

Codage morse et alphabet radio

Version scratch

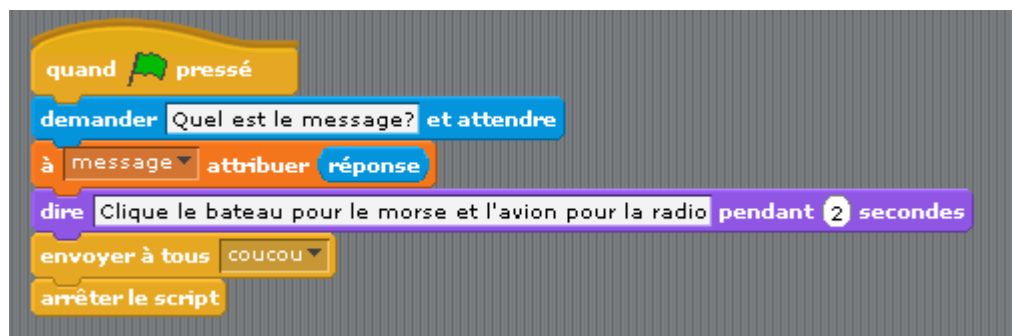
Scratch n'est pas idéal pour les manipulations de texte, néanmoins, avec un peu de patience, on y arrive.

Il y a 3 objets, le chat de scratch qui accueille l'utilisateur, lui demande un message à saisir, et lui demande de cliquer soit sur un avion (pour le codage avec l'alphabet radio), soit sur un bateau (pour le morse).

Les alphabets ne sont pas complets, je n'ai pris que les lettres. Contrainte: les lettres doivent être en minuscule, tous les caractères non reconnus seront codés par "??".

Scripts du chat: le rôle principal du chat est de saisir le message à coder et de créer les alphabets morse et radio:

Voici comment le chat saisit le message.



Voici comment il crée un alphabet (évidemment, il faut rentrer ainsi le codage des 26 lettres des 2 alphabets, c'est un peu long mais on doit faire cette saisie quel que soit le langage choisi). Ne pas oublier de vider les listes avant de les remplir, sinon elles grandissent à chaque exécution!



Quand le chat envoie le message "coucou", c'est aux autres de bosser. Il se cache.



Enfin quand l'avion ou le bateau ont fini le travail, ils envoient le message "fini": le chat réapparaît et dit le code.



Scripts de l'avion : les scripts du bateau sont identiques à l'alphabet utilisé près. D'ailleurs, pour gagner du temps, l'objet bateau a été créé en dupliquant l'avion, puis en l'éditant.

3 scripts règlent quand l'avion est montré ou caché.



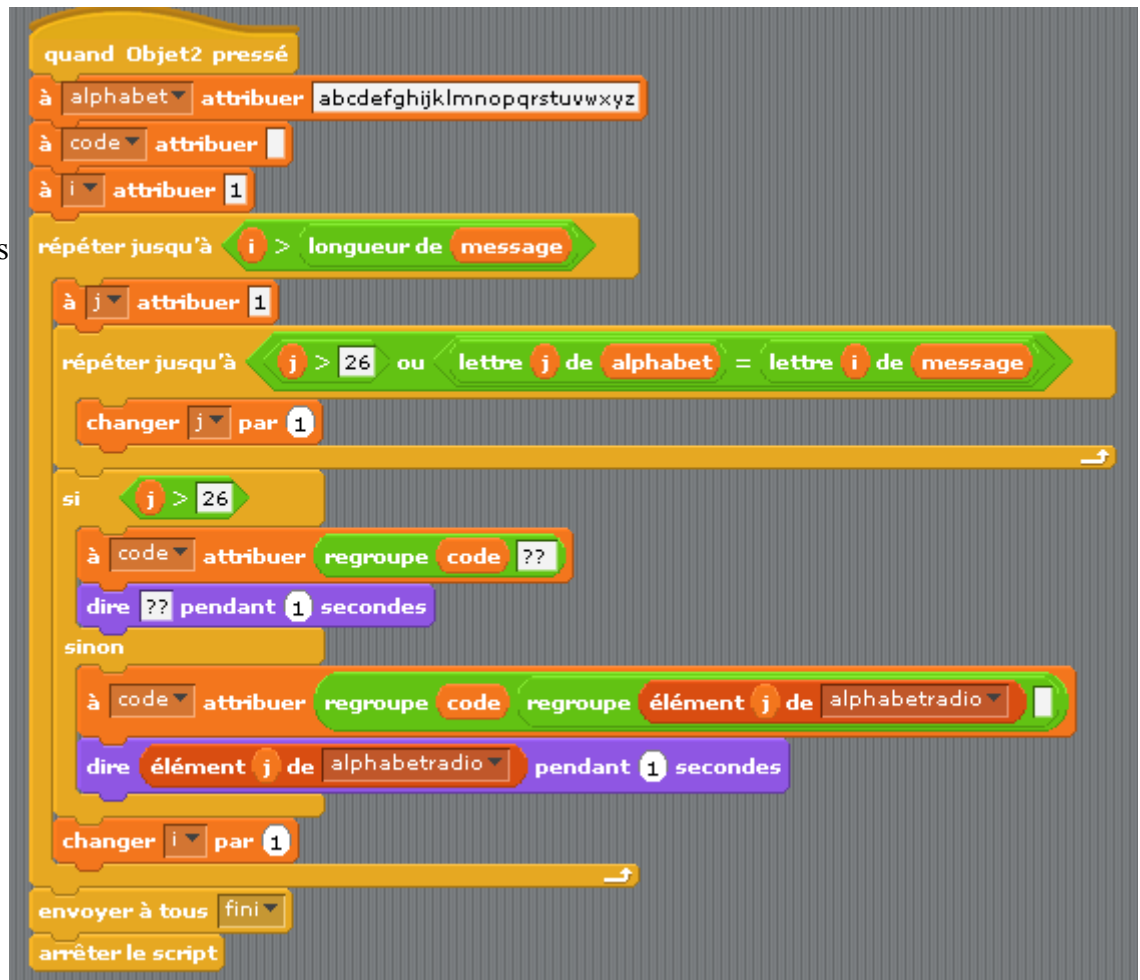
Voici le script principal, qui se déclenche en cliquant sur l'avion.

code est le résultat

on itère sur les lettres du message

la difficulté principale est de déterminer la position dans l'alphabet de chaque lettre !

Pas le choix, on parcourt à chaque fois tout l'alphabet jusqu'à trouver !



Version python

Attention: python v3. Les différences entre python v2 et v3 sont significatives sur tout ce qui concerne les chaînes de caractères.

Ici je ne montre que la partie morse:

On peut stocker l'alphabet dans un dictionnaire. C'est pratique, pour obtenir le codage de la lettre C, il suffit de taper morse['C']. Comme la saisie en texte est plus efficace, on peut se permettre de rentrer l'alphabet morse complet.

```
# codage en Morse et en alphabet radio

# l'aphabet morse
morse = {'A': '.-', 'B': '-...', 'C': '-.-.', 'D': '-..', 'E': '.',
        'F': '..-.', 'G': '--.', 'H': '....', 'I': '...', 'J': '-.-.-',
        'K': '-.-', 'L': '-..', 'M': '--', 'N': '-.', 'O': '---',
        'P': '-.-.', 'Q': '-.-.-', 'R': '.-.', 'S': '...', 'T': '-',
        'U': '...', 'V': '...-', 'W': '-.-', 'X': '-.-.-', 'Y': '-.-.-',
        'Z': '-.-.-', '1': '.----', '2': '..---', '3': '...--', '4': '....-',
        '5': '.....', '6': '-....', '7': '--...', '8': '---..', '9': '---.-',
        '0': '-----', '.': '.-.-.-', ',': '-.-.-.-', '?': '..-.-.-',
        '/': '-.-.-'}

morse_fin_transmission = '.....-'
```

L'alphabet est indexé par les lettres minuscules. On commence par manipuler le message pour supprimer les caractères accentués et tout mettre en majuscule. On utilise pour ça la méthode maketrans de la classe string, qui demande une table de traduction générée par str.maketrans.

```
# on demande le message à coder
message = input("entrez un message à coder: ")
# mise en minuscule, élimination des accents, mise en majuscule
in_tab = "àâäéèëïïöùüûç"
out_tab = "aaaaaaaaiiouuuc"
trans_tab = str.maketrans(in_tab, out_tab)
message = message.lower().translate(trans_tab).upper()
print("message à coder : "+message)
```

On effectue le codage proprement dit en comptant les caractères non codés car n'apparaissant pas dans le dictionnaire (notamment les espaces).

```
morse_out=""
errors=0
for c in message:
    if c in morse:
        morse_out = morse_out + morse[c] + " "
    else: errors=errors+1
print(morse_out)
print(morse_fin_transmission)
print(str(errors)+ " caractères non codés")
```

Et voilà, c'est quand même plus efficace, même si le résultat est plus austère qu'avec scratch !