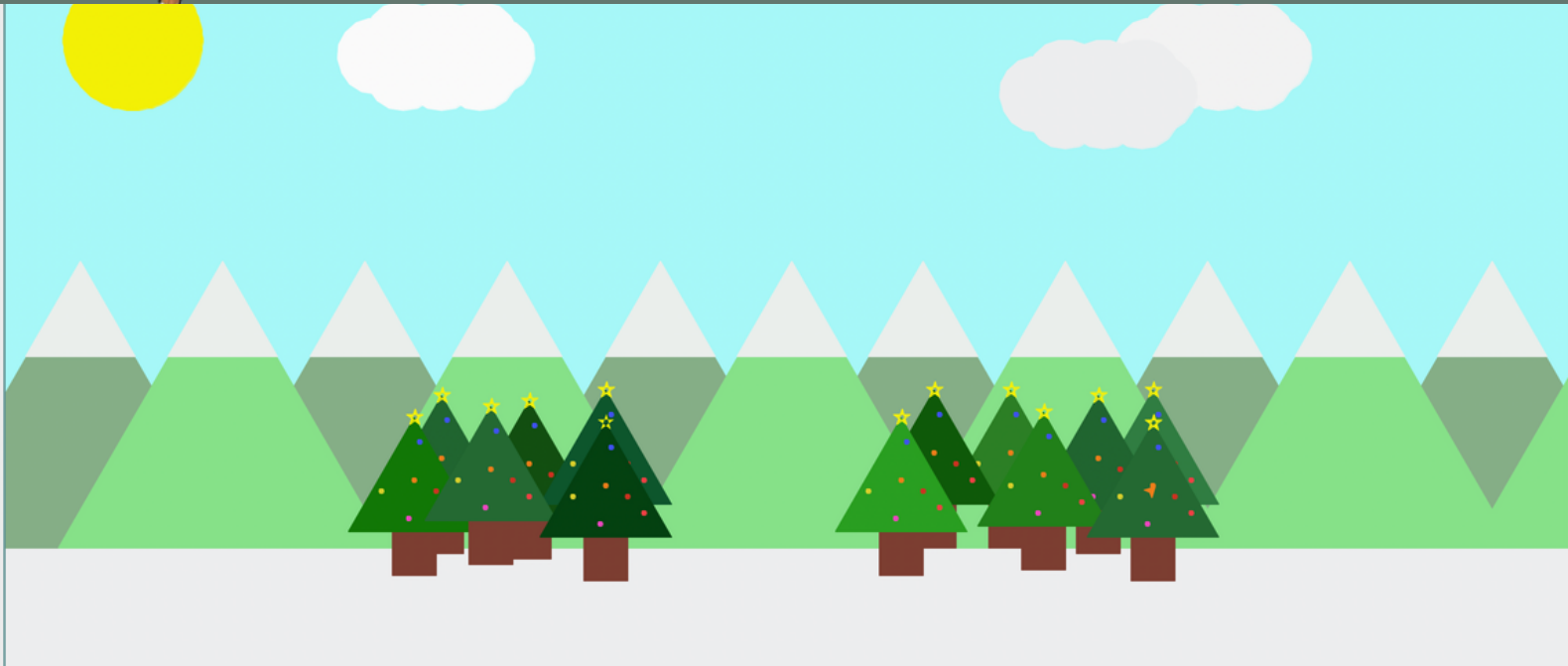


PROJET PROGRAMMATION IMPÉRATIVE

HABIB NAJAR, PICHOFF. PAULINE, GROUPE 1



THÈME GRAPHIQUE

- paysage enneigé
- skis

DESCRIPTION DU PROJET:

Le but de ce projet est d'écrire un jeu des skis avec comme interface graphique le module Turtle.

INSPIRATION:



PLAN DU RAPPORT:

- règles du jeu
- structure du projet
- difficultés rencontrées

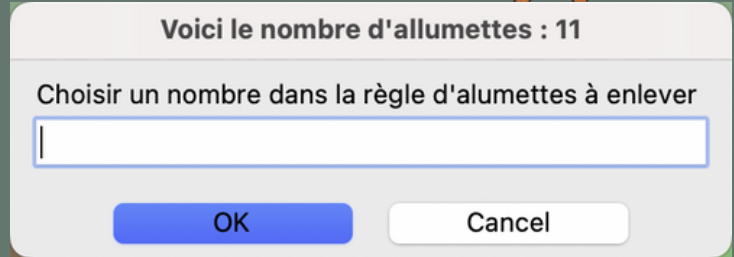


RÈGLE DU JEU

jeu des skis



- Le jeu se joue en solitaire contre l'ordinateur.
- A chaque partie, un nombre de skis est tiré au hasard.
- Le joueur croise des paires de skis, il choisi le nombre de skis a croiser selon la règle donnée.
- La dernière personne a enlever tous les skis, ou la dernière personne à ne plus pouvoir jouer, gagne.



CODE PYTHON

le programme suit l'algorithme suivant:

Tirer un nombre aléatoire d'allumettes

Tant qu'il n'y a pas de perdant :

Afficher les allumettes

Faire jouer le joueur

Si le joueur ne peut pas jouer : le joueur a perdu

Sinon

Afficher les allumettes

Faire jouer l'ordinateur

Si l'ordinateur ne peut pas jouer, l'ordinateur a perdu



STRUCTURE DU PROJET

Ce jeu vidéo a été créé en 3 parties:

- le code de base du jeu
- l'interface graphique
- la fusion de ces codes

Le code de base du jeu:

- plusieurs fonctions:
définirRegle(),
tourJoueur(nombreAlumettes,règle),
tourOrdi(nombreAlumettes,règle),
lancerPartie(),
rejouer(scoreJoueurTemp,scoreOrdiTemp)

L'interface graphique:

- plusieurs fonctions:
polygone(nbcote,longcote,couleur),
montagne(x,y,couleur,x1,x2), herbe(x,y),
bouleDeNoel(x,y,couleur), decoSapin(x,y),
sapin(x,y,couleur1,couleur2), foret(), soleil(), ski(x,y),
paireDeSki(x,y), skiCroixDroit(x,y),
skiCroixGauche(x,y), nuage(x,y,couleur), fond()

Fusion des deux codes de base:

Nous avons utiliser 4 tortues pour réaliser ce jeu, Bob, Jeff, Sam, turtle

```
import turtle
from turtle import *
import random
from turtle import Turtle, Screen
import time
Bob = turtle.Turtle()
Jeff = turtle.Turtle()
Sam = turtle.Turtle()
Sam.hideturtle()
Jeff.hideturtle()
Bob.hideturtle()
turtle.hideturtle()
screen = turtle.Screen()
screen.setup(width=1.0, height=1.0)
```

turtle -> utilisée pour le fond
sam, bob, Jeff -> impression
hide.turtle -> cacher la flèche
lors du dessin

Ce code fusionné contient ainsi toutes les fonctions créées dans les deux fichiers précédents.

Les fonctions ajoutées:

- **impressionFixe(phrase):**
 - * tortue: BOB
 - * utilité: créer un rectangle de la couleur du fond afin d'écrire les scores des joueurs a chaque parties.
- **impression(phrase):**
 - * tortue: JEFF
 - * utilité: écrire la phrase écrite en paramètre
- **impressionOrdi(phrase,x):**
 - * tortue: SAM
 - * utilité: même utilité que impression, mais a un autre emplacement.

Les fonctions modifiées:

tourJoueur(nombreAlumettes,règle)

tourOrdi(nombreAlumettes,règle)

- tortue: SAM
- **regle.copy()** -> création d'une nouvelle liste qui permet à l'ordinateur de ne pas choisir un nombre n'étant pas compris dans la règle

rejouer(scoreJoueurTemp, scoreOrdiTemp)

- tortue: SAM, BOB, JEFF
- utilité: rejouer en indiquant oui ou non. Si oui, le jeu recommence. Si non, tous les skis, les écritures sont supprimées, il ne reste que le fond, le score final est affiché pendant 10 secondes puis la page est fermée (turtle.bye())

lancerPartie()

- tortue: BOB
- utilité: permet de commencer la partie, s'occupe de changer les skis en skis croix.



DIFFICULTÉS RENCONTRÉES

- **GRAND ÉCRAN**

le fond ne s'affichait qu'au centre du grand écran, Nous avons alors regarder la taille de l'écran afin d'y ajouter des éléments toute la longueur et la largeur de celui ci

- **BUG LORSQUE L'ON DEMANDE DE NE PAS REJOUER**

lorsque l'on répondait « non » à la fonction rejouer, le jeu ne s'éteignait pas et il restait que le fond avec les dernières actions réalisées.

la création de la fonction rejouer a permis de régler ce problème. En effet, en utilisant un WHILE, nous pouvons faire 2 actions, lorsque la réponse est OUI une nouvelle partie est relancée, lorsque la réponse est NON les tortues SAM, BOB, JEFF s'effacent c'est-à-dire toutes les écritures, les skis en croix etc, et un message avec le dernier score s'affiche avant la fermeture totale du jeu.

- **LA FONCTION BUTTON**

Si une faute de frappe arrive, on si l'on clique accidentellement sur un mauvais bouton, le jeu est suspendu et nous ne pouvons plus jouer. Ce pourquoi nous avons créer la fonction BUTTON qui nous permet de recommencer le jeu en cas d'accident. Ainsi lorsque le bouton « EN CAS DE BUG CLIQUEZ ICI » est activé, la fonction jeu(regle) recommence.

- **NB CHOISI NON COMPRIS DANS LA RÈGLE, OU NB DE SKIS INF AU MINIMUM DE LA REGLE**

Lorsque le nombrechoisi par le joueur ou l'ordinateur n'est pas compris dans la règle, une erreur s'affiche. Il a fallut modifier les fonctions tourJoueur et tourOrdi en rajoutant des conditions:

- * lorsque le nombre n'est pas dans la règle
- * lorsque le nombre est trop grand comparé au nombre de skis
- * si le nombre de ski est inférieur au nombre minim compris dans la règle,

alors le joueur ne peut pas retirer d'allumettes, c'est alors le dernier joueur à avoir jouer qui gagne

* lorsque l'ordinateur choisi un nombre plus grand que le nombre de skis restants, une nouvelle liste est créée, l'ordinateur choisi alors un nombre dans cette nouvelle liste



RENDU FINAL

Le code se termine par une dernière fonction regroupant toutes les fonctions utiles au bon fonctionnement de ce jeu:

```
def Jeu(regle):  
    Sam.clear()  
    Jeff.clear()  
    Bob.clear()  
    turtle.clear()  
    fond()  
    button()  
    tracer(100)  
    impression("Voici la règle : " + str(regle))  
    resultat = lancerPartie()  
    scoreJoueurTemp, scoreOrdiTemp = resultat  
    impressionFixe("Joueur : "+str(scoreJoueurTemp)+" -  
Ordinateur : "+str(scoreOrdiTemp))  
    rejouer(scoreJoueurTemp, scoreOrdiTemp)
```

+str -> permet d'utiliser des nombres dans une chaîne de caractères, il « transforme », ici par exemple, les caractères compris dans la règle en une chaîne de caractère que l'on peut rajouter à la fin d'une phrase

RÉPARTITIONS DES TÂCHES:

Nous avons tout d'abord réalisé un code simple de notre jeu chacune de notre côté puis nous avons gardé le code de Pichoff Pauline, il était plus simple et plus facile à modifier que celui de Habib Najjar. Habib a ensuite travaillé sur le code en y ajoutant les fonctions énumérées un peu plus haut. Pendant ce temps, Pauline a réalisé l'interface graphique en créant les fonctions utiles à la réalisation du fond. Après s'être concertés et avoir testé plusieurs fois le bon fonctionnement du code et de l'interface graphique. Habib a alors fusionner les deux codes, Puis nous avons testé ensemble les différentes erreurs que nous pourrions avoir et les avons réglé ensemble.