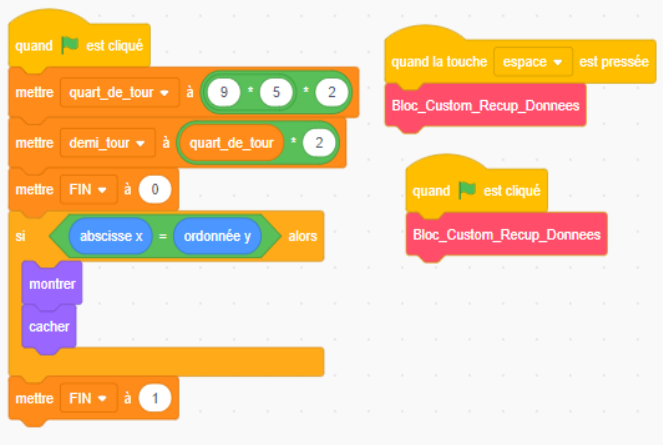
Voici un récapitulatif des tests et du résultat attendu : Mathieu Campan

À noter que je n'ai pas mis les blocs construit automatiquement par souci de clarté, on peut comme ça les comparer avec les TopCodes des tests. S'il vous manque parfois certains de ces blocs, c'est parce qu'ils ne sont pas construit lors d'une boucle infini, le programme ne s'arrêtant pas, on ne peux pas récupérer les données à l'infini. De plus, le bloc de définition de récupération des données n'est pas visible sur Scratch mais est bien là (sinon l'appel au bloc ne ferait rien)

Les blocs construit automatiquement sont ceux visible ci-dessous :
mettre quart_de_tour à 9*5*2
mettre demi_tour à quart_de_tour*2
mettre FIN à 0
mettre FIN à 1
quand espace est pressé
call bloc custom recup donnee
quand drapeau cliqué
call bloc custom recup donnee
+bloc définition recup donnee invisible

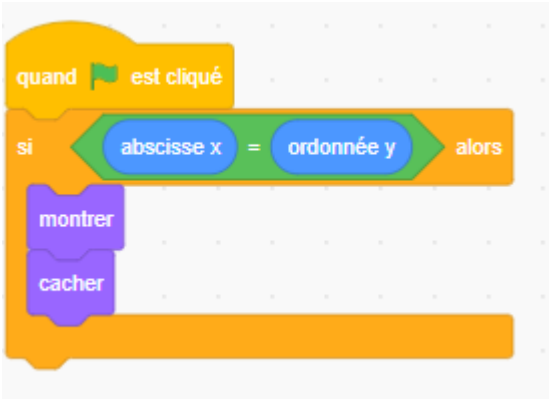
s2_topcodes.png :

version complète



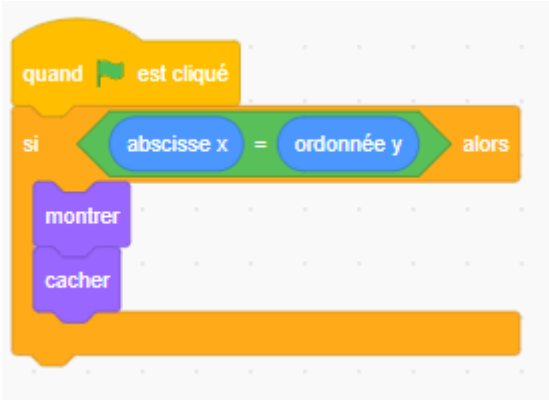
→

version simplifiée



test_S1.png

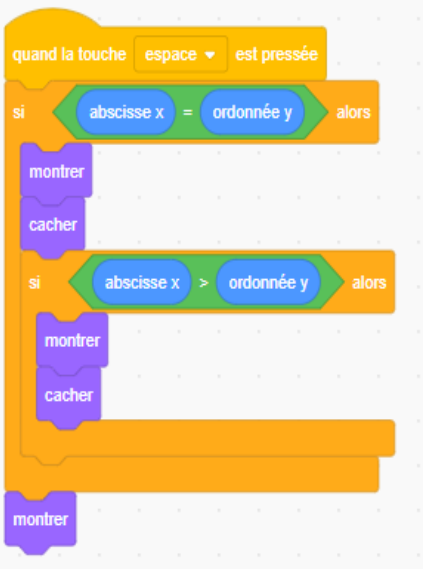
test_S2.png



test_S3.png

test_S4.png : fichier .sb3 créé, pas de résultat sur Scratch
car ligne 5 des TopCodes on fait un « Si x et y », ce qui n'a pas de sens (et ne sera normalement pas possible avec la forme d'encapsulation des blocs qui évite les problèmes de syntaxe)

test_S5.png :
En remplaçant le et par un > le programme est lisible par Scratch

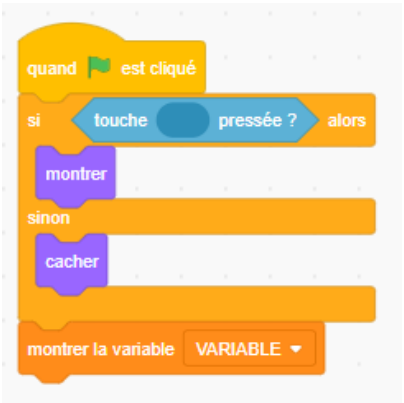


test_S6.png :
Le dernier TopCode est un TopCode montrer la variable, comme il n'y a pas d'autre TopCode disant quelle variable montrer, le programme a construit le code en gardant la valeur par défaut des fields : VARIABLE

test_S7.png :

test_S8.png :

De même, il y a un TopCode showvariable sans variable après, le programme lève crée le fichier Scratch, mais il y a un trou dans le code



On voit avec le test_S7 que le programme laisse passer des inputs n'ayant pas de sens comme le a ici, ou le var1 non initialisé mais toujours syntaxiquement juste, on pourrait rajouter des contrôles pour l'empêcher, mais il peut être bien de le laisser comme ça pour montrer à l'élève que ce n'est pas un problème de manque de bloc, mais un problème sur les blocs utilisés, ce qui peut être plus compréhensible qu'une exception avec un nom à rallonge, pour l'élève et pour le professeur.

test_S9 :

le test_S9 est très pratique pour visualiser l'utilité
des listes lors de la récupération de données.
Vous devriez voir la liste temp se modifier avec les déplacements du chat, et avoir à la fin, la liste export contenant les différentes valeurs de la liste temp :
Chaque ligne de liste_export correspond aux éléments de liste_temp à un instant donné

test_S10_1.png & test_S10_2.png : On peut remarquer deux choses avec ces tests :

- L'utilisation d'un Fin de boucle a complètement changé le programme.
- Comme ces tests comportent une boucle infini, il n'y a plus d'appels au bloc de récupération de donné (il n'a d'ailleurs pas été créé), ni la mise de Fin à 1 en fin de code. Par contre, l'initialisation des variables est toujours là.

test_S11.png :
J’ai cette fois ci laissé les blocs créés automatiquement, pour justement voir que l’on rajoute avant chaque stop all un set Fin 1
Si on voulait s’assurer que tout ai bien été sauvegarder, on pourrait rajouter un wait 1 avant et après chaque set fin 1, mais cela deviendrait vite lourd et illisible.