



## TP 4 Projet 2021

### Encodage de message dans un parachute

**Durée :** 3 séances  
**Objectifs :** Réalisation d'un logiciel complet.  
**Modalité de travail :** Le projet réalisé en binômes. L'utilisation de git est fortement recommandée.



FIGURE 1 – Vidéo YouTube [de l'atterrissage du rover \*Perseverance\* sur la planète Mars.](#)

## 1 Présentation de l'objectif

Vous avez peut-être suivi l'histoire du rover *Perseverance* récemment arrivé sur la planète Mars. Vous avez peut-être aussi entendu parler du message « caché » dans le parachute utilisé pour ralentir son atterrissage. Dans le cas contraire, vous pouvez consulter le site du journal Le Monde [et ce post](#) du forum Reddit [.](#)

Le but de ce projet est de réaliser un logiciel permettant de réaliser le plan d'un parachute pour un message et des caractéristiques données. Au minimum, ces caractéristiques comprendront les nombres de secteurs et de pistes. Les termes de *secteur* et *piste* sont utilisés dans toute la suite par analogie à la structure physique d'une disquette [.](#) (voir aussi la figure 3).

Dans le cas de *Perseverance*, un caractère est converti en une séquence de bits correspondant aux 7 bits de son code ASCII (décalé pour que le caractère @ ait comme valeur zéro, cf. tableau 1).

Caractère	Code ASCII	ASCII - 64	Binaire (7 bits)
@	64	0	0000000
A	65	1	0000001
B	66	2	0000010
C	67	3	0000011
...	...	...	...

Tableau 1 – Extrait de la table ASCII utilisée avec décalage.

Dans le parachute original, chaque caractère est placé dans un bloc de 10 éléments consécutifs (7 bits + 3 zéros) avec sur chaque cercle un marqueur de début/fin composé des deux blocs 0001111111 1111111111, comme illustré sur cette image [🔗](#) ou encore sur l'infographie du parachute au centre de cet article du site de la BBC [🔗](#). Dans le cadre de ce projet, on pourra se contenter convertir une séquence de  $N$  caractères en une séquence de  $N \times 7$  bits qui seront dessinés dans une série de trapèzes formant un motif de  $S$  secteurs et  $T$  pistes.

Par exemple, le texte ENSICAEN\_RULES (avec @ comme valeur de zéro) produira la séquence binaire :  
 0000101 0001110 0010011 0001001 0000011 0000001 0000101 0001110 0011111 0010010 0010101 000-  
 1100 0000101 0010011

et pour un parachute composé de 5 pistes et 21 secteurs ( $3 \times 7$ ), l'image de la figure 2(a).

Des indications sur la façon dont chaque trapèze peut être dessiné sont données dans la section 3.

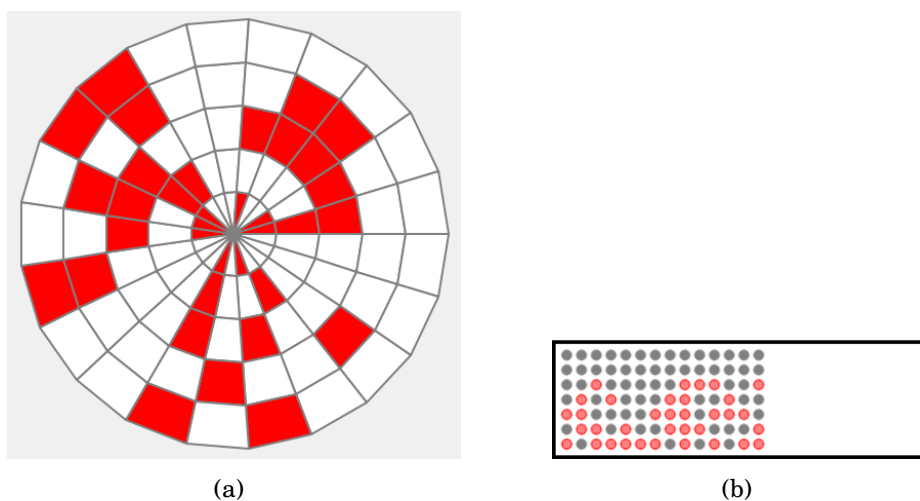


FIGURE 2 – Deux vues différentes du texte ENSICAEN\_RULES.

## 1.1 Fonctionnalités fortement suggérées

- Le message devra être géré par une classe de type modèle offrant notamment un accès au  $k$ -ième bit du message.
- Il sera possible de régler le nombre de secteurs par un *slider* et une *spinbox*.
- Il sera possible de régler le nombre de cercles concentriques (ou « pistes ») par un *slider* et une *spinbox*.
- On disposera d'une vue « binaire » (figure 2(b)) en plus de la vue parachute.
- L'application disposera de menus et actions standards ainsi que de raccourcis.
- La traduction de l'interface sera prévue dans au moins une langue supplémentaire.
- La sauvegarde du texte et des paramètres du parachute sera possible (et bien sûr la lecture depuis un fichier).

## 1.2 Idées de fonctionnalités additionnelles

On donne ici des idées d'améliorations du logiciel ou de son interface. Attention : s'il est vrai que l'ajout de fonctionnalités est un plus, en offrir un maximum ne doit pas être votre premier objectif. La priorité doit être donnée à la réalisation d'une application bien programmée, totalement fonctionnelle et ergonomique (cf. section 2).

- Disque central vide (peut être une option).

- Offrir la possibilité de changer le caractère de référence (c.-à-d. le zéro de l'encodage qui est le caractère @ par défaut)
- Permettre de choisir les deux couleurs utilisées.
- Offrir un mode de couleurs aléatoires (pour les bits à 1).
- Offrir le choix d'afficher des blocs de 10 trapèzes (7 bits + 3 blancs) comme dans le parachute original.
- Permettre de choisir des nombres de secteurs simplement parmi les multiples de 7 (ou de 10 si vous avez choisi d'implémenter cette convention).
- Offrir le choix d'ajouter le motif 0001111111 1111111111 sur chaque piste.
- Dessin plus fidèle, avec les dents de scie entre les pistes (chaque pièce de tissu est un parallélogramme et non pas un trapèze).
- Toute extension sortie de votre imagination, dans la mesure où elle apporte un plus à l'application !

Bien entendu, les paramètres que vous ajoutez devront être prévus dans les fichiers de sauvegarde.

## 2 Évaluation

Pour rappel, la note que vous obtiendrez pour ce projet sera votre unique note dans le module « Conception d'interfaces graphiques ». Votre logiciel sera jugé sur :

- les fonctionnalités implémentées ;
- la qualité de la modélisation ;
- la qualité du code source ;
- la qualité visuelle et ergonomique de l'interface (cf. cours) ;
- l'utilisation de patrons adaptés (architecturaux comme Modèle-Vue-Présentation, de conception comme Observateur).

Ce projet fera l'objet d'une présentation orale avec support et courte démonstration devant le groupe de TP lors de la 4<sup>e</sup> et dernière séance. Chaque présentation durera 15 minutes tout compris (installation, présentation/démonstration, questions). Prévoyez 5 à 7 minutes de présentation/démonstration pendant lesquelles vous devrez montrer la qualité de votre solution et de sa conception. La présentation fera aussi le point sur les fonctionnalités implémentées et celles qui ne l'ont pas été parmi celles qui étaient imposées. Vous présenterez aussi bien sûr les fonctions innovantes de votre logiciel.

Le code source compilable de votre projet devra être rendu sous forme d'archive la veille de la présentation. Cette archive devra être soignée et donc dépourvue de fichiers inutiles.

## 3 Géométrie du parachute

La figure 3 montre le trapèze associé au bit  $k = 67$  d'un message (pour une numérotation à partir de 0), à raison de 7 bits par caractère. Les différentes étapes de calcul des coordonnées des 4 sommets du trapèze sont données, ligne après ligne à partir des paramètres  $(k, R, T, S)$ , dans le tableau 2. En suivant cette démarche, dessiner le parachute consiste à dessiner les  $S \times T$  trapèzes, pour  $k = 0, \dots, (S \times T - 1)$ . Notez que la numérotation des trapèzes est faite ici dans le sens horaire et non pas trigonométrique.

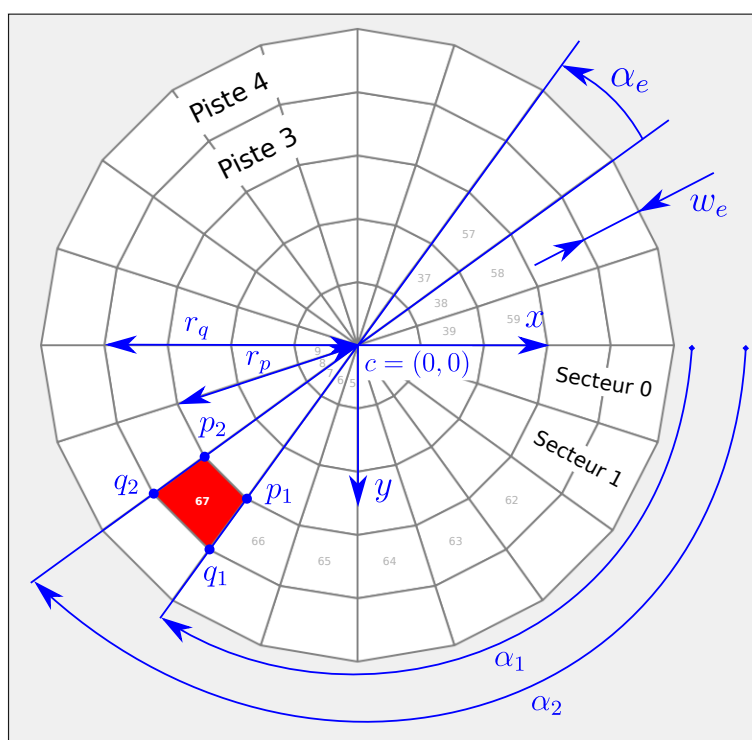


FIGURE 3 – Un parachute composé de 5 pistes et 20 secteurs, soient 100 trapèzes. En rouge, le trapèze d'indice  $k = 67$ .

$k$	Indice du trapèze (ici 67)	
$R$	Rayon du parachute	
$T$	Nombre de pistes concentriques (ici $T = 5$ )	
$S$	Nombre de secteurs (ici $S = 20$ )	
$\alpha_e$	Angle d'un secteur	$2\pi/S$
$w_e$	Épaisseur d'une piste	$R/T$
$s$	Numéro de secteur	$k \bmod S$
$\alpha_1$	Premier angle	$s \cdot \alpha_e$
$\alpha_2$	Second angle	$(s + 1) \cdot \alpha_e$
$t$	Numéro de la piste	$k \div S$
$r_p$	Rayon intérieur	$t \cdot w_e$
$r_q$	Rayon extérieur	$(t + 1) \cdot w_e$
$p_1$	Premier sommet intérieur	$(x, y) = (r_p \cdot \cos(\alpha_1), r_p \cdot \sin(\alpha_1))$
$p_2$	Second sommet intérieur	$(x, y) = (r_p \cdot \cos(\alpha_2), r_p \cdot \sin(\alpha_2))$
$q_1$	Premier sommet extérieur	$(x, y) = (r_q \cdot \cos(\alpha_1), r_q \cdot \sin(\alpha_1))$
$q_2$	Second sommet extérieur	$(x, y) = (r_q \cdot \cos(\alpha_2), r_q \cdot \sin(\alpha_2))$

Tableau 2 – Calcul des coordonnées des sommets du trapèze d'indice  $k$  dans le repère centré sur l'apex du parachute (figure 3).