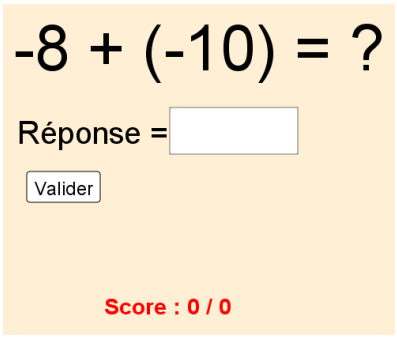
Tutoriel pour créer un exerciseur avec des valeurs aléatoires avec GeoGebra

Cycle 4

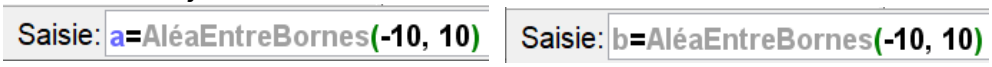
Auteur : PETIT Raphaël © novembre 20.)



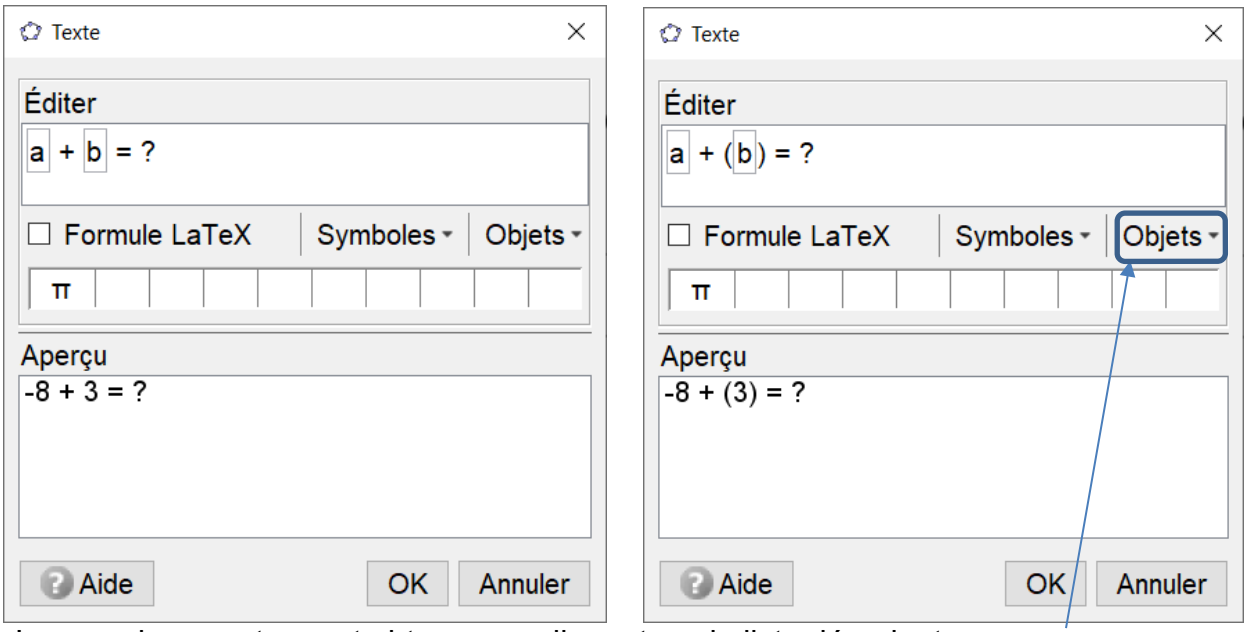
But :
Créer un exerciseur sur les sommes de nombres relatifs de valeurs absolues inférieures ou égales à 10.

Ce tutoriel est fait avec l'application GeoGebra Classique 5 en local.

Créer des objets `a` et `b`, nombres entiers choisis entre -10 et 10 :



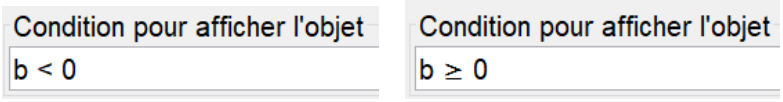
En fonction du signe de `b`, l'affichage de l'addition `a+b` utilisera ou non des parenthèses, créer deux textes :



Les nombres `a` et `b` sont obtenus en cliquant sur la liste déroulante « `Objets` ».

Chacun de ces deux objets devra être affiché ou non en fonction du signe de `b` :

> faire un clic droit sur les textes et choisir « `Propriétés...` ».
> Dans l'onglet « `Avancé` », choisir la condition pour afficher l'objet correspondante.

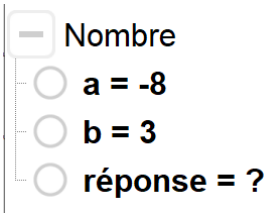


(le symbole ≥ s'obtient en écrivant >=.)

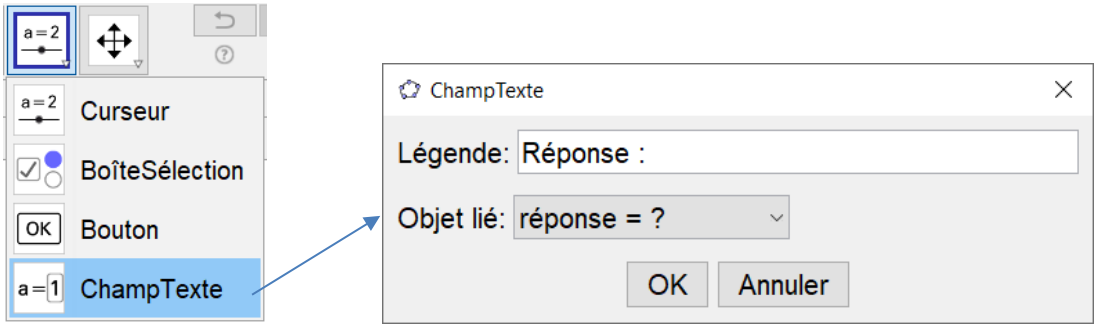
Ces deux objets s'affichant à tour de rôle, nous allons les placer au même endroit : créer un point `A` dans la fenêtre graphique puis, dans les propriétés de ces textes, dans l'onglet « `Position` », choisir le point `A` comme point de départ. Enlever l'affichage du point `A`.

Nous allons maintenant créer un objet nombre dans lequel nous enregistrerons la réponse donnée par l'utilisateur :

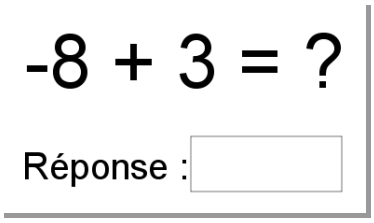
« `NaN` » signifie « Not a Number ». L'objet nombre `réponse` sera créé mais pas encore défini, comme on peut le voir dans la fenêtre algèbre :



Créer, dans la fenêtre graphique, un objet ChampTexte permettant d'accueillir la proposition de l'utilisateur :



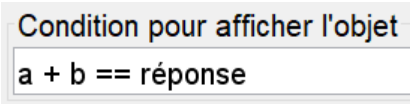
Voici l'aperçu obtenu à ce moment :



(la taille de ce ChampTexte peut être réglée dans ses `propriétés`, onglet `Style`.)

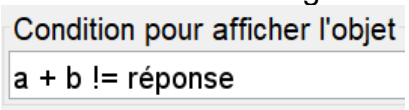
Il ne reste plus qu'à créer :
- le bouton pour valider la réponse,
- le bouton pour réinitialiser
- et proposer une autre question ainsi que les textes « `Bravo !` » et « `Erreur…` ».

Créer les deux objets textes « `Bravo !` » et « `Erreur…` ».
Le premier d'entre eux doit être affiché si la réponse est correcte, on va donc modifier sa condition d'affichage en conséquence :



(ici, la somme des contenus des objets `a` et `b` est comparée au contenu de l'objet `réponse`)

La condition d'affichage du deuxième sera la suivante :



(`!=` signifie « négation de = », c'est à dire « différent »)

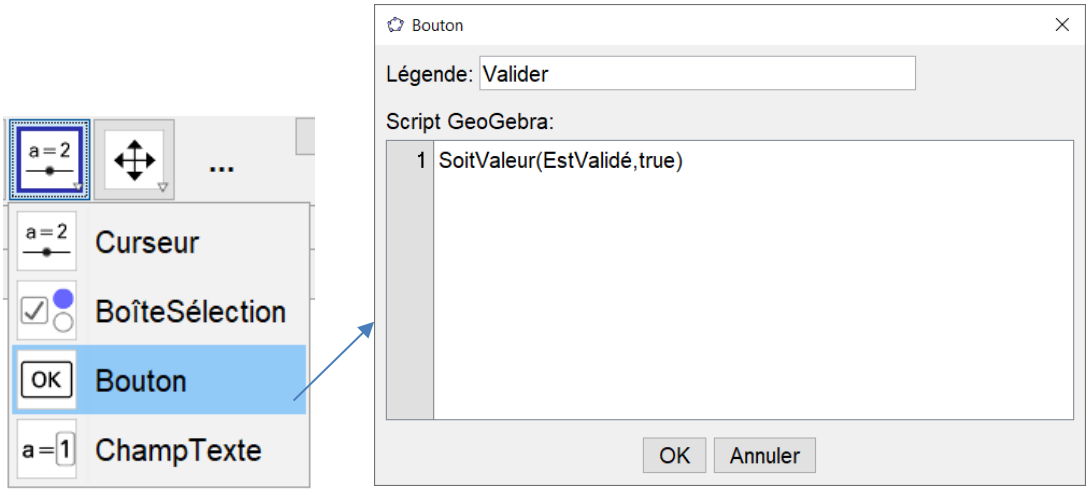
De la même façon que pour les affichages de l'opération, créer un nouveau point et positionner les deux textes sur ce même point.

Ces deux textes ne doivent s'afficher que si l'utilisateur a validé sa proposition de réponse, ie s'il a bien appuyé sur le bouton « Valider ». Il faut donc créer un booléen qui sera faux par défaut et qui deviendra vrai en cas de clic sur le bouton « `Valider` » :

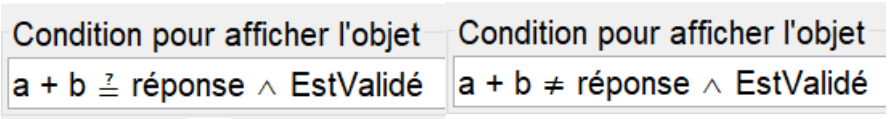
Taper la séquence suivante dans le champ de saisie : 

pour créer et initialiser l'objet booléen à false (faux).

Créer maintenant le bouton « `Valider` » avec un script qui modifiera la valeur du booléen « `Valid` » :



Il faut maintenant ajouter une condition d'affichage aux textes « `Bravo !` » et « `Erreur…` » :

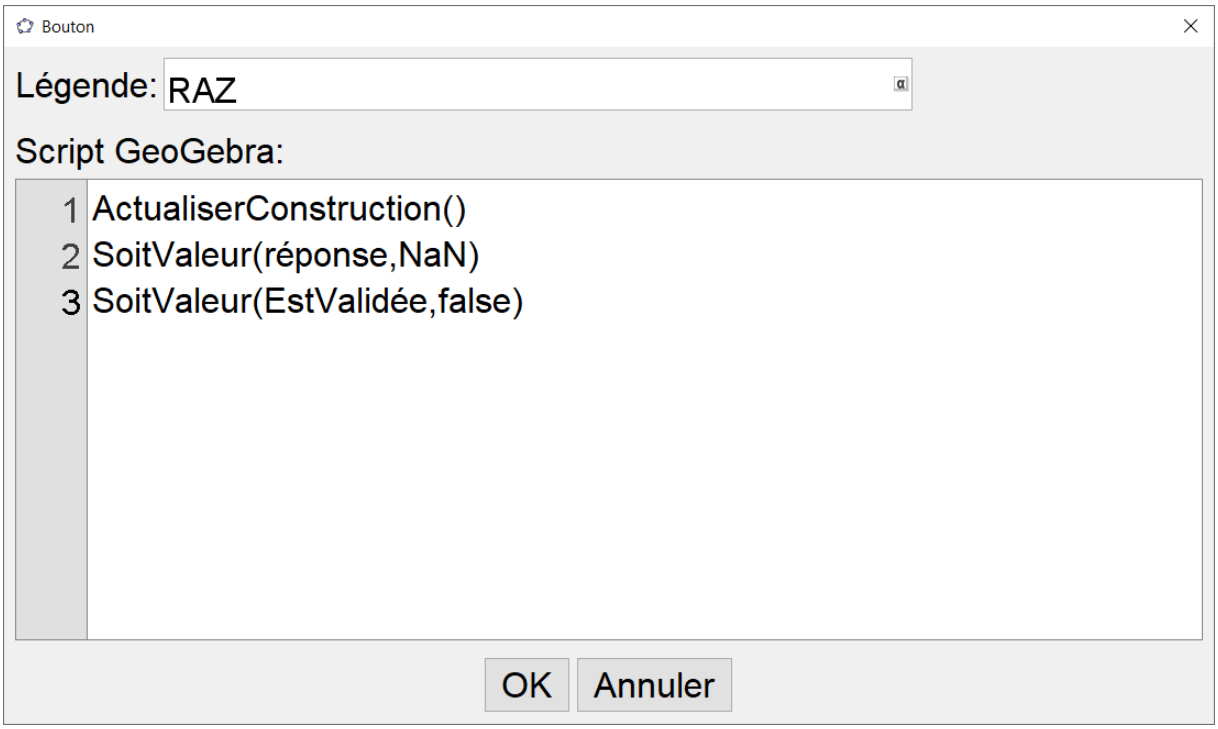


Le caractère ≟ s'obtient en tapant « == » (double symbole « = »)

Le symbole « ≠ » s'obtient en tapant « != ».

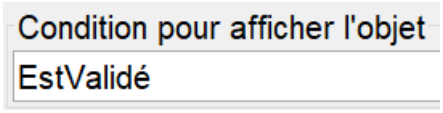
Le symbole « ∧ » correspond au ET (and) logique. Il s'obtient en tapant « && ».

Créer le bouton « `Nouvelle question` » qui permettra :
- de réinitialiser les valeurs de `a` et de `b`,
- d'effacer la précédente réponse en l'écrasant par la valeur NaN
- et le booléen « `EstValidé` » à false, ce qui permet de rendre la réponse non validée.



(La commande `ActualiserConstruction()` permet de recalculer tous les objets définis aléatoirement.)

Pour éviter que l'utilisateur n'abuse de ce bouton pour trouver une question facile, on ajoutera une condition d'affichage à ce bouton afin qu'il ne soit visible que lorsque l'utilisateur aura validé par le bouton `Valider` :



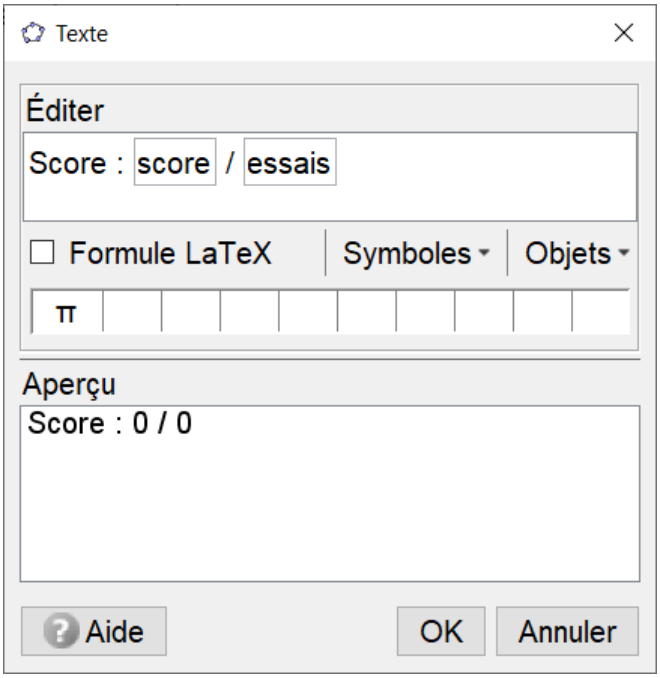
Pour aller plus loin, on peut ajouter deux compteurs, l'un pour un score, l'autre pour le nombre de tentatives de bonnes réponses.

Créer deux objets nombres « `score` » et « `essais` » initialisés à 0.

Dans le script du bouton « `Valider` », ajouter les commandes suivantes :

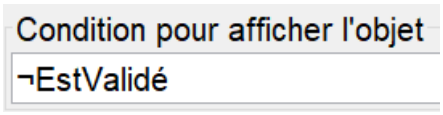
```
Si(a+b==réponse,SoitValeur(score,score+1))
SoitValeur(essais,essais+1)
```

Créer un texte qui affichera le score :



Si on s'arrête là, un problème apparaît : une fois que l'utilisateur a trouvé une bonne réponse, il peut cliquer autant de fois qu'il veut sur le bouton « `Valider` » et ainsi augmenter son score sans effort !

Pour éviter cet écueil, on peut programmer ce bouton pour qu'il disparaisse quand on a cliqué dessus et qu'il réapparaisse quand une nouvelle question est posée :



Ce qui est équivalent à tester si le booleen valid est faux : «¬EstValidé » s'obtient en tapant «!EstValidé » et renvoie la négation de « EstValidé », donc faux si EstValidé est vrai et vrai si EstValidé est faux.