

La cueillette

Année 2017 – 2018

Nathan BOUVIER, Pierrick CHARLES-TENEGAL, Robin COLLET, Bastien DE JESUS, Clovis DEGLIN, Florent DESTRUN, Amory GAILLARD, Noam LEMAITRE, Maxime MURRAY, Chloé MUYLLE, Martin PINET, Camille POUESSEL, Tom SERRA, Mathys STEBIG, élèves de classe de cinquième, quatrième et troisième.

Encadrés par Martine GIRAUD, Hélène LECOUTRE, Bernard STEELANDT.

Établissements : Collège Jacques Monod de Pérenchies et collège Théodore Monod de Lesquin

Chercheuse : Emeline SCHMISSER, UFR de Mathématiques, Université des Sciences et Techniques de Lille.

1. Présentation du sujet

Le verger est un jeu collaboratif pour enfants à partir de 2 ans.

Joueurs :

De 1 à 4 joueurs.

Le but du jeu :

Récupérer les fruits avant l'arrivée du corbeau.

Pièces du jeu :

1 corbeau, 1 dé, 4 arbres, le panier qui sert à ranger les fruits que l'on a récoltés.

Une des faces du dé permet au joueur de choisir un des fruits restants sur les arbres.

Règle du jeu :

Un joueur lance le dé.

Si le dé tombe sur une des 4 faces « couleur », le joueur prend le fruit correspondant à la couleur.

Si le dé tombe sur la face « panier », il prend le fruit de son choix.

Si le dé tombe sur la face « corbeau », le corbeau avance de 1 (sur un trajet de 5).

Les problèmes posés sont :

- Quel est la probabilité de gagner et en combien de coups ?

- Et que se passe t-il avec un jeu plus complexe (avec plus de fruits par exemple) ?



Ci-dessus : Le jeu « Le verger »

2. Annonce des conjectures et résultats obtenus

Nous avons trouvé 3 tactiques, dépendantes de la face « panier ».

Chacune de ces tactiques aboutit à des résultats légèrement différents.

3. Le détail de nos recherches

I. Différentes tactiques

Une des faces du dé permet au joueur de choisir un des fruits restant sur les arbres.

Il y a 3 tactiques, dépendantes de cette face :

- la manière enfantine qui consiste à prendre le fruit de sa couleur préférée.
- la manière logique.
- la manière anti-collaborative. (1)

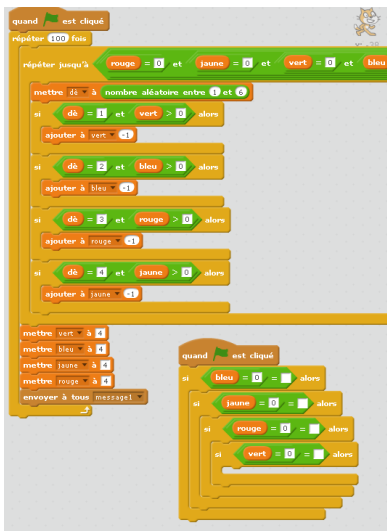
II. Études différentes sur plusieurs plates-formes informatiques (scratch, python et tableur)

Les supports utilisés pour notre recherche sont : SCRATCH, TABLEUR et PYTHON.

a. Partie Scratch :

Il nous a paru évident dès le départ d'essayer de programmer le jeu à l'aide du logiciel Scratch.

1. Nous avons commencé par programmer les faces du dé permettant d'enlever un fruit à la face correspondante.



2. Nous avons ensuite programmé la 5^e face de notre dé qui permet de choisir un fruit parmi tous. Cette face est déterminante pour influencer le pourcentage de chance de gagner ou de perdre.

3. Pour terminer, nous avons programmé le déplacement du corbeau. Le jet de dé envoie un message au corbeau qui fonctionne parallèlement. Cela envoie un message au corbeau qui dépend du jet de dé. (2)

Ci-contre : Une partie du programme sous Scratch

b. Partie Tableur

Nous avons aussi essayé de développer un tableur pour simuler les parties. Voici le tableur que nous avons fabriqué.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	signification	numéro de partie	même partie ?	numéro de lancer	lancer de dé	lancer de dé pour le panier	lancer de dé sans le panier	arbre rouge	arbre jaune	arbre bleu	arbre vert	corbeau	max fruits	Gagné?	Perdu ?	autre Partie ?	Nombre de gagné	Nombre de perdu	coups par partie	coup max par partie	coup moyen pour gagner un		
1																							
2	1=rouge	1		0	5			4	4	4	4	0	4				0	0	0			20,3118	
3	2=jaune	1		1	4	1	1	3	4	4	4	0	4				0	0	1				
4	3=bleu	1		2	1	2	2	3	3	4	4	0	4				0	0	2				
5	4=vert	1		3	5	3	2	3	2	4	4	0	4				0	0	3				
6	5=panier	1		4	6	3	2	3	1	4	4	0	4				0	0	4				
7	6=corbeau	1		5	5	3	1	2	1	4	4	0	4				0	0	5				
8		1		6	6	3	6	2	1	4	4	1	4				0	0	6				
9	total	1		7	5	3	3	2	1	3	4	1	4				0	0	7				
10	183	1		8	6	4	4	2	1	3	3	1	3				0	0	8				
11	gagnées	1		9	2	3	2	2	0	3	3	1	3				0	0	9				
12	112	1		10	1	3	3	2	0	2	3	1	3				0	0	10				
13	61 %	1		11	6	4	3	2	0	1	3	1	3				0	0	11				
14	Perdus	1		12	5	4	4	2	0	1	2	1	2				0	0	12				
15	71	1		13	3	1	3	2	0	0	2	1	2				0	0	13				
16	39 %	1		14	6	1	6	2	0	0	2	2	2				0	0	14				
17		1		15	5	1	1	1	0	0	2	2	2				0	0	15				
18		1		16	6	4	6	1	0	0	2	3	2				0	0	16				
19		1		17	6	4	4	1	0	0	1	3	1				0	0	17				
20		1		18	3	1	1	0	0	0	1	3	1				0	0	18				
21		1		19	6	4	2	0	0	0	1	3	1				0	0	19				
22		1		20	5	4	3	0	0	0	1	3	1				0	0	20				
23		1		21	3	4	6	0	0	0	1	4	1				0	0	21			21	
24		1		22	2	4	4	0	0	0	0	4	0			OUI			22				
25		2	OUI	0	5	1	1	4	4	4	4	0	4				1	0	1				
26		2		1	4	1	2	4	3	4	4	0	4				1	0	2				
27		2		2	3	1	3	4	3	3	4	0	4				1	0	3				
28		2		3	1	1	1	3	3	3	4	0	4				1	0	4				
29		2		4	1	4	1	2	3	3	4	0	4				1	0	5				
30		2		5	4	4	6	2	3	3	4	1	4				1	0	6				
31		2		6	3	4	6	2	3	3	4	2	4				1	0	7				

Le tableur qui simule les coups

Les résultats se trouvent :

- Dans la cellule U2 pour le nombre de coups moyens par partie.
- Dans les cellules A12 et A15 pour le nombre de parties gagnées ou perdues.
- Dans les cellules A13 et A16 pour le pourcentage de parties gagnées ou perdues.

Voici les formules qui se trouvent dans les cellules (que nous avons copiées verticalement). Ce sont les formules de la ligne 3.

A la ligne 1, nous avons une ligne initiale où on met les nombres de départ, sauf :

- en E1 où il y a : « =ALEA.ENTRE.BORNES(1;6) »
- en U1 où il y a : « =MOYENNE(T\$1:T\$1048576) »

Le problème avec cette méthode, c'est que nous n'arrivons pas à simuler un très grand nombre de parties de suite et que le résultat est d'autant plus imprécis. En effet, une partie c'est environ 20 coups (donc 20 lignes) et les 12 591 lignes ne permettent que de faire 585 parties. Il faudrait répéter sur bien plus de parties et ce serait alors très long (en temps de copie et de calculs).

A	Signification	
B	Numéro partie	=SI(P3="OUI";B3+1;B3)
C	Même partie ?	=P3
D	Numéro du lancer	=SI(C4="OUI";0;D3+1)
E	Numéro de lancer	=ALEA.ENTRE.BORNES(1;6)
F	Lancer de dé pour panier	=SI(H3=M3;1;SI(I3=M3;2;SI(J3=M3;3;4)))
G	Lancer de dé sans le panier	=SI(E4=5;F4;E4)
H	Arbre rouge	=SI(C4="OUI";4;SI(ET(\$G4=1;H3<>0);H3-1;H3))
I	Arbre jaune	=SI(C4="OUI";4;SI(ET(\$G4=2;I3<>0);I3-1;I3))
J	Arbre vert	=SI(C4="OUI";4;SI(ET(\$G4=3;J3<>0);J3-1;J3))
K	Arbre bleu	=SI(C4="OUI";4;SI(ET(\$G4=4;K3<>0);K3-1;K3))
L	Corbeau	=SI(C4="OUI";0;SI(ET(\$G4=6;L3<>5);L3+1;L3))
M	Max fruits	=MAX(H4:K4)
N	Gagné ?	=SI(ET(M4=0;L4<5);"OUI";"")
O	Perdu ?	=SI(L4=5;"OUI";"")
P	Autre partie ?	=SI(OU(N4="OUI";O4="OUI");"OUI";"")
Q	Nombre de gagnées	=SI(N4="OUI";Q3+1;Q3)
R	Nombre de perdues	=SI(O4="OUI";R3+1;R3)
S	Coups par partie	=SI(P4="OUI";0;S3+1)
T	Coups max par partie	=SI(S5=0;S4;"")
U	Coups moyens pour gagner une partie	

c. Partie Python

Après avoir obtenu des résultats avec scratch, nous nous sommes aperçus de l'imprécision de ce dernier pour son manque d'efficacité. C'est pour cela que notre dernier essai utilise Python, un langage plus « sobre » et

efficace qui nous permet de changer rapidement la « tactique » de jeu et le nombre de fruits. On arrive à une rapidité de plus d'un million d'essais en 5 minutes. Voici notre programme :

```
import random

un = 4;
deux = 4;
trois = 4;
quatre = 4;
serpent = 0;
win = 0;
coup = 0;

for loop in range (100000):

    while (un > 0 or deux > 0 or trois > 0 or quatre > 0 )and serpent < 5:
        jes_de_des = random.randint(1,6);
        coup = coup + 1;
        if(jes_de_des == 1 and un>0):
            un = un - 1;
        if (jes_de_des == 4 and quatre>0):
            quatre = quatre - 1;
        if(jes_de_des == 3 and trois>0):
            trois = trois - 1;
        if(jes_de_des == 2 and deux>0):
            deux = deux - 1;
        if(jes_de_des == 5):
            if(un > 0):
                un = un-1;
            else:
                if(deux > 0):
                    deux = deux-1;
                else:
                    if(trois> 0):
                        trois = trois-1;
                    else:
                        if(quatre > 0):
                            quatre = quatre-1;
        if(jes_de_des == 6):
            serpent = serpent + 1
    if (serpent < 5):
        win = win+1
    un = 4;
    deux = 4;
    trois = 4;
    quatre = 4;
    serpent = 0;
print("nbr de coup", coup/100000)

print(win)
```

4. Conclusion

Les pourcentages obtenus sur python :

58,41% quand on choisit sa couleur préférée pour 21,848 coups en moyenne.

63,02% quand on est logique pour 20,866 coups en moyenne.

34,6% quand on choisit l'arbre avec le moins de fruits pour 25,663 coups en moyenne.

68,444% quand on est logique sur le jeu en version complexe pour 40,146 coups en moyenne.

15,920% quand on choisit l'arbre avec le moins de fruit sur le jeu en version complexe pour 51.897 coups en moyenne.

56,709% quand on choisit sa couleur préférée sur le jeu en version complexe pour 43,4757 coups en moyenne.

Les pourcentages obtenus avec scratch :

63,07% sur scratch lorsqu'on choisit la stratégie logique.

Les pourcentages obtenus avec le tableur :

61% avec une moyenne de 20,3 coups lorsqu'on choisit la stratégie « logique ».

Nous estimons donc que les résultats obtenus sont cohérents et donc que notre recherche a été effectuée correctement. (3)

Notes d'édition :

(1) On regrette que les 3 tactiques ne soient pas détaillées.

(2) On regrette que le fonctionnement « en parallèle » du jet de dé qui envoie un message au corbeau ne soit pas plus détaillé.

(3) On aurait aimé avoir plus de détails sur le fonctionnement de la version complexe du jeu et en particulier du nombre de fruits. On peut également se poser la question de l'impact du nombre de fruits sur la probabilité de succès.