

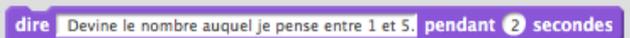
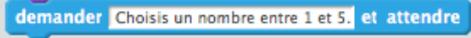
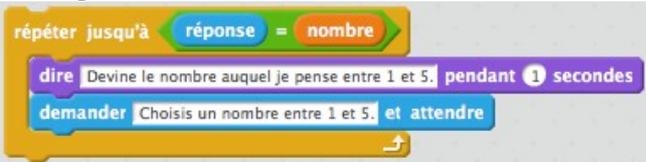
CREATION DU JEU

« Plus Grand Plus Petit »

Problématique de départ

Le projet proposé aux élèves serait de faire choisir un nombre aléatoire entre 1 et 100 à l'ordinateur puis de trouver ce nombre avec des indications du type "Plus Grand" ou "Plus petit". Un compteur du nombre d'essais pour trouver la bonne réponse permettra d'enclencher la compréhension du processus de dichotomie. Par la suite, l'enjeu pourrait être d'inverser les rôles (l'ordinateur chercherait le nombre auquel vous pensez). Pour les plus rapides, mettre en œuvre un dialogue entre les deux personnages créés peut être un objectif supplémentaire.

Sous-problèmes engendrés par la problématique :

Sous-problèmes nécessaires à la construction du jeu.	
Choisir un nombre au hasard entre 1 et 5 et affecter sa valeur à la variable nombre.	<p>On crée une variable nombre : </p> <p>On affecte à cette variable un nombre entier choisi au hasard entre 1 et 5 :</p> 
Faire en sorte que le lutin demande à l'utilisateur d'effectuer une proposition.	<p>Le lutin peut afficher un message d'introduction durant 2 secondes (facultatif) puis poser effectivement la question :</p>  
Tester l'égalité de la réponse fournie avec le nombre choisi initialement.	<p>On utilise un bloc d'égalité pour comparer la variable nombre et la réponse à la question.</p>  <p>Cela donne :</p> 
Tester la réponse fournie et poursuivre le processus jusqu'à ce que l'utilisateur trouve la bonne réponse.	<p>On peut traiter la réponse donnée avec un bloc REPETER JUSQU'A... :</p> 

On peut donc imaginer un programme de ce type là qui permet de répondre simplement à la problématique posée en posant la question au joueur tant qu'il ne trouve pas. Ce programme se retrouve dans le fichier :

PGPP-trouve-mon-nombre-1.sb2



A ce stade, le jeu est en mesure de fonctionner pour un nombre aléatoire entre deux entiers proches.

CREATION DU JEU

« Plus Grand Plus Petit »

On peut suggérer aux élèves de générer un nombre aléatoire entre 1 et 100 pour induire la nécessité de donner une indication du type "Plus grand"- "Plus petit" puis une dichotomie lors de l'utilisation par les élèves. Ceci permettra d'améliorer le projet.

Sous-problèmes additionnels permettant d'améliorer le jeu.

<p>Donner une indication de type "Plus grand" ou "Plus petit" en fonction de la réponse de l'utilisateur.</p> <p><i>Cela permet de générer des nombres aléatoires entre 1 et 100 et faire jouer les élèves pour découvrir la notion de dichotomie.</i></p>	<p>On utilise deux blocs : SI...ALORS...SINON ou SI...ALORS...</p>  <p>La deuxième conditionnelle est obligatoire. En effet, il faut éviter que le lutin dise "plus petit" quand l'utilisateur donne la bonne réponse (on peut laisser les élèves expérimenter avec un seul bloc conditionnel et les laisser se corriger). (Voir PGPP-trouve-mon-nombre-2.sb2)</p>
<p>Compter le nombre d'essais successifs avec une variable "essais".</p>	<p>Au début du programme on remet le nombre d'essai à 0 et on affiche la variable :</p>  <p>Il reste à augmenter le nombre d'essais quand la réponse est incorrecte.</p> 
<p>Féliciter le joueur s'il trouve rapidement. Cela pose la question de la stratégie optimale (dichotomie) et du nombre d'étapes nécessaires dans le pire des cas.</p>	<p>L'utilisation d'un bloc conditionnel en fin d'algorithme peut prendre cette forme :</p> 

Voici un exemple de programme obtenu à ce stade de sa conception :

PGPP-trouve-mon-nombre-3.sb2

essais 0



```

quand pressé
  dire Devine le nombre auquel je pense entre 1 et 100 pendant 2 secondes
  mettre essais à 0
  mettre nombre à nombre aléatoire entre 1 et 100
  répéter jusqu'à réponse = nombre
  demander Choisis un nombre entre 1 et 100. et attendre
  si nombre > réponse alors
    dire Plus grand pendant 2 secondes
  sinon
    si nombre < réponse alors
      dire Plus petit pendant 2 secondes
    ajouter à essais 1
    montrer la variable essais
  si essais < 7 alors
    dire Bravo. Tu as trouvé rapidement.
  sinon
    dire Bravo mais tu peux faire plus vite.
  
```

CREATION DU JEU

« Plus Grand Plus Petit »

Problème additionnel n°1 : on inverse les rôles ! L'ordinateur cherche un nombre auquel pense l'élève...

Créer un nouveau fichier et choisir un autre lutin (dans ceux prédéfinis). L'orienter dans l'autre sens pour anticiper un futur dialogue.

Cliquer sur le "I" en haut à gauche du lutin.



L'objectif peut être de commencer par trouver un nombre entre 1 et 10 auquel pense l'élève. Celui-ci répondra par "oui" ou "non".

On utilise un bloc : REPETER JUSQU'A...

La variable "proposition" est un nombre aléatoire entre 1 et 10.



Remarques :

- toute réponse, autre que "oui" fera durer le jeu.
 - L'ordinateur peut proposer plusieurs fois le même nombre !
- (Voir PGPP-je-cherche-1.sb2)

On améliore l'algorithme précédent pour que l'ordinateur ne propose pas deux fois le même nombre

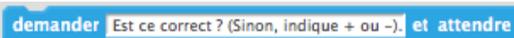
On utilise deux variables "minimum" et "maximum", initialisées respectivement à 1 et 10 au début :



On propose toujours un nombre "proposition" choisi aléatoirement entre "minimum" et "maximum" :



L'utilisateur répond "oui", "+" ou "-" :



Si la bonne réponse n'a pas été trouvée, l'une des variables "minimum" ou "maximum" est changée en fonction de la réponse fournie par l'élève :



(Voir PGPP-je-cherche-2.sb2)

CREATION DU JEU

« Plus Grand Plus Petit »

Penser à "couper en deux" (dichotomie) pour une stratégie optimale

La proposition n'est plus choisie aléatoirement entre "minimum" et "maximum" : c'est la valeur moyenne arrondie de ces deux nombres.

mettre proposition à arrondi de $\frac{\text{minimum} + \text{maximum}}{2}$

Voici un exemple de programme obtenu à ce stade de sa conception (on peut alors chercher efficacement un nombre entre 1 et 100) : **PGPP-je-cherche-3.sb2**

```
quand flag pressé
  dire "Pense à un nombre entre 1 et 100. Je vais le trouver" pendant 3 secondes
  mettre minimum à 1
  mettre maximum à 100
  répéter jusqu'à réponse = oui
    mettre proposition à arrondi de (minimum + maximum) / 2
    dire "regroupe Je te propose : proposition" pendant 2 secondes
    demander "Est ce correct ? (Sinon, indique + ou -)." et attendre
    si réponse = + alors
      mettre minimum à proposition + 1
    si réponse = - alors
      mettre maximum à proposition - 1
  dire "J'ai trouvé !!"
```

CREATION DU JEU

« Plus Grand Plus Petit »

Problème additionnel n°2 : on fait dialoguer les deux personnages précédents !

Les deux lutins et leur algorithme respectif doivent être réunis dans le même fichier.

Dans l'un des fichiers précédents (ici PGPP-je-cherche-3.sb2), cliquer droit sur le lutin puis "l'enregistrer localement comme un fichier".



Ouvrir le fichier contenant l'autre lutin "PGPP-trouve-mon-nombre-3.sb2". L'enregistrer sous "PGPP-dialogue.sb2" puis "importer un lutin depuis un fichier" :



Vos deux lutins (et leurs algorithmes) sont alors présents dans le même fichier scratch, que vous penserez à enregistrer.

L'objectif est alors de faire communiquer les lutins entre eux.

L'un d'eux doit commencer "quand le drapeau est pressé", l'autre doit attendre un message du premier. On utilise le bloc "envoyer un message" et l'évènement "Quand je reçois un message" :



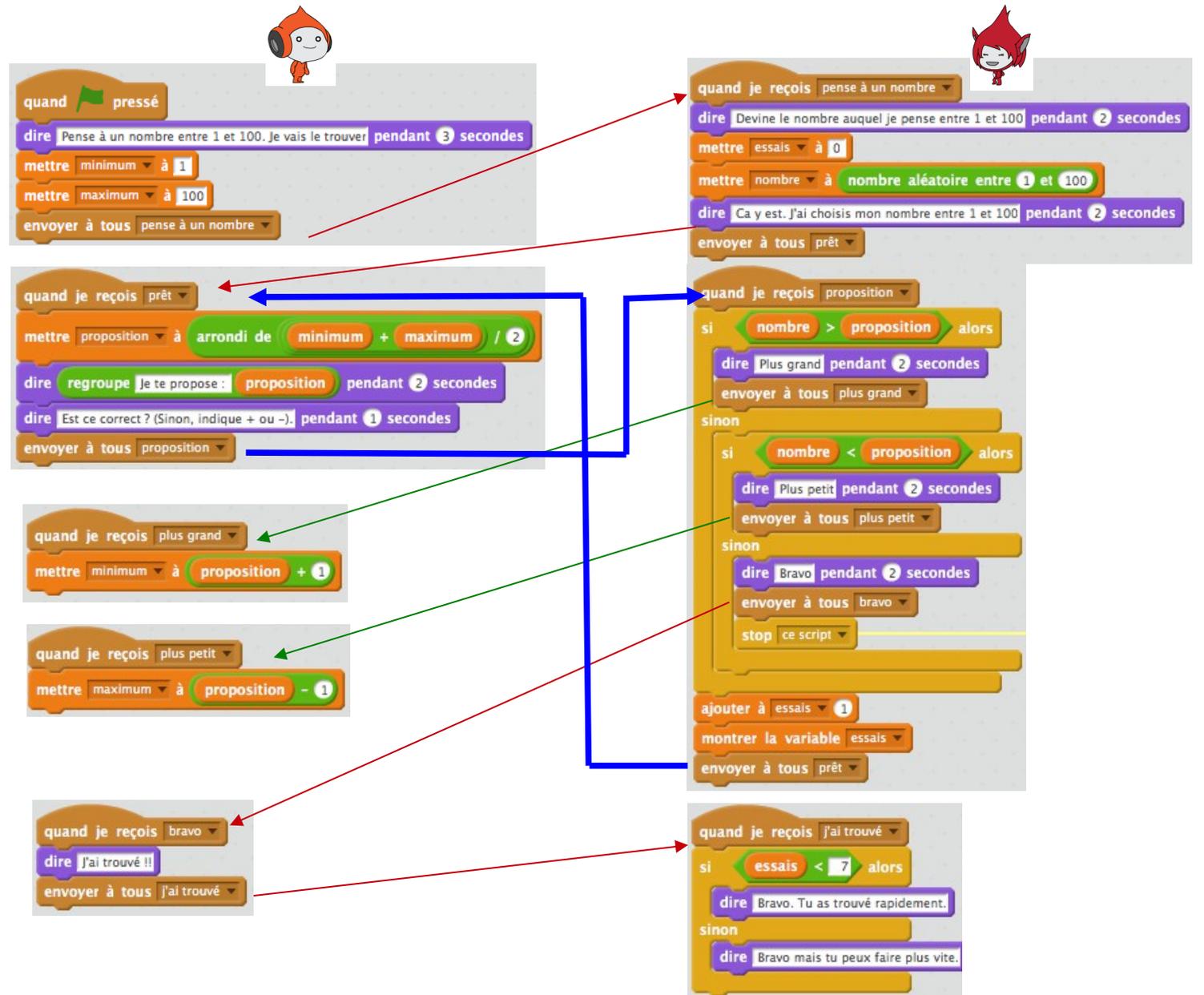
Il faut alors mettre en place un dialogue entre les deux personnages en "déconstruisant" les algorithmes existants...



On imagine les différents messages échangés. Globalement, on reprend l'ensemble des blocs précédemment créés : tout affichage à l'écran d'un message par un lutin est doublé d'un envoi de message à l'autre lutin.

Remarques :

- Les flèches rouges font avancer les lutins à une étape ultérieure de leur dialogue.
- Certains blocs deviennent alors inutiles ("répéter jusqu'à ...", "demander...") : on les sort de l'algorithme puis on les supprime à l'aide d'un clic droit.
- La boucle "répéter jusqu'à" est remplacée par le dialogue symbolisé par les flèches bleues.
- Les flèches vertes ne passent pas la main mais permettent à l'autre lutin de changer "minimum" ou "maximum"
- Le "Envoyer à tous bravo" est suivi d'un bloc "STOP" pour mettre fin à la boucle bleue (recherche dichotomique).



Voici un exemple de programme obtenu à ce stade de sa conception : **PGPP-dialogue.sb2**

Remarque : le dernier bloc "Bravo mais tu aurais pu faire plus vite" ne s'affiche plus jamais puisqu'une stratégie optimale a été mise en place.