de particules lorsque la souris bouge dans les écrans d'interface.

✓ Souris ombrée

La souris ombrée est gérée par COLOBOT. La souris normale est dessinée par Windows®. Lorsque COLOBOT ne fonctionne pas en mode « plein écran », la souris est forcément normale.

✓ Indentation automatique

L'indentation déplace automatiquement le curseur vers la droite lors de l'édition d'un programme, en fonction des accolades { et }.

✓ Grande indentation

Une grande indentation décale vers la droite d'une distance égale à 4 espaces. Sinon, le décalage est de 2 espaces.

2.6.4 Commandes



Prenez l'habitude de jouer avec la main gauche sur les touches flèches du clavier, et la main droite sur la souris. Vous devez diriger le robot et actionner les commandes *monter* et *descendre* uniquement avec la main gauche. La main droite sur la souris permet de diriger le canon et de faire feu.

Flèches gauche, droite, haut et bas

Pour faire tourner, avancer ou reculer le cosmonaute ou un robot. Dans les exercices de programmation, notez que les robots ne peuvent pas être déplacés ainsi.

Shirt et Ctrl

Pour faire monter ou descendre le cosmonaute ou un robot volant. Sur certaines planètes, le vol est impossible.

Entrée

Cette touche effectue l'action principale du robot sélectionné, qui correspond au bouton avec un cadre rouge.

Espace

Change le point de vue de la caméra. Pour la plupart des robots, la caméra passe alternativement d'une vue arrière à une vue intérieure.

. (pavé numérique)

Met le jeu en pause et montre l'endroit correspondant au dernier message affiché en haut de l'écran. Si plusieurs messages sont affichés, une nouvelle pression montre le message précédent, chronologiquement parlant. « Esc » enlève la pause et le jeu continue.

Tab

Sélectionne l'objet suivant, selon l'ordre des petites icônes tout en haut de l'écran.

Home

Sélectionne toujours rapidement le cosmonaute.

0 (pavé numérique)

Sélectionne le robot ou le bâtiment qui était sélectionné précédemment.

+ et - (pavé numérique)

Approche ou éloigne la caméra de l'objet sélectionné.

F1

Affiche les instructions sur la mission ou l'exercice à l'aide du **SatCom**.

F2

Affiche le glossaire sur la programmation à l'aide du **SatCom**.

F3

Pendant l'édition d'un programme, cette touche affiche des informations complémentaires sur l'instruction en cours de frappe.

F4, F5 et F6

Choix de la vitesse du jeu. Le mode normal x1 correspond à la touche F4. Les modes rapides x1.5 et x2 doivent être utilisés avec prudence, car les ennemis vont également plus vite !

Esc

Quitte la mission en cours.

2.6.5 Son



Bruitages :

Les bruitages sont générés par l'action en cours. Il s'agit de bruits de moteur, de pas, d'explosions, etc.

Fond sonore :

Le fond sonore dépend de la planète. Il donne une ambiance générale, indépendamment de l'action en cours. Chaque fond sonore correspond à une piste audio sur le CD. Dans les exercices de programmation, les défis ainsi que sur la lune, il n'y a pas de fond sonore.

✓ Bruitages 3D

Certaines cartes son permettent de localiser un son dans l'espace à l'aide de 4 haut-parleurs. Le réalisme est alors superbe. Si votre carte son ne le permet pas, le bouton est grisé.

2.7 Joueur

Après avoir installé COLOBOT sur un ordinateur, plusieurs personnes peuvent y jouer. Pour chaque joueur, COLOBOT enregistre automatiquement la progression dans les missions, les programmes écrits, etc.

De plus, chaque joueur peut personnaliser l'aspect du cosmonaute dans le jeu :



Vous pouvez personnaliser les éléments suivants :

- Type du visage
- Lunettes
- Couleur des cheveux
- Couleur de la combinaison
- Couleur des 4 bandes de la combinaison, aux bras et aux jambes

Les curseurs permettent de choisir une couleur différente des 9 proposées, en spécifiant les composantes rouge, verte et bleue.

3 L'écran

Pendant une mission ou un exercice, l'écran ressemble à ceci :



- 1 Le résumé des robots ou des bâtiments existants
- 2 Les actions pour l'objet sélectionné
- 3 La mini-carte
- 4 L'accès au menu

3.1 Le résumé



La partie supérieure de l'écran résume les robots ou les bâtiments disponibles. L'objet actuellement sélectionné apparaît en orange. Lorsqu'un robot exécute un programme, le cadre clignote.

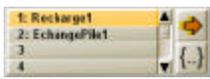
Vous pouvez cliquer sur un objet quelconque pour le sélectionner, ou appuyer sur la touche « Tab » pour sélectionner l'objet de droite.

 Le premier bouton tout à gauche permet d'alterner entre les robots et les bâtiments.

3.2 L'objet sélectionné

La partie inférieure de l'écran présente toutes les commandes de l'objet sélectionné.

3.2.1 Les 10 programmes



Pour les robots, la partie gauche donne accès aux 10 programmes. Les 4 premières lignes donnent les noms des 4 premiers programmes. L'ascenseur permet d'accéder aux 6 programmes suivants.



Ce bouton exécute ou stoppe le programme sélectionné.



Ce bouton édite le programme sélectionné (voir chapitre 5.1). Pendant ce temps, le jeu est en pause. Vous pouvez donc tranquillement réfléchir et modifier le programme.

3.2.2 Les actions



Cette partie de l'interface dépend de l'objet sélectionné.

Notez qu'il existe presque toujours un bouton plus gros que les autres, avec un cadre rouge. C'est cette action qui est effectuée lorsque vous appuyez sur la touche « Entrée ».

3.2.3 Le SatCom



Le premier bouton active le **SatCom** à la page des objectifs de la mission. Le deuxième bouton active le **Satcom** sur une page qui explique le fonctionnement de l'objet sélectionné.

3.2.4 La caméra



Ce bouton permet de changer le point de vue de la caméra. Généralement, on passe alternativement d'une vue interne à une vue arrière. La touche « barre d'espace » a le même effet.

3.2.5 Les jauges



Les jauges permettent de prendre connaissance, en un clin d'œil, de l'état d'un robot. La jauge de droite n'existe qu'avec les robots volants.

3.2.5.1 Niveau d'énergie

Cette jauge donne le niveau de la pile. La partie verte indique l'énergie restante. Une jauge entièrement rouge indique l'absence de pile, ou une pile totalement vide. Lorsqu'il reste moins de 10% d'énergie, un signal sonore retentit, pour vous inciter à trouver la station de recharge la plus proche, ou à changer la pile.

3.2.5.2 Niveau du bouclier

Cette jauge donne le niveau du bouclier individuel. La partie dorée correspond à l'état du bouclier. Lorsque la jauge est presque entièrement noire, le robot est sur le point d'être détruit. Les boules vertes tirées par les fourmis sont très corrosives. Elles attaquent rapidement votre bouclier. Un choc violent affecte aussi le bouclier, ainsi que le bouclier de l'objet heurté.

Pour régénérer le bouclier, il faut amener le robot sur un centre de réparation. Pour régénérer le bouclier d'un bâtiment, c'est un peu plus compliqué, puisque un bâtiment est fixe. Le seul moyen consiste à prendre un robot bouclier et à l'amener près du bâtiment. Lorsque le bâtiment est dans la sphère bleue, son bouclier est rapidement régénéré.

Le bouclier personnel du cosmonaute se régénère automatiquement très lentement. En revanche, le cosmonaute ne peut pas se régénérer sur un centre de réparation.

3.2.5.3 Température du réacteur

Cette jauge indique la température du réacteur des robots volants ou du cosmonaute. Plus le niveau rouge monte et plus le réacteur est chaud. Lorsque le niveau atteint presque le maximum, une sirène retentit, pour vous inciter à atterrir au plus vite. Si vous n'obéissez pas, le réacteur s'arrête et le robot tombe. Si cela vous arrive au-dessus d'un lac, boum ...

Lorsque la température régnant sur une planète est très élevée, les réacteurs surchauffent beaucoup plus vite.

3.2.6 Démolition



Les bâtiments ont un petit bouton permettant l'autodestruction. Evidemment, le titanium ayant servi à la construction du bâtiment n'est pas récupéré.

3.3 La mini-carte



La partie inférieure droite est une mini-carte, qui résume la situation avec une vue aérienne. Les zones sombres correspondent à des creux, vallées, etc. Les zones claires correspondent à des points élevés, montagnes, sommets, etc.

Les robots sont représentés en jaune, les bâtiments en bleu et les ennemis en vert.

La construction d'un radar révélera la position des ennemis sur la carte. Lorsque la souris survole la carte, le symbole de l'objet visé est agrandi et une bulle d'aide en indique le nom. Un clic sélectionne alors immédiatement l'objet, même s'il est situé très loin.

Le curseur à gauche de la carte permet de choisir le facteur de zoom. Lorsqu'il est en bas, la carte montre l'ensemble du terrain. Lorsque qu'il est en haut, un grossissement x8 est effectué.

3.4 Le menu



La petite croix en haut à droite affiche le menu suivant :

Continuer	Ferme le menu et continue la mission.
Enregistrer	Enregistre la mission en cours.
Charger	Charge une mission enregistrée précédemment. La mission en cours est abandonnée définitivement.
Options	Accès aux réglages du jeu. Certaines options ne sont pas accessibles. Pour avoir accès à toutes les options, il faut quitter la mission en cours, et choisir « Options » depuis le menu principal (voir chapitre 2.6).
Recommencer	Annule la mission en cours pour la recommencer depuis le début.
Abandonner	Annule la mission en cours et retourne à l'écran des missions.

4 Objets

4.1 Cosmonaute



Le cosmonaute vous représente dans le jeu. Si vous mourez, la mission échoue immédiatement.

Le sac de survie que vous portez sur le dos vous permet de nager dans l'eau sans limite de temps. Le réacteur placé sous ce sac vous permet de voler dans la plupart des missions.

Vous pouvez transporter la plupart des objets, mais vos déplacements sont alors fortement ralentis. Quand vous portez un objet, vous ne pouvez ni voler, ni vous déplacer sous l'eau. Utilisez dans ce cas un robot déménageur ou un robot sous-marin. A cause de la radioactivité, vous ne pouvez pas transporter du minerai d'uranium.

4.1.1 Pistolet à neutrons

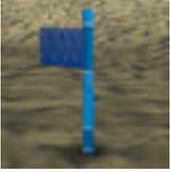


Vous pouvez construire plusieurs types de bâtiments à l'aide du pistolet à neutrons accroché à votre sac de survie. La liste des bâtiments constructibles dépend de la mission et des recherches effectuées.

Un bouton permet de vérifier si le sol est suffisamment plat pour construire des bâtiments. Les zones plates sont vertes, les zones inclinées rouges.

Remarque : Si l'option « marques sur le sol » est déclenchée (voir chapitre 2.6.2), il n'est pas possible de visualiser les zones plates.

4.1.2 Drapeaux



Deux boutons permettent de poser ou d'enlever des drapeaux de couleur. Ceci est utile pour trouver votre chemin, pour retrouver des positions importantes ou pour indiquer des positions à des robots programmés. Il est possible de poser un maximum de 5 drapeaux par couleur.

Catégories : BlueFlag, RedFlag, GreenFlag, YellowFlag et

VioletFlag

4.1.3 SatCom



Le **SatCom** est un assistant très précieux, porté à votre avant-bras gauche. Il résume l'objectif de la mission et reçoit les informations du satellite d'observation. La touche F1 permet de consulter votre **SatCom** en tout temps.

4.2 Bâtiments

4.2.1 Centre de contrôle de Houston



Ce bâtiment est le centre de contrôle de toutes les missions spatiales. Il est situé sur terre, dans le désert du Nevada. Cela fait plusieurs décennies que ce centre de contrôle ne se trouve plus dans l'état du Texas où il se trouvait du temps des missions Apollo. Il a cependant gardé son ancien nom plein de souvenirs et de prestige.

Le centre de contrôle vous transmet les instructions pour les différentes missions. Une équipe d'ingénieurs et de techniciens se relaie jour et nuit pour trouver des solutions aux problèmes que vous rencontrez.

4.2.2 Vaisseau spatial

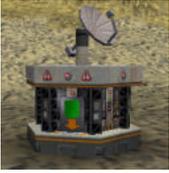


Engin avec lequel vous atterrissez sur les planètes, afin de remplir une mission. Lorsque la mission est terminée, il faut sélectionner ce vaisseau, puis cliquer sur le bouton « décoller ».

En plus de votre propre personne, le vaisseau spatial peut transporter des robots et des matières premières.

Catégorie : SpaceShip

4.2.3 Centre de recherches

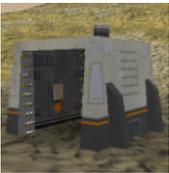


Ce bâtiment est un gigantesque centre de calcul, qui développe les plans de nouvelles technologies pour certains bâtiments ou robots.

Les 65'536 processeurs qu'il contient sont extrêmement gourmands en énergie. Chaque recherche nécessite une nouvelle pile entièrement chargée.

Catégorie : ResearchCenter

4.2.4 Fabrique de robots



Ce bâtiment permet de construire un robot à choix, à partir de titane.

- Déposer le cube de titane à l'intérieur de la fabrique.
- S'éloigner.
- Sélectionner la fabrique.
- Cliquer sur le bouton correspondant au robot à produire.

Le robot produit n'a pas de pile. Il faut lui en mettre une pour qu'il puisse quitter la fabrique.

Remarque : La liste des robots qu'il est possible de produire dépend des recherches effectuées.

Catégorie : BotFactory

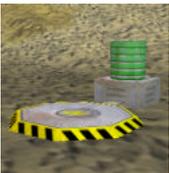
4.2.5 Convertisseur



Ce bâtiment permet de convertir du minerai de titane en titane. Il suffit pour cela de déposer le minerai au centre de la plate-forme, et de reculer pour laisser le champ libre à l'action du convertisseur.

Catégorie : Converter

4.2.6 Station de recharge



Ce bâtiment extrait l'énergie du sous-sol et recharge les piles par induction. Pour recharger un robot, il suffit d'avancer sur la plate-forme de la station, et d'attendre quelques secondes. Une pile transportée est également rechargée. Les piles atomiques ne sont pas rechargeables.

Une station de recharge a besoin d'un sous-sol riche en énergie pour être fonctionnelle. Si le rapport du satellite indique que l'énergie n'est présente que par endroits, il faut au préalable sonder le sous-sol avec un robot renifleur. S'il

dépose une croix verte, l'endroit est adapté à la construction d'une station de recharge ou d'une fabrique de piles.

Si, après la construction, la grosse pile de la station reste vide (rouge), cela signifie que le sous-sol à cet endroit ne contient pas d'énergie.

Catégorie : `PowerStation`

4.2.7 Radar



Ce bâtiment indique la direction de l'élément hostile le plus proche. De plus, les informations fournies par le radar permettent à la minicarte en bas à droite d'afficher la position des robots, bâtiments et ennemis par des carrés et triangles de diverses couleurs.

Les stations de radar servent également de stations de relais pour les communications avec la terre depuis les planètes éloignées.

Catégorie : `RadarStation`

4.2.8 Centre de réparation



Ce bâtiment répare le bouclier des robots endommagés. Le bouclier d'un robot permet d'encaisser un certain nombre de tirs ennemis. Si le niveau du bouclier d'approche de zéro, le prochain tir détruira définitivement le robot.

Catégorie : `RepairCenter`

4.2.9 Tour de défense



Ce bâtiment procure une excellente défense contre les attaques ennemies terrestres ou aériennes. Pour le faire fonctionner, il faut mettre une pile normale ou atomique sur la tour. Attention: la consommation d'énergie est importante. Une pile normale permet 8 tirs. Lorsque la tour n'a pas ou plus d'énergie, elle clignote.

Le rayon d'action d'une tour de défense est de 40 mètres. Pour visualiser la zone défendue, utilisez le bouton « rayon d'action ». Des points rouges bordent la zone circulaire pendant 20 secondes.

Catégorie : `DefenseTower`

4.2.10 Fabrique de piles



Ce bâtiment transforme un cube de titane en une pile chargée, prête à l'emploi.

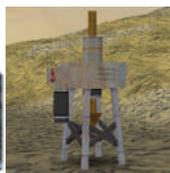
Une fabrique de piles a besoin d'un sous-sol riche en énergie pour être fonctionnelle. Si le rapport du satellite indique que l'énergie n'est présente que par endroits, il faut au préalable sonder le sous-

sol avec un robot renifleur. S'il dépose une croix verte, l'endroit est adapté à la construction d'une fabrique de piles ou d'une station de recharge.

Si, après la construction, la grosse pile de la fabrique reste vide (rouge), cela signifie que le sous-sol à cet endroit ne contient pas d'énergie.

Catégorie : PowerPlant

4.2.11 Derrick



Ce bâtiment permet d'extraire du sous-sol différentes matières premières. Afin de déterminer l'emplacement adapté pour la construction d'un derrick, il faut au préalable sonder le sous-sol avec un robot renifleur. En fonction des marques qu'il dépose, vous pouvez déterminer ce qu'un derrick construit à cet endroit pourra extraire :

- Croix rouge → minerai de titanium.
- Rond jaune → minerai d'uranium.
- Croix verte → énergie, ne convient pas pour la construction d'un derrick.

Catégorie : Derrick

4.2.12 Centrale nucléaire



Ce bâtiment transforme du minerai d'uranium en une pile atomique directement utilisable.

Catégorie : NuclearPlant

4.2.13 Laboratoire



Ce bâtiment analyse la matière organique afin de découvrir de nouvelles technologies issues des insectes. Posez une boule de matière organique sur le socle, sélectionnez le laboratoire, et cliquez sur le bouton correspondant à la recherche à effectuer.

Catégorie : AutoLab

4.2.14 Paratonnerre



Le paratonnerre protège de la foudre, dans un rayon de 50 mètres. De plus, tous les robots et toutes piles normales placés dessous sont rechargés lorsque le paratonnerre est frappé par la foudre. Les piles atomiques ne sont pas rechargées.

Dans le vaste cosmos, rares sont les planètes ayant des orages magnétiques. Parmi les planètes visitées, seule Orphéon présente ce phénomène.

Pour visualiser la zone de protection d'un paratonnerre, sélectionnez-le, puis utilisez le bouton « rayon d'action ». Des points rouges bordent la zone circulaire pendant 20 secondes.

Votre vaisseau spatial fait également office de paratonnerre. Par contre il ne recharge pas les piles.

Catégorie : PowerCaptor

4.2.15 Borne d'échange



Ce bâtiment stocke des informations numériques. Une borne peut contenir jusqu'à 10 informations. Chaque information est référencée par un nom. Pour enregistrer une nouvelle information, il faut utiliser l'instruction `send`, et pour obtenir une information existante, il faut utiliser l'instruction `receive`.

Catégorie : ExchangePost

4.2.16 Coffre-fort



Ce bâtiment a été construit par la première expédition, pour protéger un nouveau type de robot. Afin qu'il soit en sécurité, et que les insectes ne puissent pas y accéder, il est verrouillé par 4 clés. Ce n'est que lorsque vous aurez déposé ces clés sur les 4 emplacements prévus à cet effet qu'il s'ouvrira.

Catégorie : Vault

4.3 Objets transportables

4.3.1 Minerai de titanium



Le minerai de titanium est à la base de pratiquement toutes les constructions, qu'il s'agisse de bâtiments ou de robots. Il se trouve directement à la surface de certaines planètes, ou en sous-sol. Dans ce dernier cas, il faudra construire un derrick pour l'extraire.

Tel quel, le minerai de titanium n'est pas utile. Pour pouvoir l'exploiter, il faut le transformer en titanium à l'aide d'un convertisseur.

Catégorie : TitaniumOre

4.3.2 Minerai d'uranium



Le minerai d'uranium permet de fabriquer des piles atomiques dans une centrale nucléaire. Ce minerai se trouve en surface ou en sous-sol. Dans ce dernier cas, il faut construire un derrick pour l'extraire.

Tous les robots déménageurs peuvent transporter le minerai d'uranium. En revanche, à cause de sa forte radioactivité, le cosmonaute ne peut pas le transporter.

Catégorie : UraniumOre

4.3.3 Titanium



Le titane permet de fabriquer une grande quantité de bâtiments ou de robots différents. Il est généralement apporté en petite quantité par le vaisseau spatial. Pour en obtenir plus, il faudra convertir du minerai de titane.

Catégorie : Titanium

4.3.4 Pile normale



La pile fournit l'énergie aux robots, ainsi qu'à certains bâtiments. Une pile entièrement rouge est vide. L'énergie contenue est représentée par une zone verte plus ou moins grande.

Vous pouvez recharger une pile en vous approchant d'une station de recharge.

Pour fabriquer une pile, il faut une fabrique de piles et un cube de titane.

La pile atomique fabriquée à partir de minerai d'uranium a une capacité 100 fois plus grande que la pile normale.

Catégorie : PowerCell

4.3.5 Pile atomique



La pile atomique fournit l'énergie aux robots, ainsi qu'à certains bâtiments. Elle offre une capacité 100 fois plus grande que la pile normale.

Les piles atomiques ne peuvent être ni rechargées, ni recyclées.

Pour produire une pile atomique, il faut une centrale nucléaire et du minerai d'uranium.

Catégorie : NuclearCell

4.3.6 Boîte noire



Une boîte noire est en réalité orange, afin d'être facilement repérable. Chaque vaisseau spatial est équipé d'une boîte noire qui enregistre en continu tous les événements notables.

Une boîte noire peut également servir à laisser des informations sur place. La première expédition a laissé sur chaque planète une boîte noire contenant les coordonnées de la planète vers laquelle elle s'est dirigée ensuite.

Catégorie : BlackBox

4.3.7 Matière organique



La matière organique est sécrétée par les insectes. Si vous arrivez à en subtiliser dans un nid, cela vous permettra d'effectuer des analyses dans votre laboratoire afin de découvrir de nouvelles technologies dont vous pourrez équiper les robots.

La guêpe transporte souvent des boules de matière organique, afin de vous bombarder.

Catégorie : OrgaMatter

4.3.8 Clés



Les clés permettent d'ouvrir le coffre-fort où la première expédition a mis en sécurité un nouveau système d'armement. Pour cela, il faut 4 clés :

- Clé A, bleue en forme de triangle
- Clé B, rouge en forme de pentagone
- Clé C, verte en forme d'étoile à 6 branches
- Clé D, jaune de forme octogonale

Catégories : KeyA, KeyB, KeyC et KeyD

4.4 Robots

4.4.1 Les modes de propulsion

Un robot déménageur, renifleur, shooter ou orgaShooter peut être équipé de 4 moyens de propulsion différents. Vous découvrirez ces systèmes au fur et à mesure de votre progression dans les missions.

	Roues	Chenilles	Réacteur	Pattes
Déménageur				
Renifleur				
Shooter				
OrgaShooter				

4.4.1.1 Propulsion à roues

La propulsion avec des roues est le mode standard, rapide, consommant peu d'énergie et adapté aux terrains relativement plats. Lorsque le relief devient trop accidenté, il faut opter si possible pour un robot volant ou, si le vol est impossible, pour une propulsion à chenilles.

4.4.1.2 Propulsion à chenilles

Les chenilles permettent de gravir de fortes pentes. En contrepartie, le déplacement est lent et gourmand en énergie. Pour de courtes distances en terrain relativement plat, il vaut mieux opter pour une propulsion à roues. Quand il est possible de construire des robots volants, ceux-ci représentent la meilleure solution pour les longues distances.

4.4.1.3 Propulsion à réacteur (robot volant)

Les robots volants peuvent franchir des obstacles tels que des montagnes ou des étendues d'eau, mais ils consomment beaucoup d'énergie. Au sol, ils se déplacent lentement. Pour des déplacements courts, il vaut mieux opter pour des robots à roues.

Il faut surveiller la température du réacteur affichée en bas de la fenêtre. Lorsque le réacteur surchauffe, il s'arrête complètement et le robot tombe (voir chapitre 3.2.5.3).

4.4.1.4 Propulsion à pattes

Les robots à pattes consomment deux fois moins d'énergie que les robots à roues lors des déplacements. De plus, ils peuvent gravir de très fortes pentes.

4.4.2 Les fonctions

Un robot à roues, à chenilles, à réacteur ou à pattes peut avoir 4 fonctions différentes.

4.4.2.1 Fonction déménageur

Le robot déménageur transporte des matières premières, des piles ou d'autres petits objets. Les objets suivants peuvent être transportés :

- Minerai de titanium
- Minerai d'uranium
- Cube de titanium
- Pile normale
- Pile atomique
- Boîte noire
- Clé
- Matière organique



Prend ou dépose un objet, à la position indiquée par les trois boutons suivants :



L'objet est devant le robot, au sol ou à l'arrière d'un autre robot.



L'objet est la propre pile du robot.



L'objet est derrière le robot, au sol.

4.4.2.2 Fonction renifleur

Un robot renifleur peut explorer le sous-sol. S'il découvre quelque chose, il dépose les marques suivantes :

- Croix rouge → minerai de titanium.
- Rond jaune → minerai d'uranium.
- Croix verte → énergie, utile par exemple pour la station de recharge.

4.4.2.3 Fonction canon shooter

Le canon shooter permet de se défendre contre les ennemis. Mais attention, l'énergie consommée est très importante. Avec une pile normale, il est possible de tirer quatre fois.

Astuce : Pendant le tir, il est possible de déplacer la souris, pour *balayer* toute une zone.

4.4.2.4 Fonction canon orgaShooter

Le canon orgaShooter est plus efficace que le canon shooter. Il projette de petites boules de matière organique corrosive. Une pile normale permet de tirer onze fois.

Astuce : Pendant le tir, il est possible de déplacer la souris, pour *balayer* toute une zone.

4.4.3 Robot secoueur



Ce robot frappe un énorme coup par terre, ce qui retourne sur le dos toutes les fourmis et les araignées dans un rayon de 100 mètres. Un insecte sur le dos n'est pas mort. Il essaie désespérément de se retourner. Après environ 60 secondes d'efforts, il y parvient généralement.

Pour visualiser la zone d'impact, utilisez le bouton « rayon d'action ». Des points rouges bordent la zone circulaire pendant 20 secondes.

L'énergie consommée par le robot est importante. Avec une pile normale, un coup vide les 2/5ème de la pile.

4.4.4 Robot recycleur



Robot permettant de transformer une épave de robot en un cube de titanium.

4.4.5 Robot bouclier



Robot bouclier, qui protège de toutes les attaques ennemies dans un rayon variant entre 10 et 25 mètres. Ce robot est le seul moyen de traverser des passages infestés de champignons verts. Le bouclier personnel des robots et des bâtiments est régénéré par l'action de cette protection.

Une pile normale permet de maintenir le bouclier en activité pendant 20 secondes avec le rayon maximal de 25 mètres, ce qui est en général insuffisant. Une pile atomique est beaucoup mieux adaptée à ce robot.

La consommation d'énergie est proportionnelle au rayon du bouclier. Avec un rayon minimal de 10 mètres, le bouclier est maintenu 2.5x plus longtemps.

4.4.6 Robot sous-marin



Robot sous-marin avec un bras manipulateur. Avec le cosmonaute, ce robot est le seul à pouvoir se déplacer sous l'eau. Il faut contrôler le niveau de la pile avant de plonger, car le remplacement ou la recharge de la pile est impossible sous l'eau.

Le sous-marin ne peut prendre ou déposer des objets qu'au sol. Il est incapable de prendre la pile à l'arrière d'un robot, par exemple.

4.4.7 Robot phazer



Robot à canon phazer très puissant. En orientant le canon vers le haut, on arrive à tirer jusqu'à la distance de 60 mètres. Ce canon est la seule arme qui puisse venir à bout de la reine pondreuse.

4.5 Ennemis

4.5.1 Fourmi



La fourmi représente l'ennemi le plus courant et sans doute le plus redoutable. Elle tire de petites boules corrosives, qui finissent par faire exploser les robots et les bâtiments.

Vous êtes bien plus résistant contre cette arme que les robots et les bâtiments et votre bouclier se régénère au fur et à mesure que vos blessures guérissent, mais si elles se mettent à plusieurs, elles finissent en général par vous avoir à l'usure.

Catégorie : AlienAnt

4.5.2 Araignée



L'araignée fonctionne à la manière d'un commando suicide: lorsqu'elle arrive à proximité d'une cible, elle gonfle son abdomen et explose en une multitude de petits fragments pyrogènes. Ces fragments incendient tout ce qu'ils heurtent.

Catégorie : AlienSpider

4.5.3 Guêpe

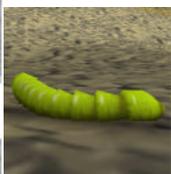


La guêpe est redoutable, car elle vole rapidement, ce qui la rend difficile à abattre. Elle cherche des boules de matière organique qu'elle lâche de haut sur vos robots et bâtiments et sur vous-même.

La tour de défense est le meilleur moyen de s'en débarrasser.

Catégorie : AlienWasp

4.5.4 Ver



Dès qu'un ver s'approche d'un robot programmé, celui-ci est atteint par un virus. Le virus modifie le programme, qui ne fonctionne plus du tout ou de façon erronée.

Le ver avance tantôt à découvert, tantôt sous terre. Lorsqu'il est sous terre, il est indétectable et indestructible.

Catégorie : AlienWorm

4.5.5 Reine pondreuse



Cet immense insecte pond des œufs desquels naîtront divers insectes hostiles. Sa carapace est particulièrement résistante aux tirs de vos systèmes d'armement offensifs.

Catégorie : AlienQueen

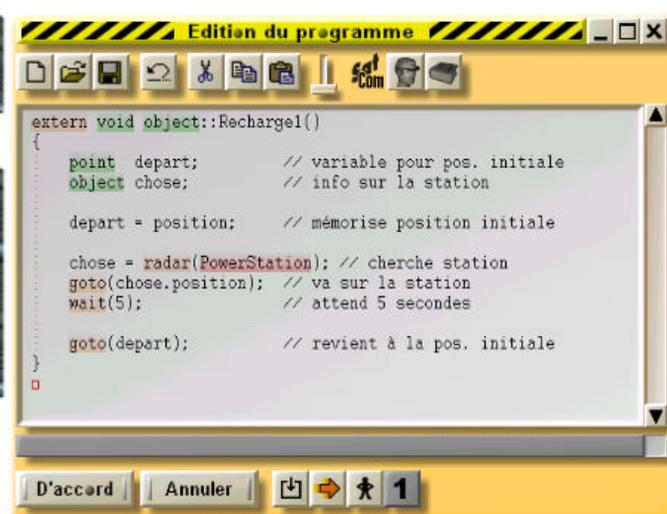
5 Programmation

Tous les robots sont programmables.

5.1 L'éditeur de programmes

Pour entrer dans l'éditeur de programme, il faut :

- Sélectionner un robot.
- Choisir le programme à éditer parmi les 10 à disposition.
- Cliquer le bouton « édite le programme sélectionné ».



Le jeu est automatiquement mis en pause pendant l'édition. Si vous approchez la souris des bords gauche ou droite de l'écran, le point de vue tourne. Si vous approchez la souris des bords supérieur ou inférieur, le point de vue avance ou recule. Cela vous permet d'observer tranquillement la situation. Il est très utile de réduire la fenêtre pendant ce temps.

- **Taille réduite**, pour réduire la fenêtre à une barre tout en bas de l'écran.
- **Taille maximale**, pour agrandir la fenêtre sur tout l'écran.
- **Fermer**, équivalent au bouton « D'accord ».

Si vous entrez dans l'éditeur pendant que le programme s'exécute, le jeu n'est pas mis en pause. Cela permet alors d'observer le déroulement du programme (voir chapitre 5.1.11).

La fenêtre de l'éditeur peut être déplacée en tirant la barre de titre avec la souris. Elle peut également être redimensionnée en tirant les bords ou les coins. La prochaine fois que vous ferez appel à l'éditeur, la fenêtre sera à la même position et aura les mêmes dimensions.

Les mots clés du langage apparaissent en couleur, pour faciliter la compréhension :

Couleur	Nature	Exemple
Orange	Instruction	aim, fire, turn, goto, while, etc.
Vert	Type d'une variable	object, float, int, etc.
Rouge	Constante, catégorie	TitaniumOre, Converter, BlackBox, etc.

Lorsque le curseur est sur un mot clé du langage, la barre sous la fenêtre d'édition résume succinctement la syntaxe. Un clic sur cette barre appelle le **SatCom** pour vous donner plus de précisions. La touche « F3 » a le même effet.

Un double-clic sur un mot le sélectionne rapidement. Si le bouton de la souris est maintenu pressé, les caractères sont mis en évidence. « Shift-flèche » met également en évidence quelques caractères. Les touches flèches actionnées avec la touche « Ctrl » permettent de se déplacer par mots. « Ctrl+Shift+flèche » met en évidence par mots entiers.

5.1.1 Nouveau



Efface tout le programme en cours d'édition, et crée un squelette vide de la forme :

```
extern void object::Nouveau( )
{
}

```

Nouveau est le nom du programme, que vous pouvez remplacer par un nom de votre invention, mais sans espaces. Par exemple Recherche, ApporteTitanium, RevientBase, etc.

Remarque : Si vous avez cliqué ce bouton par erreur, il suffit d'annuler l'action.

5.1.2 Ouvrir et Enregistrer



Tous les programmes que vous créez sont automatiquement sauvegardés dans la mission ou l'exercice en cours. En revanche, si vous désirez réutiliser un programme dans une autre mission, il faut l'enregistrer explicitement, pour pouvoir le rouvrir pas la suite.

Privé Le programme est mis dans un dossier privé, qui dépend donc du nom donné au début du jeu (voir chapitre 2.7).

Public Le programme est mis dans un dossier commun à tous les joueurs.

Juste en dessous de la barre de titre du dialogue apparaît le nom du dossier. Par exemple, **SavegameJoueur\Program** signifie que le dossier est généralement **c:\Program Files\Colobot\Savegame\Joueur\Program** (selon le dossier dans lequel vous installé le jeu). Cela vous permet de récupérer un programme dans une

application extérieure à COLOBOT. Par exemple pour envoyer un programme par e-mail à un ami.

Les touches « Ctrl+O » et « Ctrl+S » permettent d'accéder rapidement aux fonctions *Ouvrir* et *Enregistrer*.

5.1.3 Annuler



Annule la dernière action effectuée dans le programme. Il est ainsi possible d'annuler en tout temps les 20 dernières modifications.

Le touche « Ctrl+Z » a le même effet.

5.1.4 Couper, Copier et Coller



Coupe ou *copie* les caractères mis en évidence, pour pouvoir les *coller* ailleurs. Si aucun caractère n'est mis en évidence, c'est toute la ligne qui est prise en compte.

Les touches « Ctrl+X », « Ctrl+C » et « Ctrl+V » permettent d'accéder rapidement aux fonctions *Couper*, *Copier* et *Coller*.

5.1.5 Taille des caractères

Ce petit curseur permet de choisir la taille des caractères pendant l'édition et dans le **SatCom**. Cette taille est mémorisée jusqu'à ce qu'elle soit changée à nouveau.

5.1.6 SatCom : Instructions



Donne les objectifs de la mission ou de l'exercice. Vous pouvez aussi appuyer sur « F1 ».

5.1.7 SatCom : Aide à la programmation



Donne les instructions générales sur la programmation, à partir desquelles vous avez accès à toutes les explications spécifiques par des liens hypertextes. Vous pouvez aussi appuyer sur « F2 ».

5.1.8 D'accord

Compile le programme puis quitte l'éditeur si aucune erreur n'est détectée. En cas d'erreur, la partie incriminée est mise en évidence et la barre sous la fenêtre d'édition indique la nature de l'erreur.

5.1.9 Annuler

Quitte l'éditeur sans compiler le programme. Toutes les modifications effectuées dans le texte sont néanmoins enregistrées.

5.1.10 Compiler



Compile le programme, sans quitter l'éditeur. Cela vous permet de vérifier si le programme contient ou non des erreurs.

5.1.11 Démarrer/stopper



Démarre ou stoppe le programme, sans quitter l'éditeur. Cela permet de faire du *debug*, c'est-à-dire de chercher les causes d'un éventuel mauvais fonctionnement. Lorsque le bouton de droite représente un homme avec les mains sur les hanches, le programme est exécuté en *pas à pas*, c'est-à-dire instruction par instruction.

Remarque : Lorsque le programme est en exécution, la fenêtre d'édition devient orange. Vous pouvez alors observer le déroulement du programme, mais vous ne pouvez plus le modifier.

5.1.12 Pause/continuer



Passé alternativement du mode *pas à pas* au mode sans interruption.

5.1.13 Un pas



Exécute l'instruction suivante, en mode *pas à pas*. La partie sous la fenêtre d'édition montre alors le contenu des différentes variables du programme. Ce contenu change à chaque pas, en fonction des affectations effectuées.

5.2 Le langage CBOT

Le langage CBOT est très proche du C++ et du Java™. Il est adapté à COLOBOT, et également adapté à une approche pédagogique. Dans ce manuel, nous ne présentons que quelques exemples simples. Pour avoir une description complète, utilisez le **SatCom** en appuyant sur « F2 ».

Attention de respecter les majuscules et les minuscules. Par exemple, si vous écrivez `Radar`, le programme ne sera pas compilé. Il faut écrire `radar`.

Chaque instruction est terminée par un point-virgule ;.

Un commentaire commence par // et se termine à la fin de la ligne. Il n'a aucune incidence sur le déroulement du programme, mais sert uniquement à le rendre plus explicite.

5.2.1 L'instruction `radar`

Avec l'instruction `radar()`, le robot peut chercher des objets comme des ennemis, des robots ou des bâtiments. Donnez entre parenthèses le nom de l'objet que vous cherchez. Mettez le résultat dans une variable de type `object`. Voici par exemple comment chercher la fourmi la plus proche :

```
// Tout au début du programme:
object chose; // déclaration de la variable

// Cherche la fourmi la plus proche:
chose = radar(AlienAnt);
```

5.2.2 L'instruction goto

L'instruction `goto()`; permet de déplacer le robot vers une certaine position. En général, on déplace le robot vers un objet qu'on a détecté avec l'instruction `radar()`; . Si on a mis les informations rendues par `radar()`; dans une certaine variable, il faut écrire le nom de la variable suivi de `.position` pour obtenir la position de l'objet. Voici par exemple comment chercher un cube de titane, aller vers la position du cube et le saisir :

```
object chose;
chose = radar(Titanium);
goto(chose.position);
grab();
```

5.2.3 Les instructions grab et drop

L'instruction `grab()`; permet de saisir avec la pince du bras manipulateur un objet qui se trouve au sol, sur l'emplacement arrière d'un robot ou sur le socle d'un bâtiment. L'instruction `drop()`; dépose l'objet actuellement transporté. Voici par exemple comment saisir ce qui se trouve devant le robot et le poser 5m plus loin :

```
grab(); // prend l'objet
move(5); // avance de 5m
drop(); // dépose l'objet
```

5.2.4 L'instruction fire

L'instruction `fire()`; permet de faire feu avec le canon du robot. En général, on tire des rafales d'une seconde :

```
fire(1);
```

5.2.5 L'instruction while

L'instruction `while() {}` permet de répéter des instructions plusieurs fois. L'utilisation la plus courante de `while` consiste à répéter des instructions encore et encore, à l'infini. Pour ce faire, on écrit `while (true) {}` et on met les instructions à répéter entre les accolades `{}`. Voici comment répéter le fait de chercher une araignée, de se tourner vers elle et de tirer :

```
while (true)
{
    chose = radar(AlienSpider);
    turn(direction(chose.position));
    fire(1);
}
```

Il suffit d'exécuter ce programme une seule fois, il tuera toutes les araignées autour de lui.

5.2.6 L'instruction *distance*

Avec l'instruction `distance(,)` vous pouvez calculer la distance entre deux positions. Si vous écrivez `position` tout seul, cela donne la position du robot qui exécute le programme. Si vous écrivez le nom d'une variable suivie de `.position`, cela donne la position de l'objet décrit par la variable. Voici comment avancer d'une distance égale à la distance entre le robot et la fourmi la plus proche :

```
chose = radar(AlienAnt);  
move(distance(position, chose.position));
```

Ceci est bien sûr parfaitement suicidaire, mieux vaut s'arrêter 40m avant, pour être à la bonne distance de tir :

```
chose = radar(AlienAnt);  
move(distance(position, chose.position) - 40);
```

5.2.7 L'instruction *if*

L'instruction `if() { }` permet d'exécuter des instructions seulement à une certaine condition. La condition est donnée entre parenthèses `()`, les instructions entre accolades `{ }`. Voici par exemple comment faire en sorte que le robot tire seulement si la distance à la cible est inférieure à 40m :

```
chose = radar(AlienAnt);  
if (distance(position, chose.position) < 40)  
{  
    fire(1);  
}
```

Si l'instruction `radar()` ne trouve pas l'objet cherché, elle rend la valeur `null`.

Ainsi il est possible de tester si un objet n'existe pas avec `(chose == null)`, ou de tester si l'objet existe avec `(chose != null)`. Deux signes égal `==` testent l'égalité, un point d'exclamation suivi d'un signe égal `!=` teste l'inégalité. Voici comment aller se recharger seulement s'il y a une station de recharge :

```
chose = radar(PowerStation);  
if (chose != null)  
{  
    goto(chose.position);  
    wait(5);  
}
```

5.2.8 L'instruction *motor*

L'instruction `motor(,)` permet de donner directement une vitesse aux moteurs gauche et droite du robot. `motor` conserve la vitesse du moteur pendant l'exécution des instructions suivantes. Ceci est utile pour faire tourner le robot pendant l'instruction `fire()`. Ainsi, il est possible de balayer toute une zone avec la rafale. Voici par exemple comment balayer toute la zone qui se trouve devant le robot :

```
turn(45); // tourne à gauche de 45 degrés
motor(0.5, -0.5); // rotation lente à droite
fire(2); // feu
motor(0,0); // immobilise le robot
```

En faisant avancer le moteur gauche à demi-vitesse et en faisant reculer le moteur de droite à demi-vitesse, le robot tourne lentement sur lui-même pendant la rafale de 2 secondes.

5.2.9 L'instruction *turn*

Utilisez l'instruction `turn()` pour faire tourner le robot d'un certain nombre de degrés sur lui-même. 90 degrés signifie un quart de tour, 180 degrés signifie un demi-tour, 360 degrés signifie un tour complet. Voici quelques exemples avec `turn()`:

```
turn(90); // quart de tour à gauche
turn(-90); // quart de tour à droite (négatif)
turn(180); // demi-tour
```

Pour se tourner vers un objet trouvé avec l'instruction `radar()`, il faut calculer l'angle de la rotation avec l'instruction `direction()`:

```
chose = radar(AlienSpider);
turn(direction(chose.position));
```

Puis il suffit de faire feu, et il y a un élément hostile en moins.

6 Copyright

La photo de la nébuleuse NGC3603 servant de fond pour la planète Orphéon a été prise avec le télescope spatial Hubble. Elle est utilisée avec l'autorisation des auteurs Wolfgang Brandner (JPL/IPAC), Eva K. Grebel (Université de Washington), You-Hua Chu (Université d'Illinois Urbana-Champaign) et de la NASA.

Le son de tonnerre de la planète Orphéon est utilisé avec l'autorisation limitée de CREATIVE moyennant la mention :

Material from products are used by limited permission from CREATIVE.

Windows[®] est une marque déposée de Microsoft.

DirectX[®] est une marque déposée de Microsoft.

Java™ est une marque déposée de Sun Microsystem.

7 Développeur

Daniel Roux, Denis Dumoulin, Otto Kölbl, Michael Walz, Didier Gertsch

EPSITEC SA, Mouette 5, CH-1092 Belmont

colobot@epsitec.ch

www.colobot.com

7.1 Beta testeurs

Adrien Roux, Didier Raboud, Nicolas Beuchat, Joël Roux

Michael Jubin, Daniel Sauthier, Nicolas Stubi, Patrick Thévoz

8 Editeur de la version française

ALSYD, 43 Ch. du vieux Chêne, F-38240 Meylan

support@fr.alsyd.com

www.alsyd.com

9 Table des matières

1	Introduction.....	1
1.1	Histoire.....	1
1.2	Configuration requise.....	2
1.3	Installation.....	2
1.4	Désinstallation.....	2
2	Menu principal.....	2
2.1	Missions.....	2
2.2	Jeu libre.....	2
2.3	Programmation.....	2
2.4	Défis.....	3
2.5	Suppl.....	3
2.6	Options.....	3
2.6.1	Affichage.....	3
2.6.2	Graphique.....	4
2.6.3	Jeu.....	6
2.6.4	Commandes.....	7
2.6.5	Son.....	9
2.7	Joueur.....	10
3	L'écran.....	11
3.1	Le résumé.....	11
3.2	L'objet sélectionné.....	11
3.2.1	Les 10 programmes.....	12
3.2.2	Les actions.....	12
3.2.3	Le SatCom.....	12
3.2.4	La caméra.....	12
3.2.5	Les jauges.....	12
3.2.6	Démolition.....	13
3.3	La mini-carte.....	13
3.4	Le menu.....	14
4	Objets.....	14
4.1	Cosmonaute.....	14
4.1.1	Pistolet à neutrons.....	14
4.1.2	Drapeaux.....	15
4.1.3	SatCom.....	15
4.2	Bâtiments.....	15
4.2.1	Centre de contrôle de Houston.....	15
4.2.2	Vaisseau spatial.....	15
4.2.3	Centre de recherches.....	16
4.2.4	Fabrique de robots.....	16
4.2.5	Convertisseur.....	16
4.2.6	Station de recharge.....	16
4.2.7	Radar.....	17
4.2.8	Centre de réparation.....	17
4.2.9	Tour de défense.....	17
4.2.10	Fabrique de piles.....	17
4.2.11	Derrick.....	18

4.2.12	Centrale nucléaire.....	18
4.2.13	Laboratoire.....	18
4.2.14	Paratonnerre.....	18
4.2.15	Borne d'échange.....	19
4.2.16	Coffre-fort.....	19
4.3	Objets transportables	19
4.3.1	Minerai de titane	19
4.3.2	Minerai d'uranium	19
4.3.3	Titane.....	20
4.3.4	Pile normale.....	20
4.3.5	Pile atomique.....	20
4.3.6	Boîte noire.....	21
4.3.7	Matière organique.....	21
4.3.8	Clés	21
4.4	Robots	22
4.4.1	Les modes de propulsion.....	22
4.4.2	Les fonctions	23
4.4.3	Robot secoueur.....	24
4.4.4	Robot recycleur.....	24
4.4.5	Robot bouclier.....	24
4.4.6	Robot sous-marin	25
4.4.7	Robot phazer	25
4.5	Ennemis	25
4.5.1	Fourmi.....	25
4.5.2	Araignée.....	25
4.5.3	Guêpe.....	26
4.5.4	Ver.....	26
4.5.5	Reine pondreuse.....	26
5	Programmation.....	27
5.1	L'éditeur de programmes.....	27
5.1.1	Nouveau.....	28
5.1.2	Ouvrir et Enregistrer.....	28
5.1.3	Annuler.....	29
5.1.4	Couper, Copier et Coller.....	29
5.1.5	Taille des caractères.....	29
5.1.6	SatCom : Instructions.....	29
5.1.7	SatCom : Aide à la programmation.....	29
5.1.8	D'accord	29
5.1.9	Annuler	29
5.1.10	Compiler.....	30
5.1.11	Démarrer/stopper.....	30
5.1.12	Pause/continuer.....	30
5.1.13	Un pas	30
5.2	Le langage CBOT.....	30
5.2.1	L'instruction radar.....	30
5.2.2	L'instruction goto.....	31
5.2.3	Les instructions grab et drop	31
5.2.4	L'instruction fire.....	31



5.2.5	L'instruction while.....	31
5.2.6	L'instruction distance	32
5.2.7	L'instruction if.....	32
5.2.8	L'instruction motor.....	32
5.2.9	L'instruction turn.....	33
6	Copyright.....	34
7	Développeur.....	34
7.1	Beta testeurs	34
8	Editeur de la version française.....	34
9	Table des matières.....	35