

Extrait du Les nouvelles technologies pour l'enseignement des mathématiques

<http://revue.sesamath.net/spip.php?article261>

De l'algorithmique pour faire des mathématiques

- N°18 - Janvier 2010 - Les articles hors dossier -

Date de mise en ligne : vendredi 29 janvier 2010

Les nouvelles technologies pour l'enseignement des mathématiques

Sommaire

- [Le déroulement effectif de cette séquence](#)
- [Bilans et perspectives](#)
- [Conclusion](#)

Ma problématique : faire de l'algorithmique, en classe de 2nde, afin de répondre à la demande institutionnelle (le programme de mathématiques !) tout en faisant progresser le cours « habituel » de mathématiques.

Résumé : en utilisant les possibilités algorithmiques et graphiques du logiciel [Algobox](#), j'ai bâti la séquence décrite dans cet article qui m'a permis de faire travailler mes élèves de 2nde sur l'interprétation et la modification d'algorithmes tout en rencontrant plusieurs notions mathématiques :

- statistiques : fréquences.
- probabilités : probabilité comme limite des fréquences, événement impossible, réunion et intersection d'événements.
- logique : ET / OU
- analyse : courbe représentative d'une fonction, aire sous la courbe représentative d'une fonction.

Les travaux proposés sont inspirés par la [Méthode de Monte-Carlo](#) pour calculer une aire.

Principe général : Les algorithmes utilisés reposent tous sur les idées suivantes :

- calcul aléatoire des coordonnées (comprises entre 0 et 1) d'un nombre déterminé de points.
- les points ne sont tracés que lorsque les coordonnées vérifient une condition (par exemple : $y \leq x^2$ ou $y \geq \sqrt{1-x^2}$ ou encore $y = \sqrt{1-x^2}$).
- calcul de la proportion de points tracés par rapport au nombre total de points calculés.

Modélisation : Les algorithmes utilisés fournissent des sorties graphiques et des fréquences statistiques qui doivent être modélisées afin d'interpréter certains phénomènes observés.

La modélisation probabiliste et son interprétation géométrique doivent en retour permettre de créer des algorithmes visant à obtenir des sorties graphiques imposées tout en prévoyant des fréquences liées aux sorties produites.

Analyse à priori

- Dans le programme de la classe de 2nde, il est dit que « la démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique », reprenant ainsi les préconisations de la « [commission Kahane](#) » sur « l'enseignement des sciences mathématiques ». J'ai donc décidé d'utiliser des algorithmes très directement liés à des notions mathématiques de base ou plus élaborées (voire hors du champ de la classe de 2nde).
- Ce qui fait la force des algorithmes utilisés en informatique c'est très souvent, pour ne pas dire toujours, le traitement répétitif d'instructions (boucles Pour ou Tant que) ainsi que la possibilité de faire des choix (structures

SIALORS). J'ai donc décidé de mettre en oeuvre des algorithmes où la répétition fait sens, c'est à dire que les répétitions (parfois très nombreuses) sont un constituant essentiel de la résolution du problème posé.

- Dans le programme de la classe de 2nde, il est indiqué que parmi les divers travaux à mener, il faut entraîner les élèves « à interpréter des algorithmes plus complexes. ». J'ai donc décidé de faire lire et modifier des algorithmes proposés à mes élèves par mes soins sur des sujets mathématiques divers : « fonctions, géométrie, statistiques et probabilité, logique ».
- Une des difficultés de la mise au point d'un algorithme c'est le contrôle (fait-il ce que je souhaite ?). Afin d'inciter à ce contrôle et afin de le faciliter, j'ai décidé d'utiliser des logiciels permettant une programmation d'algorithmes dont les sorties sont graphiques. Parmi toutes les possibilités évoquées dans divers documents et sites web, j'ai choisi d'utiliser essentiellement [Algobox](#) à cause de sa clarté, du nombre réduit d'instructions élémentaires fournies et de la facilité avec laquelle on peut réaliser des sorties graphiques.
J'ai aussi mené en début d'année des travaux de géométrie avec [Scratch](#) qui m'a rappelé la tortue Logo et mes lectures de [Seymour Papert](#) ("Jaillissement de l'esprit. Ordinateurs et apprentissage", Flammarion, 1981)
- Les situations proposées en mathématiques doivent couvrir des sujets divers, cependant le temps imparti à l'algorithmique ne peut être extensible. J'ai décidé de choisir un sujet qui permette des ouvertures vers plusieurs domaines mathématiques : « fonctions, géométrie, statistiques et probabilité, logique ».

Mes choix

Les travaux sont inspirés de la [Méthode Monte-Carlo](#) permettant un calcul probabiliste des aires. Le calcul aléatoire de points repérés placés dans un carré de côté 1 et leur tracé conditionné par un lien entre leur abscisse et leur ordonnée (inéquation ou équation) permet de travailler :

- en statistiques sur la fréquence de tracé des points.
- en probabilité : la probabilité de tracé d'un point comme limite de la fréquence du tracé.
- en probabilité toujours : interprétation géométrique de la probabilité de tracé d'un point.
- en probabilité encore : notion d'événement impossible (lorsque la condition de tracé du point est sous forme d'une égalité).
- en probabilité toujours : notion de réunion et d'intersection d'évènements (lorsque la condition est écrite avec un ET ou bien avec un OU).

Le logiciel choisi est [Algobox](#) . Les élèves n'ont pas encore utilisé ce logiciel et vont le découvrir dans cette séquence. Les raisons de ce choix sont les suivantes :

- obtention facile de tracés dans un repère en plus des sorties textuelles.
- syntaxe de programmation facile à comprendre.
- l'essentiel des structures simples est disponible.
- l'algorithme est clairement présenté.
- facilité supposée de son utilisation.

Informations pratiques : pour certaines étapes vous pourrez tester l'algorithme concerné grâce à une page web qui a été générée avec Algobox. Cependant la sortie graphique ne fonctionne qu'avec Firefox. La sortie texte fonctionne avec Internet Explorer et Firefox.

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8).

Le déroulement effectif de cette séquence

Les documents élèves

La 1ère fiche élève (utilisée lors de la 1ère séance de 1h30 pour la 1ère page et lors de la 2nde séance pour la 2nde page) :



Fiché élève 1ère partie

La 2nde fiche élève (utilisée lors de la 1ère séance de 1h30 pour la 1ère page et lors de la 2nde séance pour la 2nde page) :



Fiché élève 2nde partie

Les algorithmes fournis aux élèves (ex1 pour la fiche 1 et TP1 pour la fiche 2)



Les algorithmes fournis aux élèves

Chaque étape ci-dessous correspond à une question des documents élèves.

1ère étape : lire et comprendre ce que fait un algorithme donné. L'algorithme calcule aléatoirement les coordonnées de 100 points choisies entre 0 et 1. Il affiche les coordonnées calculées (sortie « texte ») et le point correspondant (sortie « graphique »). Les points affichés sont dans un carré de côté 1.

Pour en savoir plus..

L'algorithme est fourni aux élèves. (Voir le document joint à cet article)

Activité de l'élève

Consignes :

Lancer Algobox et ouvrir le fichier fournit (ex1) . Tester l'algorithme et exprimer en langage courant ce qu'il fait.

Commentaires :

Il n'y a que peu de difficultés. Il faut cependant arriver à une formulation précise comme :

- *l'algorithme calcule aléatoirement les coordonnées de 100 points choisies entre 0 et 1.*
- *Il affiche ces couples de coordonnées (sortie « texte ») et le point correspondant (sortie « graphique »).*

Pour dire précisément ce que fait l'algorithme, il ne suffit pas regarder les sorties texte et graphique, il faut aussi lire cet algorithme pour trouver le nombre de points affichés et aussi déterminer le choix aléatoire des coordonnées entre 0 et 1.

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

L'algorithme :

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  POUR N ALLANT_DE 1 A 100
7  DEBUT_POUR
8  x PREND_LA_VALEUR random()
9  y PREND_LA_VALEUR random()
10 AFFICHER x
11 AFFICHER " ;"
12 AFFICHER y
13 TRACER_POINT (x,y)
14 FIN_POUR
15 FIN_ALGORITHME
```

Voici la sortie graphique obtenue :



- Sortie graphique : les points sont tous dans un carré de côté 1 (on peut regretter qu'AlgoBox ne gradue pas les axes).
- Sortie texte (Résultats) : on y voit les coordonnées des points affichés.

++++Intentions algorithmiques

L'algorithme calcule les coordonnées de 100 points choisis aléatoirement entre 0 et 1. Il affiche ces couples de

coordonnées (sortie texte) et le point correspondant (sortie « graphique »). Nous en avons profité pour repérer l'organisation de l'algorithme (variables, instruction d'affectation, d'affichage, répétition d'instruction)

++++Intentions mathématiques

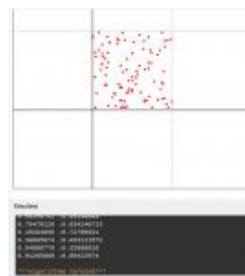
L'aléatoire est présent dans le calcul des coordonnées (random) et doit être reconnu par les élèves grâce au placement aléatoire des points dans la sortie graphique.

++++En images

Algorithme utilisé

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  POUR N ALLANT_DE 1 A 100
7  DEBUT_POUR
8  x PREND_LA_VALEUR random()
9  y PREND_LA_VALEUR random()
10 AFFICHER x
11 AFFICHER " ;"
12 AFFICHER y
13 TRACER_POINT (x,y)
14 FIN_POUR
15 FIN_ALGORITHME
```

Voici la sortie graphique obtenue :



- Sortie graphique : les points sont tous dans un carré de côté 1 (on peut regretter qu'AlgoBox ne gradue pas les axes).
- Sortie texte (Résultats) : on y voit les coordonnées des points affichés.

2ème étape : insérer une instruction d'affichage qui permet de mieux visualiser les « passages » dans la boucle POUR.

Pour en savoir plus ..

Le travail proposé est ici très simple.

Activité de l'élève

Consignes :

Modifier l'algorithme afin que l'on voit à quelle étape de la boucle POUR correspond le couple de coordonnées affiché.

Commentaires : Le travail est effectué en commun car les élèves découvrent Algobox. J'en profite pour indiquer quelques manipulations de base (introduire une ligne vide avant d'y placer une instruction). On doit obtenir ceci :

L'affichage indiquant l'étape du calcul de la boucle Pour (valeur de N) est placée avant l'affichage des coordonnées correspondantes. Cela nécessite un repérage du point d'insertion de l'instruction correspondante dans l'algorithme.

++++Intentions algorithmiques

Pour réaliser cette modification , il faut introduire une nouvelle instruction d'affichage en la plaçant au « bon endroit ». Il est donc nécessaire de s'approprier la structure de l'algorithme.

++++En images

Algorithme modifié (voir ligne 10)

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  POUR N ALLANT_DE 1 A 100
7  DEBUT_POUR
8  x PREND_LA_VALEUR random()
9  y PREND_LA_VALEUR random()
10 AFFICHER N
11 AFFICHER x
12 AFFICHER " ;"
13 AFFICHER y
14 TRACER_POINT (x,y)
15 FIN_POUR
16 FIN_ALGORITHME
```

Seule la sortie texte est modifiée :

```
Résultats
98
0.87994385 ;0.081573486
99
0.74105835 ;0.26815796
100
0.23138428 ;0.92678833

***Algorithme terminé***
```

3ème étape : *introduire une structure conditionnelle SI afin que le tracé du point (sortie graphique) ne soit effectué que lorsque les coordonnées du point satisfont à la condition $y \leq x^2$*

Pour en savoir plus..

L'élève doit utiliser une instruction SI dans l'algorithme qui introduit un choix sur les points effectivement tracés.

Activité de l'élève

Consignes :

Modifier l'algorithme afin que la sortie graphique (tracé du point) soit conditionnelle (la condition de tracé proposée est $y \leq x^2$).

Commentaires :

Une réflexion commune permet de trouver qu'il faut utiliser une structure SI.

- J'indique ensuite comment déplacer l'instruction de tracé du point car la méthode est un peu « archaïque » avec Algobox. Il me faut aussi préciser aux élèves que x^2 s'écrira $x*x$ avec ce logiciel.
- La sortie graphique montre tout de suite une différence nette avec ce qui précède. C'est une validation intuitive de la réussite dans la modification de l'algorithme.
- La sortie texte permet de vérifier à la calculatrice si les points calculés aux différentes étapes sont tracés ou non : dans l'exemple ci-dessous, le 99ème point est tracé mais pas le 100ème.

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

De l'algorithmique pour faire des mathématiques

Le placement de la structure SI dans l'algorithme et le déplacement de l'instruction de tracé du point donnent lieu à des allers-retours sur la structure de l'algorithme. La « validation » est ici graphique et intuitive.

++++Intentions mathématiques

Les points effectivement tracés sont situés sous ou sur la parabole d'équation $y=x^2$.

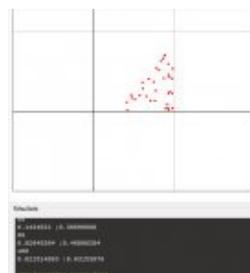
Nous retrouverons cela lors de la leçon sur la fonction carré.

++++En images

L'algorithme avec la structure SI (lignes 14 à 17) et l'instruction de tracé qui devient conditionnelle (ligne 16)

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  POUR N ALLANT_DE 1 A 100
7  DEBUT_POUR
8  x PREND_LA_VALEUR random()
9  y PREND_LA_VALEUR random()
10 AFFICHER N
11 AFFICHER x
12 AFFICHER " ;"
13 AFFICHER y
14 SI (y<=x*x) ALORS
15 DEBUT_SI
16 TRACER_POINT (x,y)
17 FIN_SI
18 FIN_POUR
19 FIN_ALGORITHME
```

Sorties texte et graphique :



La sortie graphique est nettement différente.

4ème étape : *introduire et utiliser une variable pour compter le nombre de points effectivement tracés*

Pour en savoir plus ..

L'utilisation de cette nouvelle variable nécessite plusieurs étapes : déclaration, affectation, affichage.

Activité de l'élève

Consignes :

Modifier l'algorithme afin que l'on puisse compter le nombre de points tracés.

Commentaires :

Dans un 1er premier temps on peut faire afficher le nombre de points tracés au fur et à mesure de la réalisation de la boucle POUR.

Ensuite, on peut faire en sorte que cet affichage ne soit fait qu'à la fin de l'algorithme.

Il me faut indiquer aux élèves comment incrémenter de 1 la variable comptant le nombre de points affichés car ils n'ont pas encore rencontré ce type d'affectation et ... cela ne s'invente pas !.

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

++++Intentions algorithmiques

Introduction d'une nouvelle variable (qu'il faut tout d'abord déclarer).

Modifier son affectation conditionnellement : il faut donc placer correctement cette instruction dans la structure SI.

L'instruction d'affichage du nombre de points tracés peut être placée dans la structure SI : on verra ainsi évoluer le nombre de points affichés et on pourra même vérifier la validité de l'affichage en vérifiant avec les coordonnées du point.

Ensuite cette instruction peut être déplacée après la fin de la boucle POUR afin d'alléger les sorties texte. On voit ainsi l'importance du placement de cette instruction dans le déroulement de l'algorithme.

Voici les sorties texte et graphique obtenues (le nombre de points tracé est affiché seulement à la fin).



++++Intentions mathématiques

En relevant dans un tableau le nombre de points affichés lors de 10 essais, on fait un relevé statistique qui sera exploité ensuite.

++++En images

- Déclaration de la nouvelle variable **NbPointsTraces** : ligne 5.
- Incrémentation conditionnelle de cette variable : ligne 18.
- Affichage de cette variable : ligne 19 et 20.

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPointsTraces EST_DU_TYPE NOMBRE
6  DEBUT_ALGORITHME
7  POUR N ALLANT_DE 1 A 100
8  DEBUT_POUR
9  x PREND_LA_VALEUR random()
10 y PREND_LA_VALEUR random()
11 AFFICHER N
12 AFFICHER x
13 AFFICHER " ;"
14 AFFICHER y
15 SI (y<=x*x) ALORS
16 DEBUT_SI
17 TRACER_POINT (x,y)
18 NbPointsTraces PREND_LA_VALEUR NbPointsTraces+1
17 FIN_SI
18 FIN_POUR
19 AFFICHER "Nombre de points tracés : "
20 AFFICHER NbPointsTraces
21 FIN_ALGORITHME
```

Sortie texte et graphique lorsque l'affichage du nombre de points tracé est fait uniquement en fin d'algorithme :



5ème étape : *modifier le nombre de répétitions de la boucle POUR afin de calculer les coordonnées de 250 points.*

[Pour en savoir plus ..](#)

Activité de l'élève

Consignes :

Modifier l'algorithme afin que soient calculées les coordonnées de 250 points.

Commentaires :

Le logiciel suspend l'algorithme à cause du trop grand nombre de données qu'il doit afficher (sortie texte). Il faut l'autoriser à poursuivre.

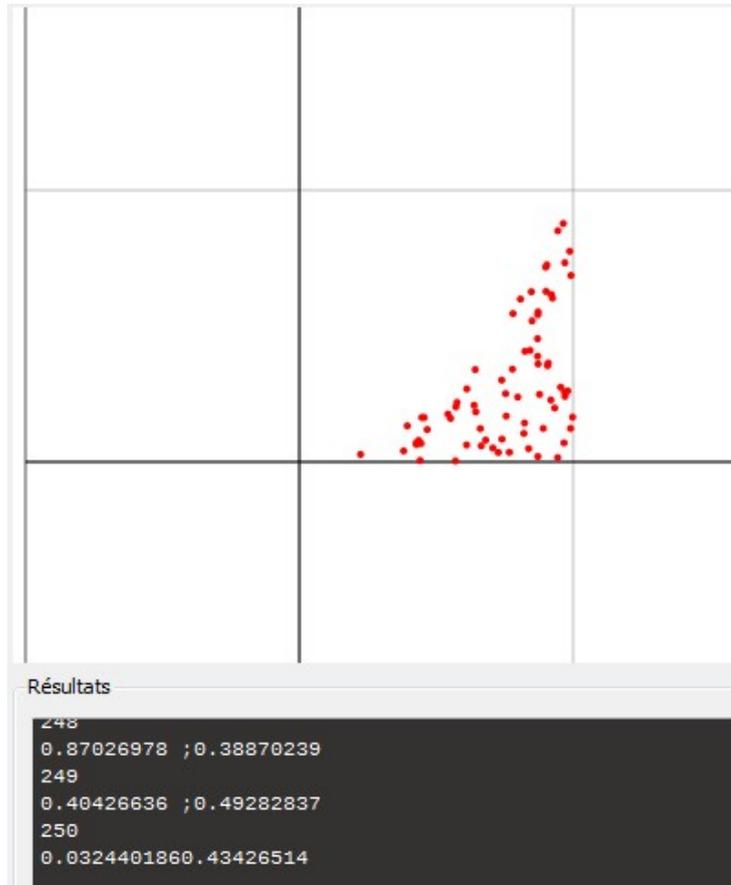
[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

Bien fixer le rôle du Pour.

Voici les sorties texte et graphique obtenues :



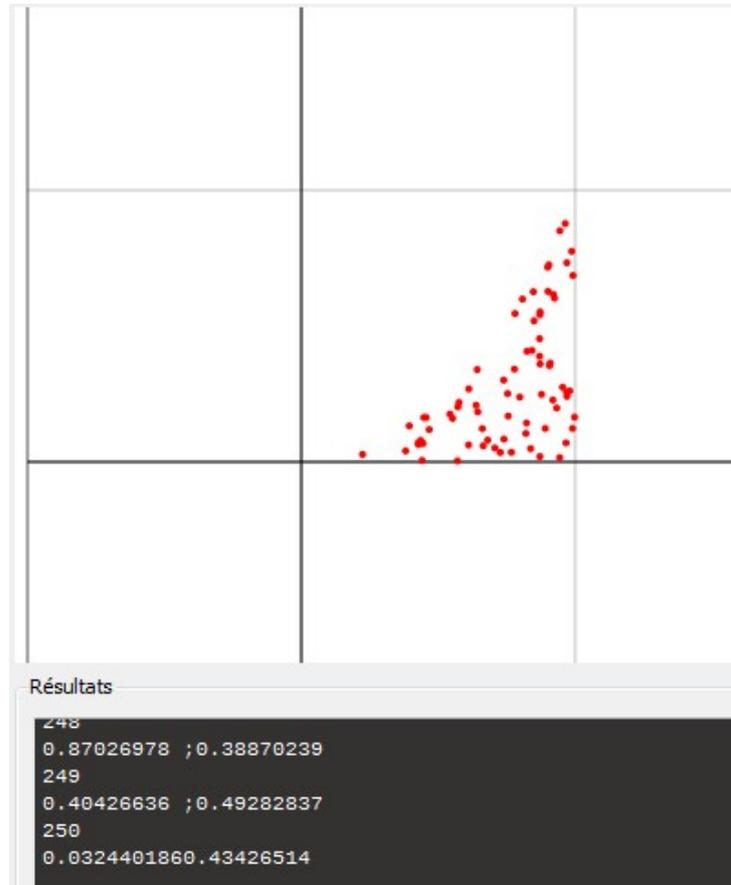
++++Intentions mathématiques

Un autre relevé statistique est fait . La comparaison avec le tableau obtenu précédemment ne pourra se faire qu'en passant aux fréquences car le nombre total de points calculés est différent.

++++En images

- la ligne 7 a été modifiée.

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPointsTraces EST_DU_TYPE NOMBRE
6  DEBUT_ALGORITHME
7  POUR N ALLANT_DE 1 A 250
8  DEBUT_POUR
9  x PREND_LA_VALEUR random()
10 y PREND_LA_VALEUR random()
11 AFFICHER N
12 AFFICHER x
13 AFFICHER " ;"
14 AFFICHER y
15 SI (y<=x*x) ALORS
16 DEBUT_SI
17 TRACER_POINT (x,y)
18 NbPointsTraces PREND_LA_VALEUR NbPointsTraces+1
17 FIN_SI
18 FIN_POUR
19 AFFICHER "Nombre de points tracés : "
20 AFFICHER NbPointsTraces
21 FIN_ALGORITHME
```



6ème étape : *calculer et afficher la proportion (fréquence) de points tracés par rapport au nombre de points calculés*

[Pour en savoir plus ..](#)

Activité de l'élève

Consignes :

Modifier l'algorithme pour que soit donné à la fin de l'algorithme la proportion (fréquence) de points tracés par rapport au nombre de points calculés.

Commentaires : Le diviseur ne peut être N (compteur de la boucle) car à la fin de la boucle il est augmenté de 1 automatiquement.

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

Voici les sorties texte et graphique obtenues :

```
24 AFFICHER N
25 AFFICHER a
26 AFFICHER " "
27 AFFICHER y
28 SI (TYPE) ALORS
29   DEBIT_RI
30   TRACER_POINT (x,y)
31   REPRESENTER MESSAGE_NOMBRE REPRESENTATION
32   FIN_SI
33 FIN_POUR
34 AFFICHER "Nombre de points tracés : "
35 AFFICHER REPRESENTATION
36 PROPORTION (NOMBRE_POINTS/NOMBRE_POINTS_MAX)
37 AFFICHER "Proportion de points tracés : "
38 AFFICHER PROPORTION
39 FIN_ALGORITHME
```



```
Execution
N: 1000000
a: 0.5
y: 0.5
Nombre de points tracés : 1000000
Proportion de points tracés : 0.5
```

++++Intentions mathématiques

Pour le cours de Mathématiques, on récupère dans un tableau des fréquences qui pourront être traitées statistiquement lors des leçons sur les probabilités

++++En images

- La variable utilisée pour le calcul de la proportion porte un nom « naturel » (ligne 6)
- Elle est affectée ligne 24 sans utiliser N (il aurait fallu diviser par N-1)
- Et affichée lignes 25 et 26

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPointsTraces EST_DU_TYPE NOMBRE
6  Proportion EST_DU_TYPE NOMBRE
7  DEBUT_ALGORITHME
8  POUR N ALLANT_DE 1 A 250
9    DEBUT_POUR
10   x PREND_LA_VALEUR random()
11   y PREND_LA_VALEUR random()
12   AFFICHER N
13   AFFICHER x
14   AFFICHER " ;"
15   AFFICHER y
16   SI (y<=x*x) ALORS
17     DEBUT_SI
18     TRACER_POINT (x,y)
19     NbPointsTraces PREND_LA_VALEUR NbPointsTraces+1
20     FIN_SI
21   FIN_POUR
22   AFFICHER "Nombre de points tracés : "
23   AFFICHER NbPointsTraces
24   Proportion PREND_LA_VALEUR NbPointsTraces/250
25   AFFICHER "Proportion de points tracés : "
26   AFFICHER Proportion
27  FIN_ALGORITHME
```

7ème étape : supprimer des instructions d'affichage qui ralentissent le fonctionnement d'Algobox.

Pour en savoir plus ..

Activité de l'élève

Consignes :

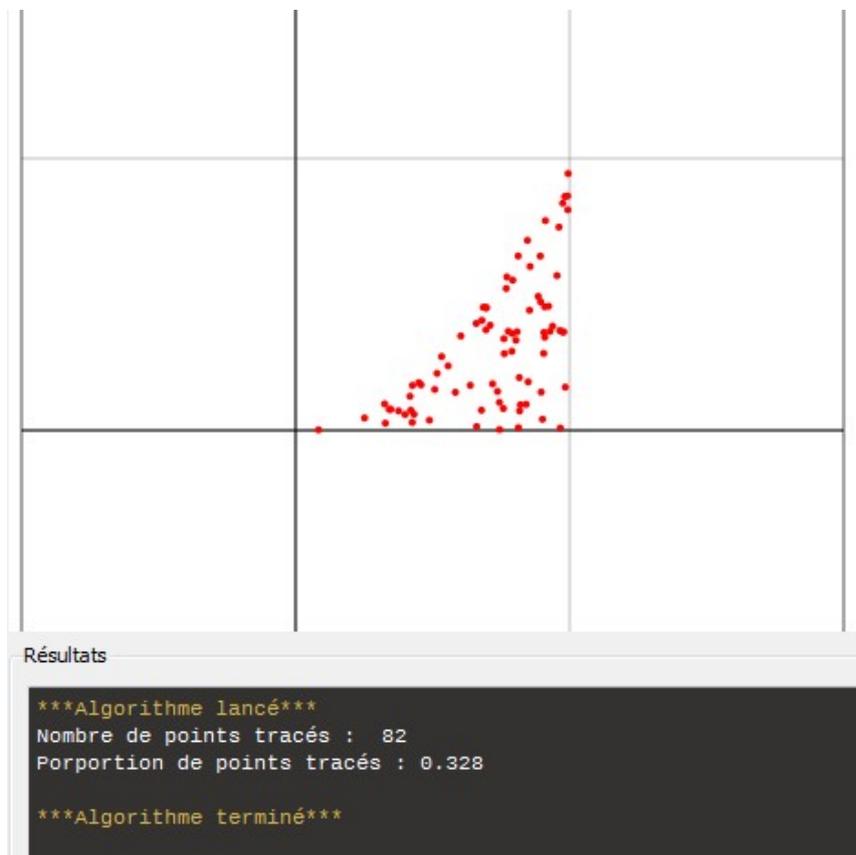
Modifier l'algorithme pour que les coordonnées des points ne soient plus affichées.

Commentaires : Le trop grand nombre d'instructions d'affichage (sortie texte) provoque l'apparition de fenêtres intempestives où Algobox demande si on veut vraiment poursuivre (au vu du grand nombre de sorties texte demandées).

++++Intentions algorithmiques

Lire l'algorithme afin de supprimer les « bonnes lignes d'instructions ». Cela nécessite de bien comprendre le rôle de chaque instruction.

Voici les sorties texte et graphique obtenues :



8ème et 9ème étapes : *introduire une instruction de lecture d'un nombre saisi par l'utilisateur puis faire un relevé statistique.*

Pour en savoir plus ..

Après avoir modifié l'algorithme afin que le nombre points à calculer soit demandé au début, les élèves l'utilisent pour remplir un tableau de relevés statistiques où le nombre points à calculer est fixé (10 essais avec 10 valeurs différentes du nombre de points à calculer).

Activité de l'élève

Consignes :

Modifier l'algorithme pour que le nombre de points à calculer soit demandé au début.

Faire le relevé statistique dans le tableau fournit.

Observer les résultats obtenus dans le tableau et faire des conjectures sur la proportion calculée.

Commentaires :

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

Introduire une instruction de lecture d'une variable qui donnera le nombre de répétitions de la boucle POUR.

Modifier l'algorithme afin d'utiliser cette variable.

++++Intentions mathématiques

Relevé statistique qui permettra de faire le lien entre fréquence observée et probabilité.

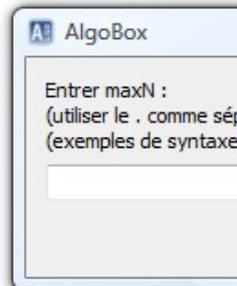
Les conjectures faites sur la proportion calculée lorsque l'on augmente le nombre de points calculés doit permettre de faire ce lien.

++++En images

- la nouvelle variable est déclarée en ligne 7 : le nom choisi est évocateur.
- il faut ensuite insérer la ligne 9 (lecture de la variable maxN).
- et modifier les lignes 10 et 22.

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPointsTraces EST_DU_TYPE NOMBRE
6  Proportion EST_DU_TYPE NOMBRE
7  maxN EST_DU_TYPE NOMBRE
8  DEBUT_ALGORITHME
9  LIRE maxN
10 POUR N ALLANT_DE 1 A maxN
11     DEBUT_POUR
12     x PREND_LA_VALEUR random()
13     y PREND_LA_VALEUR random()
14     SI (y<=x*x) ALORS
15         DEBUT_SI
16         TRACER_POINT (x,y)
17         NbPointsTraces PREND_LA_VALEUR NbPointsTraces+1
18         FIN_SI
19     FIN_POUR
20 AFFICHER "Nombre de points traces : "
21 AFFICHER NbPointsTraces
22 Proportion PREND_LA_VALEUR NbPointsTraces/maxN
23 AFFICHER "Proportion de points traces : "
24 AFFICHER Proportion
25 FIN_ALGORITHME
```

AlgoBox demande la valeur choisie pour la variable maxN :



10ème et 11ème étapes : *modifier la condition de tracé des points calculés et conjecturer la forme du « nuage » de points obtenu.*

[Pour en savoir plus ..](#)

On utilise une nouvelle condition de tracé qui doit permettre d'identifier la forme du nuage de points tracés (c'est un quart de cercle).

Activité de l'élève

Consignes :

Modifier l'algorithme pour que la condition de tracé du point soit maintenant $y \leq \sqrt{1-x^2}$

Quelle semble être la forme du nuage des points tracés ?

Faire une conjecture entre les fréquences calculées et une propriété liée à la forme du nuage des points tracés.

Commentaires :

Il faut modifier la condition de la structure SI. La fonction à utiliser pour la racine-carrée (sqrt) doit être donnée aux élèves (l'information nécessaire n'est pas indiquée clairement dans l'aide d'Algobox)

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

Ré-écriture de la condition dans la structure SI qu'il faut donc situer.

Voici les sorties texte et graphiques obtenues :

++++Intentions mathématiques

Les élèves peuvent relancer l'algorithme en choisissant d'augmenter le nombre de points calculés (200 000 au maximum).

La forme du nuage de points est facile à conjecturer (quart de disque).

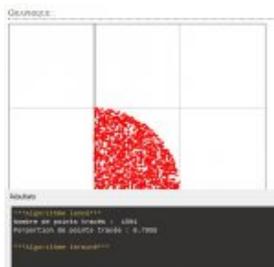
Cela pourra donner lieu à un travail mathématique ultérieur (montrer que les points ayant des coordonnées vérifiant $y \leq \sqrt{1-x^2}$ sont bien dans un quart de cercle.

++++En images

- La condition de la ligne 14 a été modifiée.

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPointsTraces EST_DU_TYPE NOMBRE
6  Proportion EST_DU_TYPE NOMBRE
7  maxN EST_DU_TYPE NOMBRE
8  DEBUT_ALGORITHME
9  LIRE maxN
10 POUR N ALLANT_DE 1 A maxN
11     DEBUT_POUR
12     x PREND_LA_VALEUR random()
13     y PREND_LA_VALEUR random()
14     SI (y<=sqrt(1-x*x)) ALORS
15         DEBUT_SI
16         TRACER_POINT (x,y)
17         NbPointsTraces PREND_LA_VALEUR NbPointsTraces+1
18         FIN_SI
19     FIN_POUR
20 AFFICHER "Nombre de points traces : "
21 AFFICHER NbPointsTraces
22 Proportion PREND_LA_VALEUR NbPointsTraces/maxN
23 AFFICHER "Proportion de points traces : "
24 AFFICHER Proportion
25 FIN_ALGORITHME
```

Voici la sortie graphique obtenue :



12ème étape : modifier la condition de tracé des points calculés : on prend une égalité au lieu d'une inégalité. Observer les sorties texte et graphique correspondantes.

Pour en savoir plus ..

La nouvelle condition de tracé doit entraîner des sorties surprenantes pour les élèves.

Activité de l'élève

Consignes :

Modifier l'algorithme pour que la condition d'affichage du point soit maintenant $y = \sqrt{1-x^2}$

Quelle semble être la forme du nuage des points tracés ?

Faire une conjecture entre les fréquences calculées et une propriété liée à la forme du nuage des points tracés.

Commentaires :

Ici une aide est nécessaire car la condition s'écrit $y == \text{sqrt}(1-x*x)$ dans Algobox.

Les élèves n'ayant pas doublé le signe = obtiennent une sortie graphique qui dans un premier temps pourrait paraître correcte (elle est proche d'une courbe) mais qui est incorrecte. Rechercher une explication à cette sortie graphique sous forme de pseudo-courbe serait intéressant mais cela m'a paru hors de propos avec mes élèves.

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)

Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

Rencontrer la nécessité d'un double signe = pour traduire une condition d'égalité. Cette convention est utilisée dans beaucoup de langages informatiques.

++++Intentions mathématiques

L'observation de la proportion de points tracés (toujours 0 !!!) ouvre des pistes vers l'évènement impossible en probabilité.

Expliquer le phénomène observé va nécessiter de bien distinguer surface et ligne.

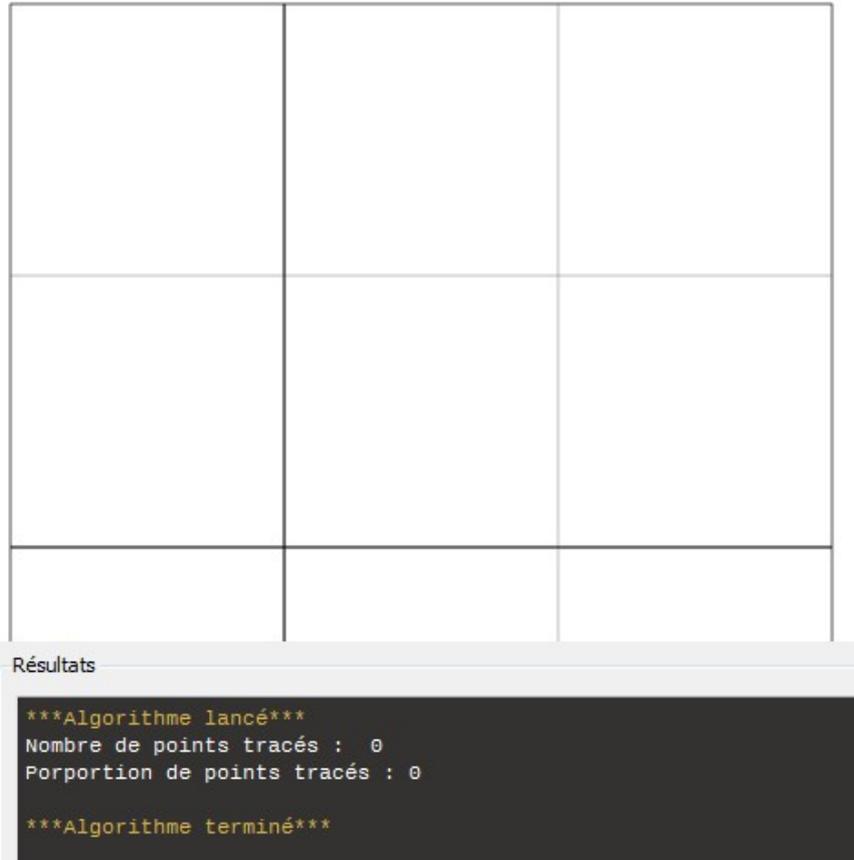
Cela va aussi nécessiter de distinguer fréquence statistique et probabilité.

++++En images

- la condition est modifiée en ligne 14.

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPointsTraces EST_DU_TYPE NOMBRE
6  Proportion EST_DU_TYPE NOMBRE
7  maxN EST_DU_TYPE NOMBRE
8  DEBUT_ALGORITHMME
9  LIRE maxN
10 POUR N ALLANT_DE 1 A maxN
11   DEBUT_POUR
12   x PREND_LA_VALEUR random()
13   y PREND_LA_VALEUR random()
14   SI (y==sqrt(1-x*x)) ALORS
15     DEBUT_SI
16     TRACER_POINT (x,y)
17     NbPointsTraces PREND_LA_VALEUR NbPointsTraces+1
18     FIN_SI
19   FIN_POUR
20 AFFICHER "Nombre de points traces : "
21 AFFICHER NbPointsTraces
22 Proportion PREND_LA_VALEUR NbPointsTraces/maxN
23 AFFICHER "Proportion de points traces : "
24 AFFICHER Proportion
25 FIN_ALGORITHMME
```

Voici des sorties bien surprenantes :



13ème, 14ème et 15ème étapes : Lire un algorithme, écrit sur papier, et décrire ce qu'il fait.
Tenter de prévoir la forme de la sortie graphique que l'on obtiendra.
Contrôler la conjecture en utilisant l'algorithme.

[Pour en savoir plus ..](#)

Il est demandé aux élèves d'anticiper les résultats obtenus lavant que l'algorithme ne soit mis en oeuvre.

Activité de l'élève

Consignes :

Que fait l'algorithme écrit sur la feuille fournie ?

Quelle est la forme du nuage de points que vous attendez ?

Charger l'algorithme TP1 qui est fourni et le faire fonctionner afin de contrôler la conjecture faite.

Commentaires :

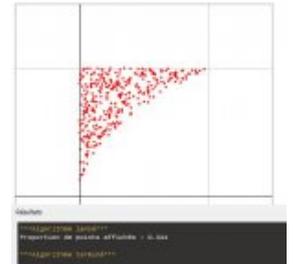
Les changements par rapport aux algorithmes précédents sont « minimales ».

[L'algorithme peut être testé en cliquant sur ce lien](#) (qui s'ouvre dans un nouvel onglet ou une nouvelle fenêtre)
Si les lettres accentuées ne sont pas bien visualisées, vous pouvez changer l'encodage des caractères (choisir Unicode UTF-8)

++++Intentions algorithmiques

Pour se rendre compte de la différence entre cet algorithme et les précédents, la lecture de la condition de tracé est indispensable ainsi que celle du nombre de répétitions de la boucle POUR.

Voici les sorties texte et graphiques obtenues :



++++Intentions mathématiques

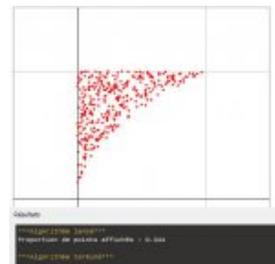
Faire un relevé statistique des fréquences de tracé des points calculés.
Les fréquences obtenues ressemblent bien sûr à celles obtenues avec la condition $y \leq x^2$.
Voici ouverte une autre piste de réflexion !

++++En images

- L'algorithme fournit sur papier aux élèves :

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPoints EST_DU_TYPE NOMBRE
6  p EST_DU_TYPE NOMBRE
7  DEBUT_ALGORITHME
8  POUR N ALLANT_DE 1 A 1000
9    DEBUT_POUR
10   x PREND_LA_VALEUR random()
11   y PREND_LA_VALEUR random()
12   SI (y>=sqrt(x)) ALORS
13     DEBUT_SI
14     TRACER_POINT (x,y)
15     NbPoints PREND_LA_VALEUR NbPoints+1
16     FIN_SI
17   FIN_POUR
18   p PREND_LA_VALEUR NbPoints/1000
19   AFFICHER "Proportion de points affichés : "
20   AFFICHER p
21 FIN_ALGORITHME
```

Les sorties texte et graphique correspondantes :



16^{ème} étape : *modifier une condition afin que la sortie graphique corresponde à un modèle. Utiliser des conditions faisant intervenir un connecteur OU puis un connecteur ET.*

Pour en savoir plus ..

Ce travail s'apparente à un « jeu de cible ». La cible est un graphique fourni. Le jeu consiste à modifier l'algorithme pour que la sortie graphique obtenue ressemble à la cible.

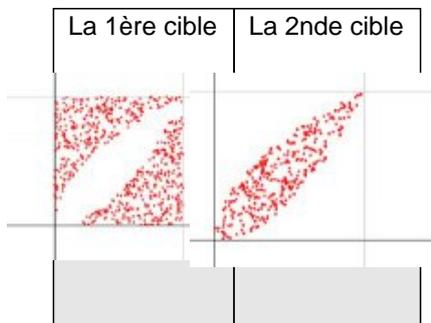
Activité de l'élève

Consignes :

*Modifier cet algorithme afin d'obtenir une sortie graphique proche de la 1^{ère} figure fournie.
recommencer afin d'obtenir une sortie graphique proche de celle de la 2nde figure*

Commentaires :

Il n'est pas facile d'imaginer l'usage de 2 conditions connectées mais la cible graphique est là pour y aider. Elle permet aussi d'aider à faire le choix entre ET et OU.



++++Intentions algorithmiques

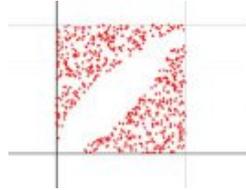
Un travail de logique sur le ET et le OU est fait à cette occasion.

++++Intentions mathématiques

On prépare le calcul de la probabilité de l'intersection de 2 évènements et de celle la réunion de ces 2 évènements.

++++En images

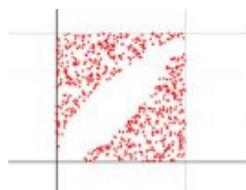
- La 1ère cible :



- L'algorithme pour la 1ère cible

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPoints EST_DU_TYPE NOMBRE
6  p EST_DU_TYPE NOMBRE
7  DEBUT_ALGORITHME
8  POUR N ALLANT_DE 1 A 1000
9  DEBUT_POUR
10 x PREND_LA_VALEUR random()
11 y PREND_LA_VALEUR random()
12 SI (y>=sqrt(x) OU y<=x*x) ALORS
13 DEBUT_SI
14 TRACER_POINT (x,y)
15 NbPoints PREND_LA_VALEUR NbPoints+1
16 FIN_SI
17 FIN_POUR
18 p PREND_LA_VALEUR NbPoints/1000
19 AFFICHER "Proportion de points affichés : "
20 AFFICHER p
21 FIN_ALGORITHME
```

- La 2nde cible :



- L'algorithme pour la 2nde cible

```
1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  y EST_DU_TYPE NOMBRE
4  N EST_DU_TYPE NOMBRE
5  NbPoints EST_DU_TYPE NOMBRE
6  p EST_DU_TYPE NOMBRE
7  DEBUT_ALGORITHME
8  POUR N ALLANT_DE 1 A 1000
9    DEBUT_POUR
10   x PREND_LA_VALEUR random()
11   y PREND_LA_VALEUR random()
12   SI (y<=sqrt(x) ET y>=x*x) ALORS
13     DEBUT_SI
14     TRACER_POINT (x,y)
15     NbPoints PREND_LA_VALEUR NbPoints+1
16     FIN_SI
17   FIN_POUR
18   p PREND_LA_VALEUR NbPoints/1000
19   AFFICHER "Proportion de points affichés : "
20   AFFICHER p
21 FIN_ALGORITHME
```

Bilans et perspectives

Analyse à posteriori de la séquence :

Du côté algorithmique et logiciel : il a fallu soutenir les élèves dans les premières questions en dirigeant certaines modifications après discussion dans la classe sur ce qu'il y avait lieu de faire. Le rôle du POUR n'est acquis que peu à peu (certains hésitent à le modifier lorsque la consigne consiste à changer le nombre de points calculés).

Les recherches sur les conditions faisant intervenir ET / OU peuvent se faire en allant « à la pêche » à la bonne réponse. La tâche qui consiste à analyser les sorties graphiques risque d'être court-circuitée.

Du côté mathématique : l'aléatoire a été facilement perçu et a permis d'expliquer les différences observées dans les calculs demandés au logiciel.

Le mot probabilité a été prononcé par quelques élèves à propos du comportement des proportions obtenues lorsque l'on augmente le nombre de points calculés mais ce n'est pas la conjecture qui a été faite le plus souvent. Les conjectures étaient plutôt du type « la proportion est entre 0,3 et 0,5 ».

Le lien entre la fréquence et l'aire du quart de disque a aussi été faite par plusieurs élèves.

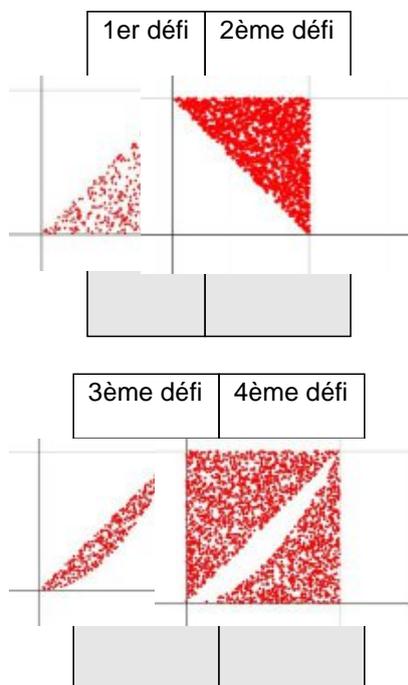
La proportion nulle et l'absence de sortie graphique lorsque la condition est une égalité a interpellé les élèves sans que nous ayons le temps d'exploiter cela : nous y reviendrons lors de la leçon sur l'évènement impossible.

Les pistes ouvertes vers les probabilités seront exploitées pour alimenter la leçon à venir.

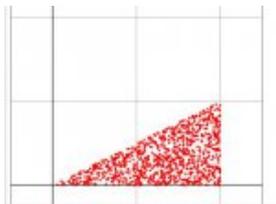
Perspectives

Pour prolonger ce travail et maintenir la réflexion jusqu'au leçons pouvant le ré-utiliser (fonction carré, probabilités) je vais organiser, en fin d'heure, pendant quelques séances, un « défi » sur le thème suivant :

- vidéo-projection d'une sortie graphique (voir quelques exemples ci-dessous)
- réflexion collective sur la condition à utiliser pour obtenir une telle sortie
- réflexion collective afin de conjecturer sur les fréquences de tracé des points qui seront calculées par l'algorithme.



D'autres défis sont encore envisageables dont celui-ci :



La proportion calculée ne pourra pas s'interpréter directement comme approximation de l'aire « sous la droite d'équation $y=0,5x$ ». Cela permettra de d'affiner la modélisation mise en place avec les exemples précédents.

Conclusion

La situation de départ s'est révélée riche de possibilités (dont certaines ne me sont apparues qu'après avoir réalisé une 1ère séance de 1h30 avec mes élèves).

Je n'ai pas terminé l'exploration des suites que je pourrai donner à ce travail.

La séquence est largement perfectible.

Par exemple, la 1ère condition pourrait être plus simple y . Celle-ci pourrait faciliter l'interprétation de la proportion calculée en terme d'aire.

D'autre part, on pourrait comparer les proportions calculées avec une condition utilisant une inégalité stricte comme y et la condition correspondante utilisant une inégalité large $y \leq x^2$. Cela pourrait être une entrée vers l'étude de la fonction carré.

D'autres pistes pourront sûrement être suivies

Pour terminer, je tiens à remercier le concepteur d'Algobox.

Je n'aurais pas imaginé une séquence aussi longue et finalement riche sans l'appui d'un logiciel fournissant aussi aisément des sorties graphiques.