

Collectionneur d'œufs en chocolat

Présentation lors de la semaine des maths aux élèves de 2ndB



Problème

Chaque jour, on achète un œuf en chocolat qui contient un jouet à l'intérieur. Celui-ci est donc inconnu avant la dégustation de ce chocolat. Ce jouet appartient à une collection d'un nombre fixé de jouets. Après combien d'achats peut-on espérer avoir fini la collection ?

Par **DEKEYSER Alexis** & **JARNE-JOURDAN Vasco** de 2ndB du Lycée Val de Durance de Pertuis

Présentations lors du forum des maths à l'Université St Charles de Marseille

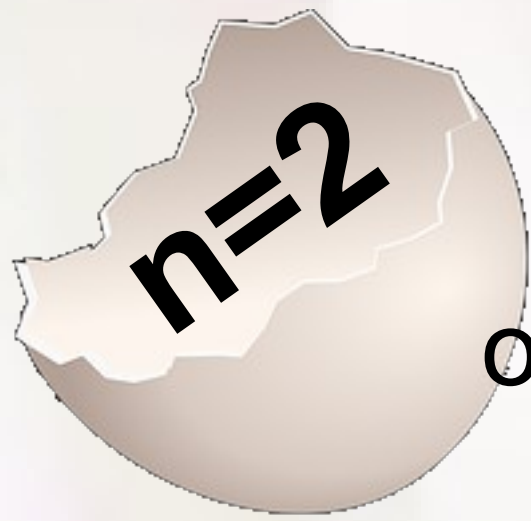


Cas simples

On note n le nombre de jouets de la collection.

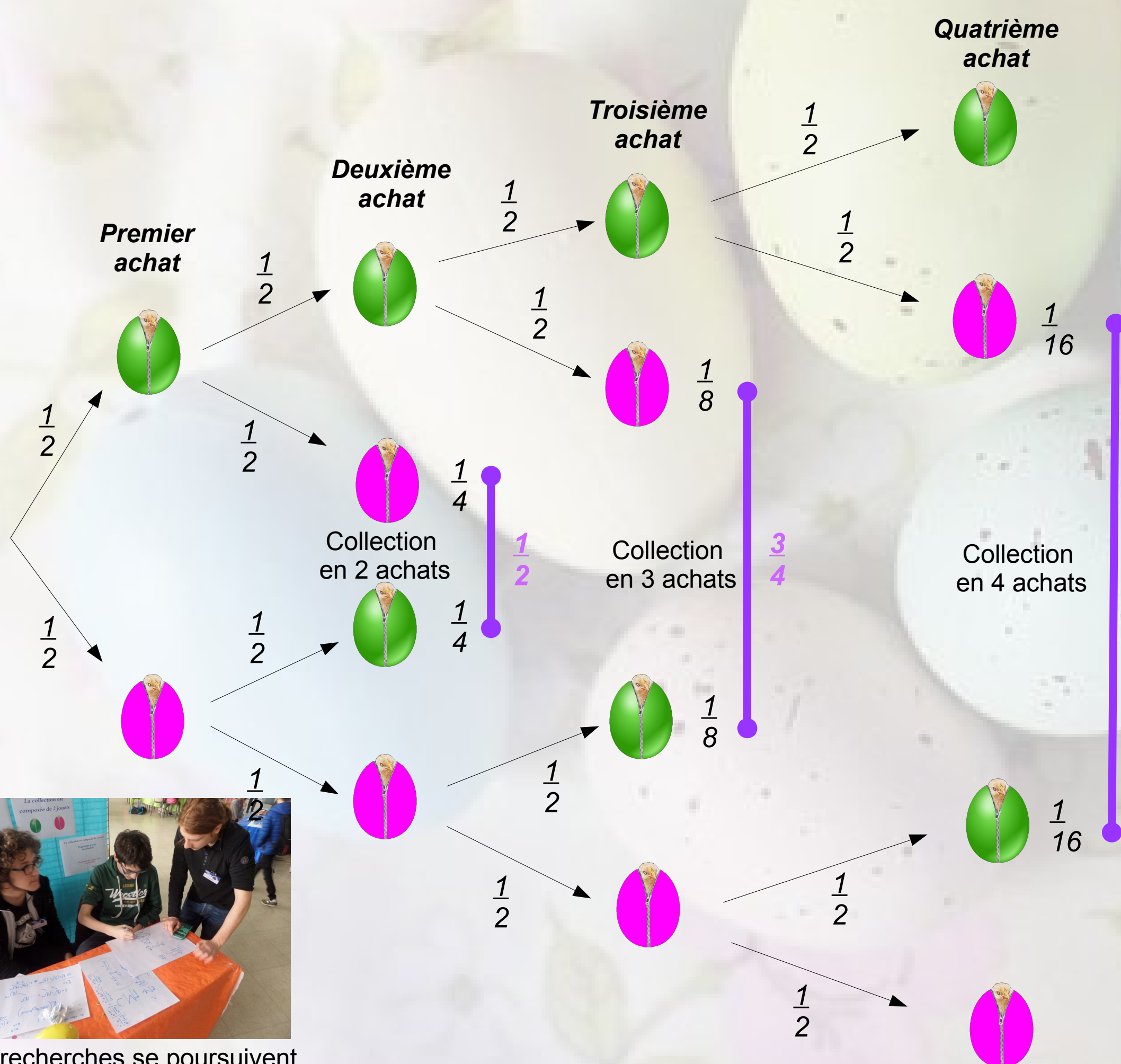


Pour $n=1$, il suffit d'un achat pour avoir la collection



Pour $n=2$,

on peut faire un arbre de probabilités avec l'évolution pour chaque achat



Les recherches se poursuivent durant le forum des maths

Propriété

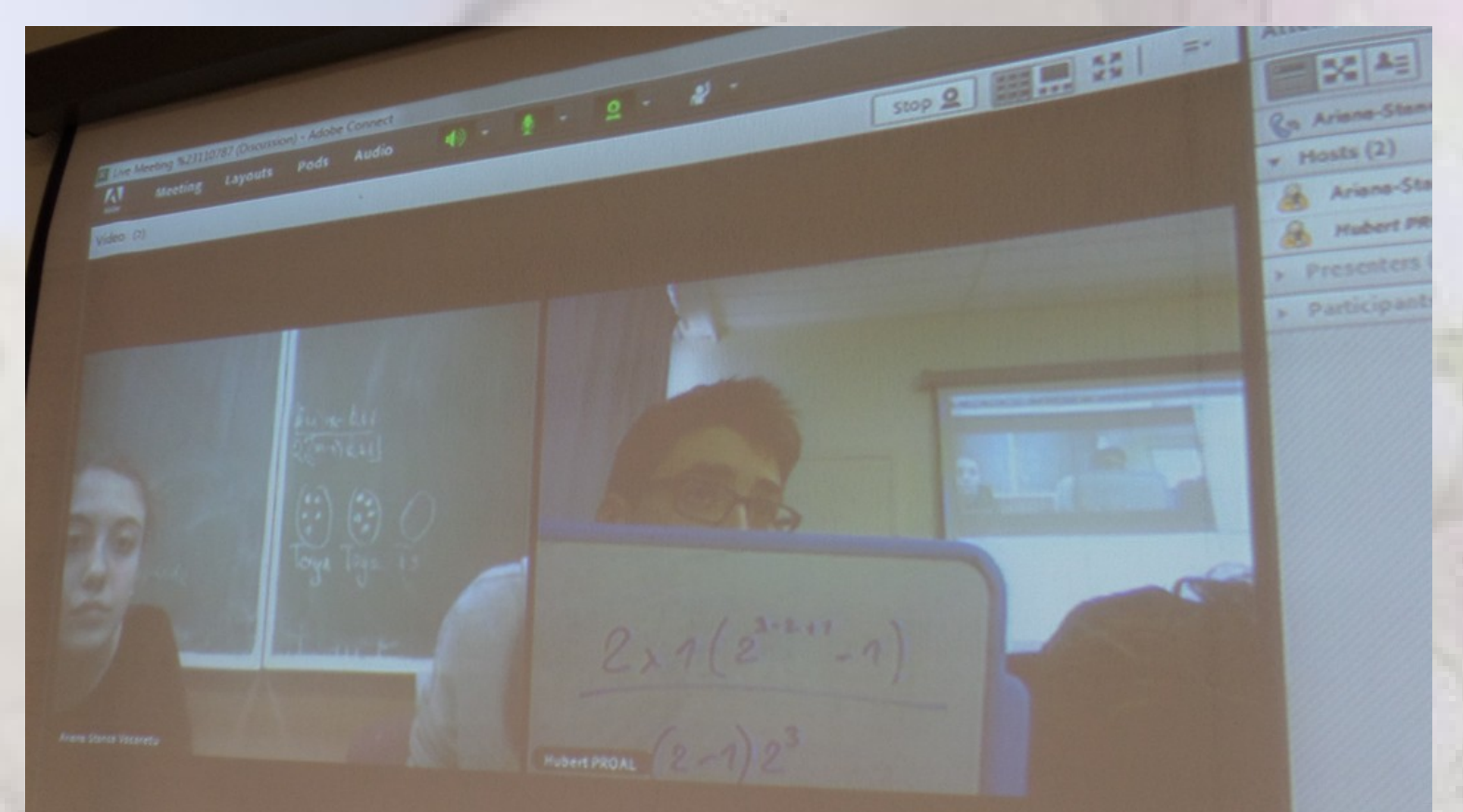
Pour une collection de $n=2$ jouets, la probabilité de ne pas avoir la collection au bout de k achats est de

$$2 \times \left| \frac{1}{2} \right|^k = \frac{1}{2^{k-1}}$$



Pour $n=3$,

les calculs théoriques ou la réalisation de l'arbre pour avoir une formule générale n'ont pas abouti



Séminaire en vidéo-conférence avec Cluj sur le problème $n=3$


```

nb_achat=[0 for i in range(100)]
nb_jouet=int(input("Nombre de jouets dans la collection"))

def produit(liste):
    taille=len(liste)
    produit=1
    for i in range(taille):
        produit=produit*liste[i]
    return produit

def achat(nombre):
    compteur=0
    collection=[0 for i in range(nombre)]
    while produit(collection)==0:
        compteur=compteur+1
        A=randint(1,nombre)-1
        collection[A]=collection[A]+1
    return compteur

for i in range(10000):
    nb_jour=achat(nb_jouet)
    nb_achat[nb_jour]=nb_achat[nb_jour]+1

def somme(liste,n):
    S=0
    for i in range(1,n+1):
        S=S+liste[i]
    return S

def seuil(liste):
    s=1
    while somme(liste,s)/10000<=0.95:
        s=s+1
    return s

def moyenne(liste):
    m=0
    for i in range(100):
        m=m+liste[i]
    return m/10000

print("La moyenne de cette simulation est ",moyenne(nb_achat))
print("La barre des 95% est franchie au bout de ",seuil(nb_achat)," achats")

```

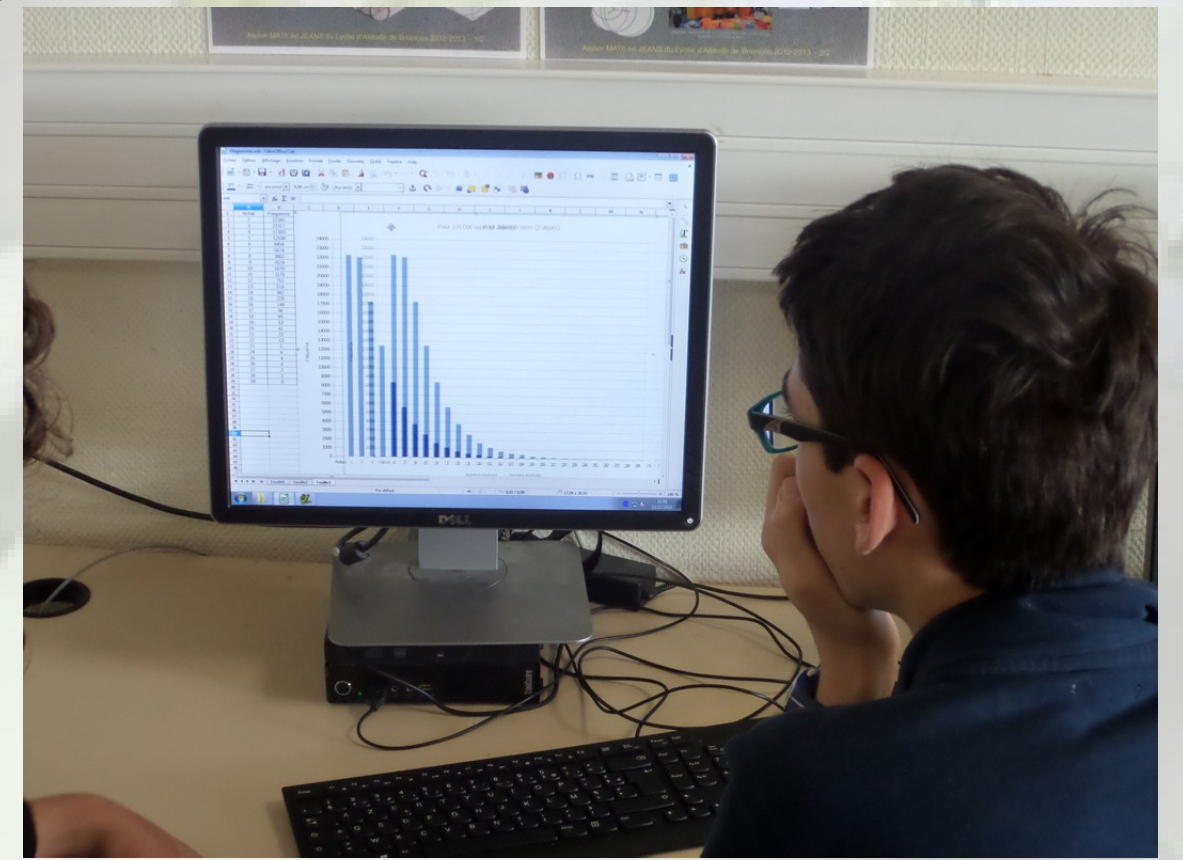


3 Simulations

Nous avons fait, en Python, plusieurs simulations, selon les valeurs de n



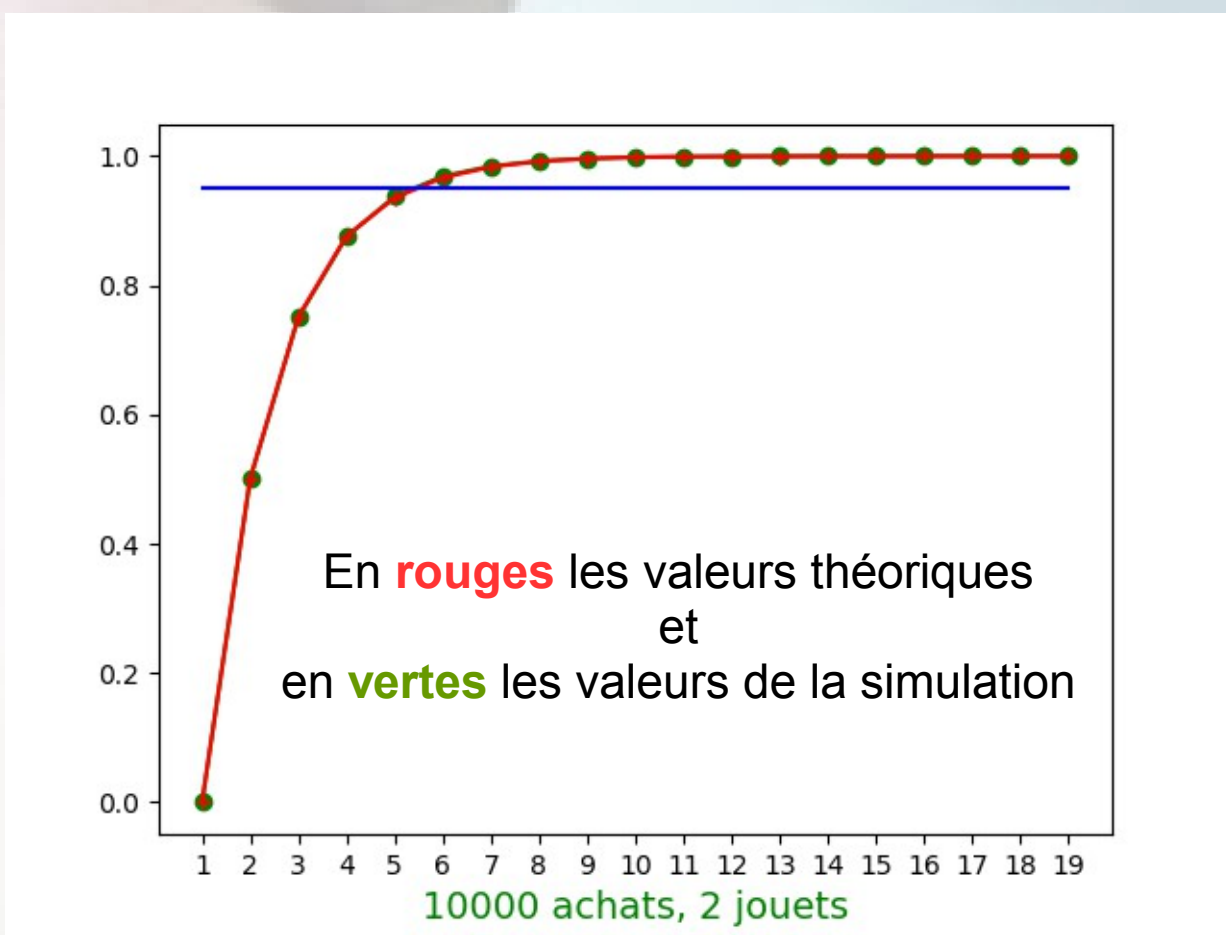
Matériel d'expérimentations et de présentations du sujet



Analyse des résultats lors des séances de recherche



4 Confrontation Simulation-théorie dans le cas n=2

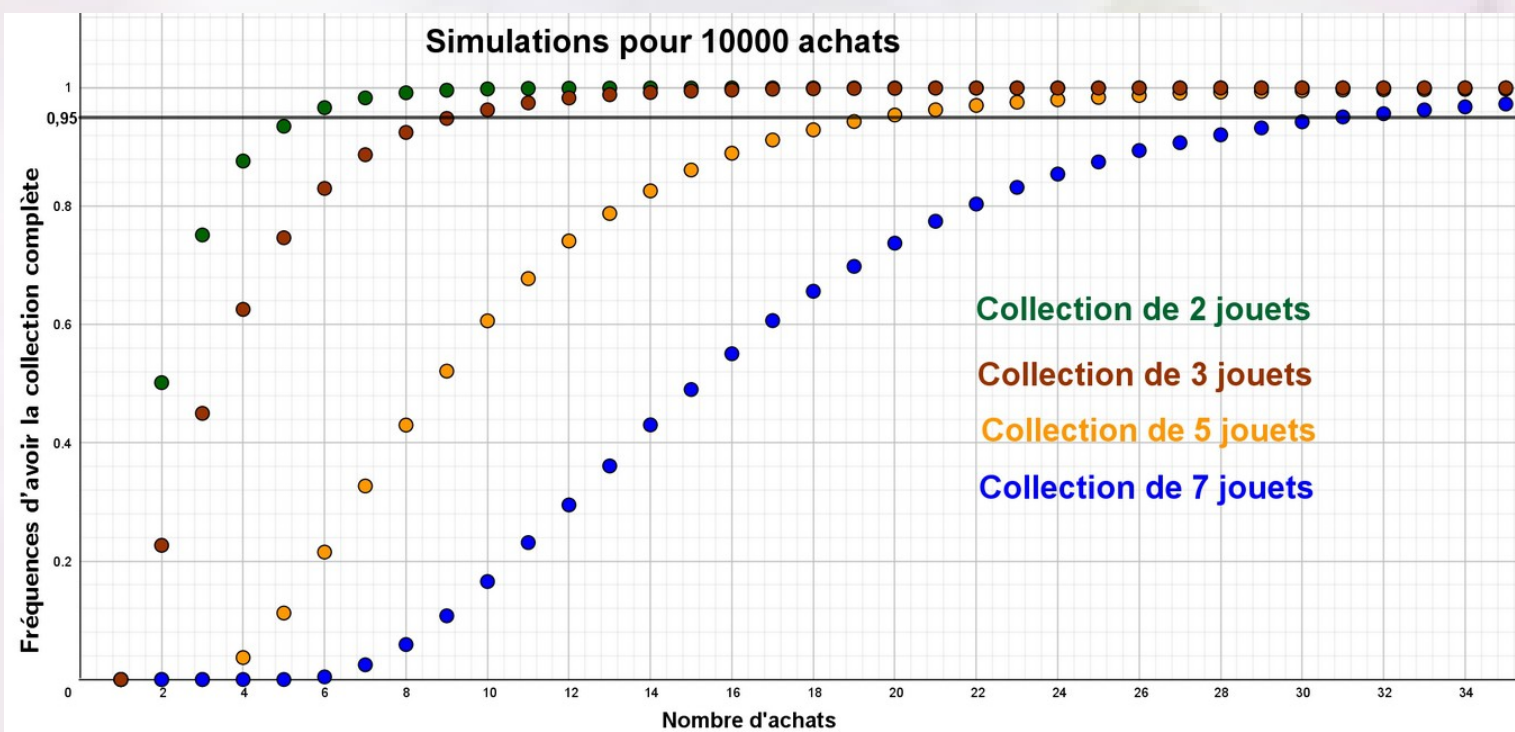
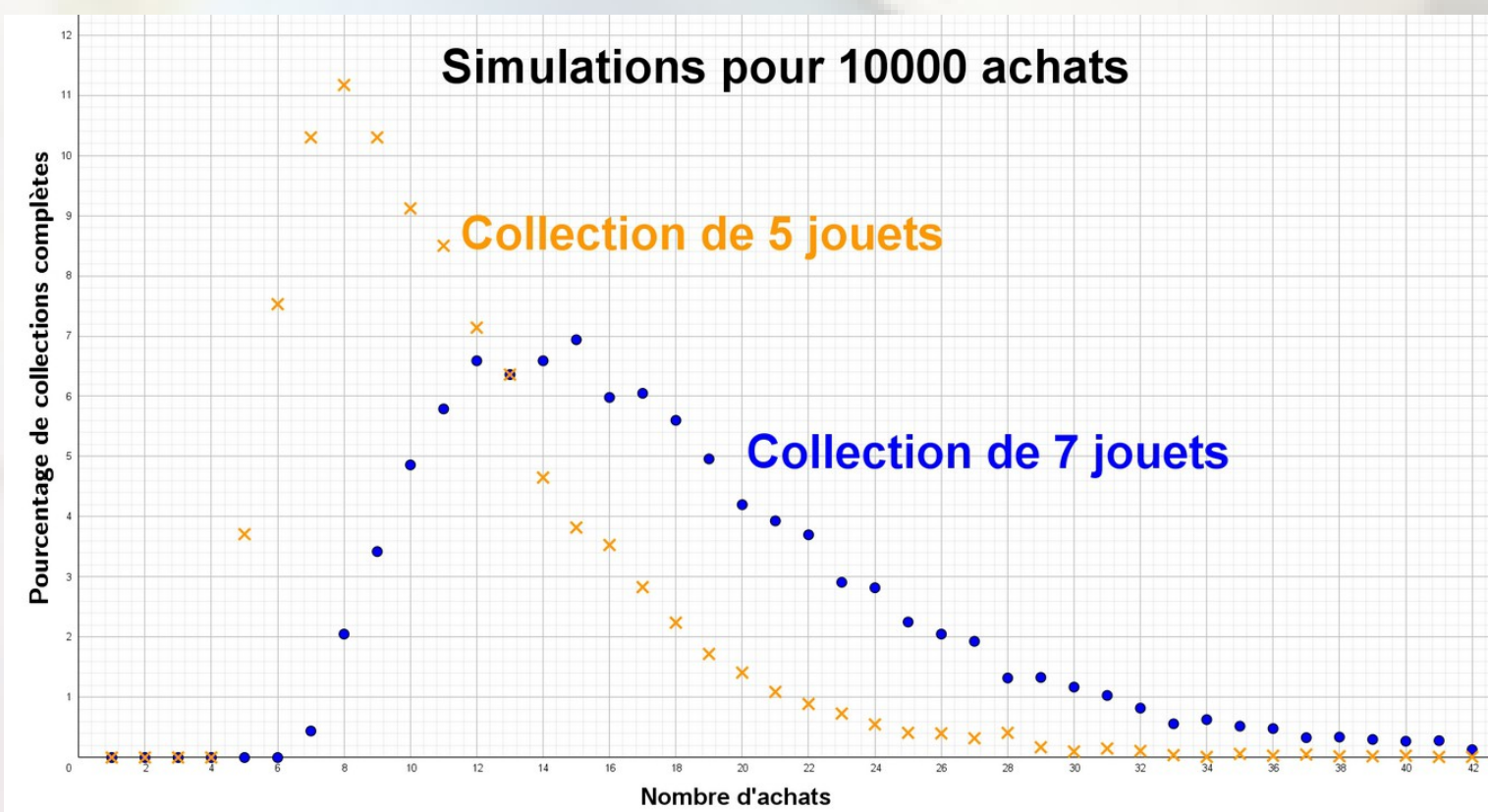
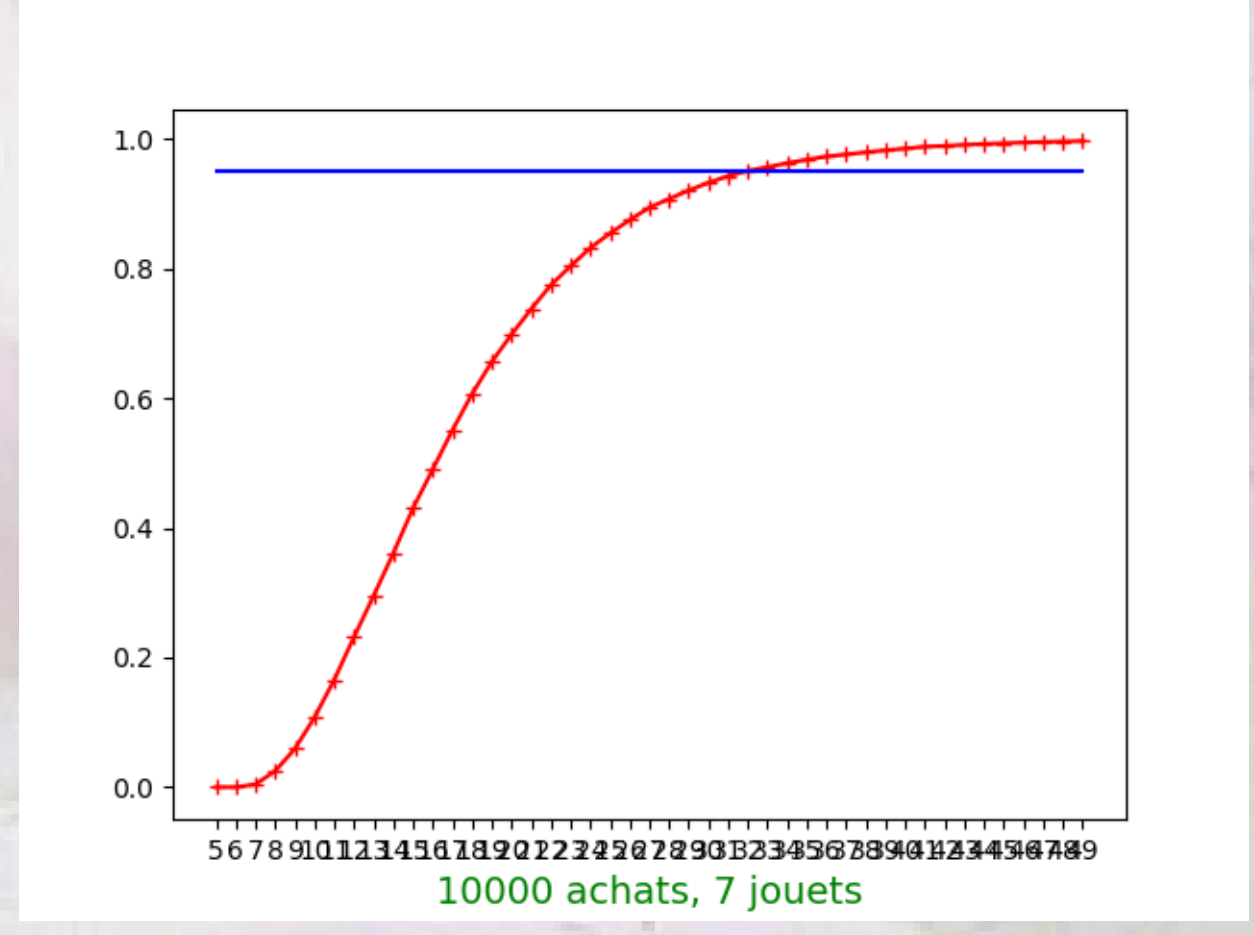
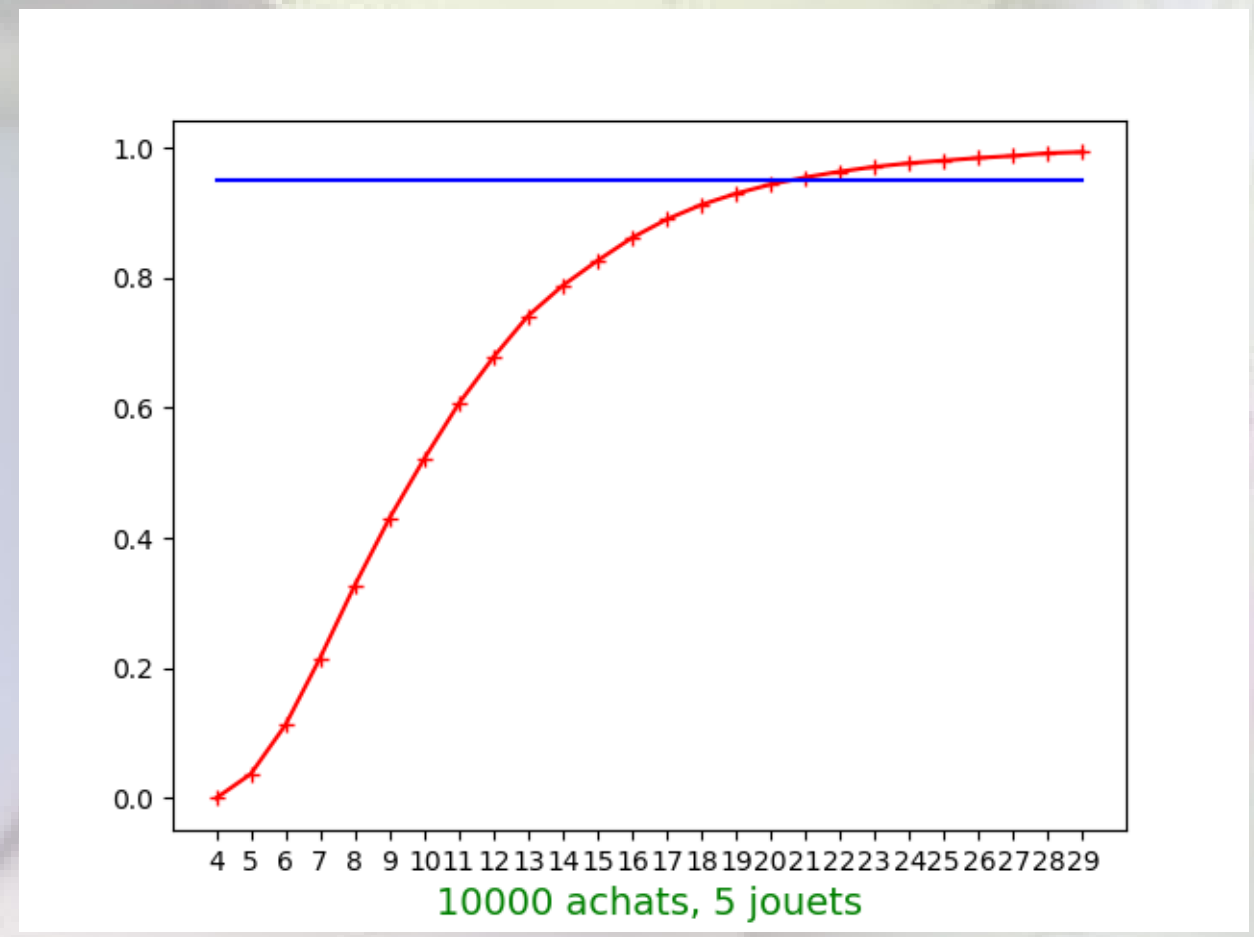
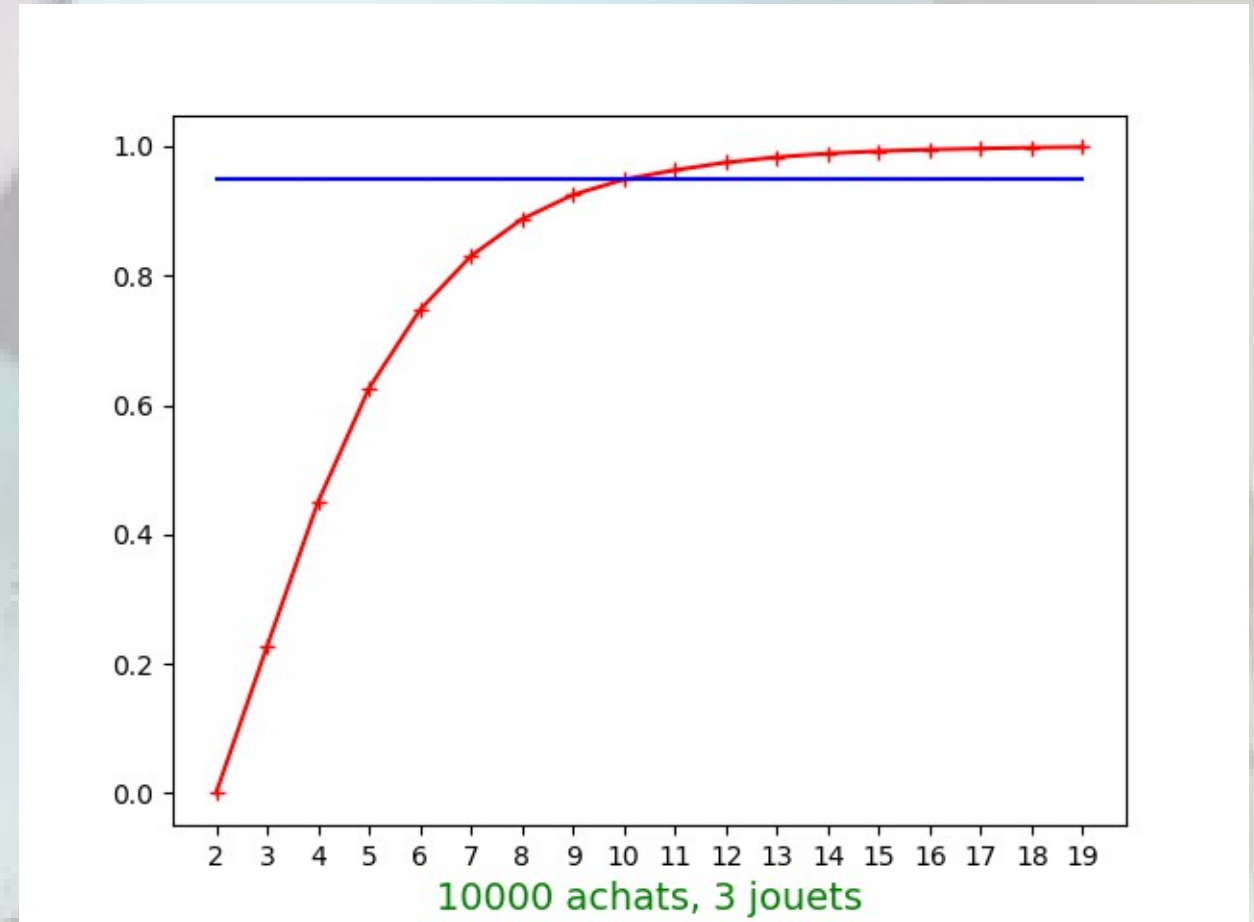


Présentations en collège lors de la semaine des maths



5 Résultats des simulations

Nos simulations sur un échantillon de 10000 achats nous conduisent aux résultats suivants



Nombre de jouets	Nombre d'achats moyen pour avoir la collection	Nombre d'achats où la fréquence d'avoir la collection dépasse 95%
2 jouets	3	6
3 jouets	5,48	10
5 jouets	11,41	20
7 jouets	18,19	32

Nos recherches ne nous ont pas permis de trouver une formule dans le cas générale