Qui est-ce ?

Une introduction aux codes correcteurs d'erreurs

Sur une feuille sont dessinés des personnages, qui se distinguent selon 4 éléments, qu'ils possèdent ou non : des lunettes, une moustache, un chapeau, des cheveux. Tous les personnages sont différents (par exemple, il ne peut pas y en avoir deux avec des lunettes, pas de moustache ni de chapeau et des cheveux). Un joueur choisit secrètement un des personnages. L'autre joueur doit deviner duquel il s'agit en posant des questions, auxquelles l'autre ne peut répondre que par « oui » ou « non », du type « a-t-il des cheveux ? ». Tous les personnages possibles sont dessinés dans le jeu.

1. **Combien de personnages y a-t-il ?**

   $2^4 = 16$. *Les puissances ne sont vues qu'en quatrième, au niveau jaune on dira* $2 \times 2 \times 2 \times 2$.

2. **Combien faut-il poser de questions pour être sûr de pouvoir trouver ?**

   On pose autant de questions qu'il y a de critères, soit 4 questions.

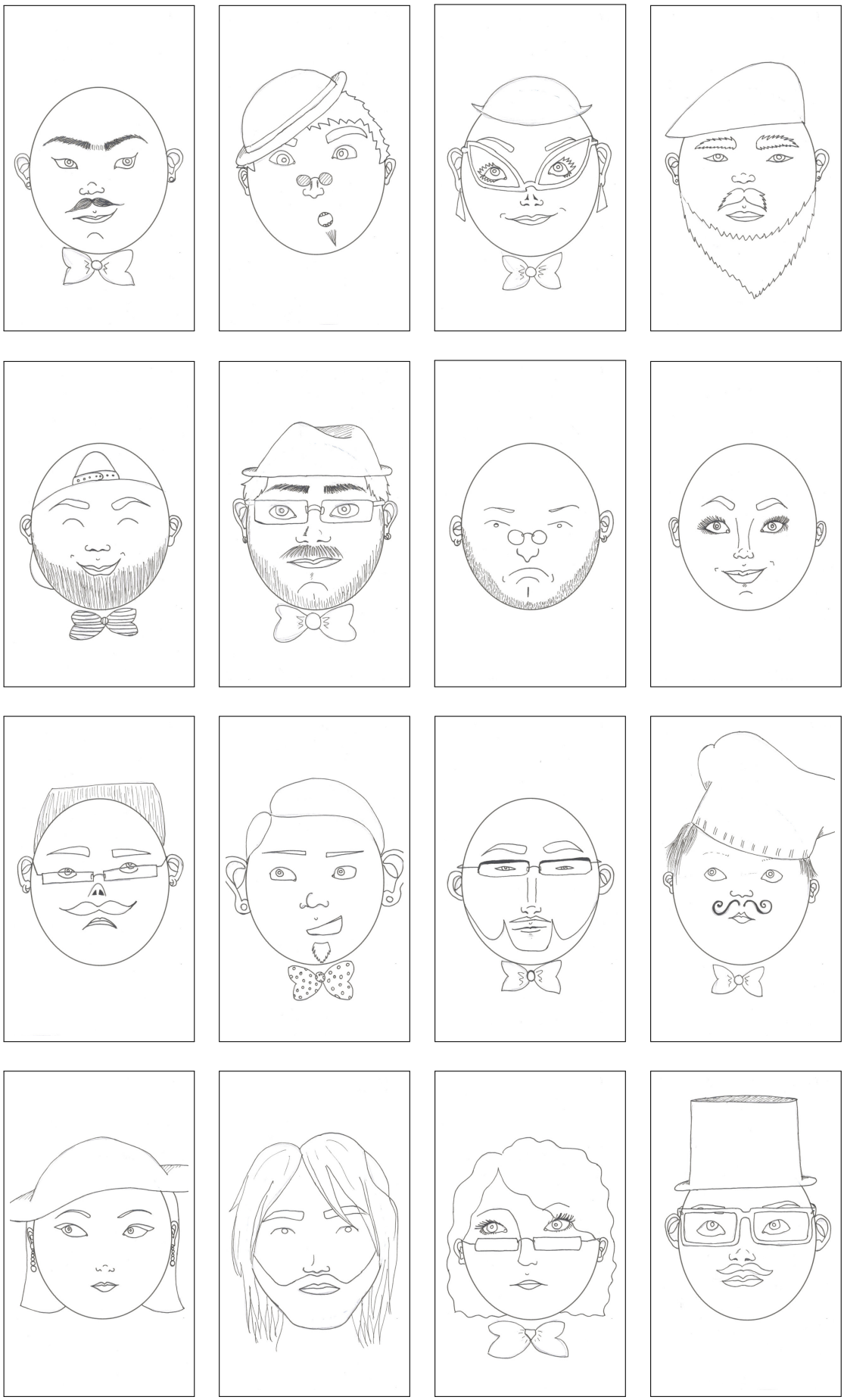
3. **On rajoute maintenant des éléments sur certains de ces personnages : boucle d'oreille, barbe, nœud papillon. Combien de questions faut-il poser pour être sûr de pouvoir trouver ?**

   Cela ne change rien... toujours les 4 mêmes questions !

On joue à ce jeu avec les éléments supplémentaires et en rajoutant la règle suivante : le joueur qui a choisi le personnage secret a le droit de mentir au plus une fois en répondant aux questions. Ces questions sont, dans l'ordre :

(1) A-t-il des lunettes ? **(2) A-t-il une moustache ?** **(3) A-t-il un chapeau ?**
(4) A-t-il des cheveux ? **(5) A-t-il des boucles d'oreille ?** **(6) A-t-il une barbe ?**
(7) A-t-il un nœud papillon ?

On peut lancer une partie sur l'ordinateur (jeu.html, à ouvrir de préférence avec Firefox, éventuellement Chrome, éviter Internet explorer et Safari). Il y a aussi un jeu de cartes (plus grandes) qu'on peut utiliser si les dessins sur l'écran sont trop petits ou pour jouer sans ordinateur.



On représente un personnage par un élément de $\mathbf{F}_2{}^7$ *:* $(x_1, \ldots, x_7)$ *avec* $x_i = 1$ *si la réponse à la*

question i est « oui », 0 sinon. Les personnages représentés dans le jeu sont les 16 éléments du $\mathbf{F}_2$-espace vectoriel engendré par $(1,0,0,0,1,1,0)$, $(0,1,0,0,1,0,1)$, $(0,0,1,0,0,1,1)$ et $(0,0,0,1,1,1,1)$.

4. **Quel est le nombre minimal de différences entre deux personnages distincts du jeu ?**

   Il y en a 3.

5. **Pourquoi cela permet-il de deviner le personnage secret, même si le joueur a menti une fois ?**

   Quand on ment à une question, on décrit un personnage (fictif) qui a une seul différence avec le personnage qu'on avait choisi. Et ce personnage fictif a au moins 2 différences avec les 15 autres personnages du jeu. Remarque : si deux personnages du jeu ne présentaient que 2 différences, on pourrait en mentant une fois décrire un personnage qui n'aurait qu'une différence avec ces deux personnages, et on n'aurait aucun moyen de savoir quel était le bon.

   Le niveau jaune s'arrête là mais si l'enfant a envie de continuer, on peut lui proposer la méthode de codage d'un personnage avec des 0 et des 1 présentée dans le niveau orange. On peut également expliquer que les données informatiques sont formées de 0 et de 1 et que lors d'une communication il peut y avoir des erreurs (comme quand on joue au téléphone arabe) qu'on aimerait pouvoir corriger, tout comme on a su trouver un mensonge dans notre jeu.

Voyons maintenant comment jouer à ce jeu avec mensonge, sans l'aide de l'ordinateur !

On représente chaque personnage par une suite de sept chiffres, 0 ou 1. Le troisième personnage, celui qui a des lunettes, pas de moustache, un chapeau, des cheveux, pas de boucle d'oreille, une barbe, pas de nœud papillon, sera ainsi représenté par 1011010**. On code les réponses du joueur aux questions de la même façon, par exemple si j'ai choisi le personnage numéro 3 et que je mens à la question « a-t-il des boucles d'oreille ? », ma réponse sera** 1011110**.**

Remarque : Le code correspondant à une carte est inscrit au dos de la carte.

On reçoit une réponse. Pour savoir si elle est correcte ou mensongère, on va calculer trois nombres, les « syndromes » qui valent chacun 0 ou 1.

- s_1 **:** **on garde un chiffre sur deux** □X□X□X□ **on en fait la somme.** s_1 **vaut 0 si la somme est paire, 1 si elle est impaire.**
- s_2 **:** X□□X□□X**, on fait la somme,** s_2 **vaut 0 si elle est paire, 1 si elle est impaire.**
- s_3 **:** XXX□□□□ **(que les quatre derniers chiffres), on fait la somme,** s_3 **vaut 0 si elle est paire, 1 si elle est impaire.**

Exemple : Si la réponse obtenue est 0011110**, on a**

$$s_1 = 0,\ \ s_2 = 1,\ \ s_3 = 1.$$

Notons H la matrice suivante, à coeffcients dans $\mathbf{F}_2$:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Si la réponse obtenue est $x = (x_1, \ldots, x_7)$, les syndromes associés sont donnés par le produit matriciel Hx.

6. **Le joueur répond sans mensonge. Calculer, pour chaque personnage, s_1, s_2 et s_3.**
   Répartir la tâche dans le groupe, chacun en calcule 1 ou 2 ! Dans tous les cas, on trouve $s_1 = s_2 = s_3 = 0$.
   En effet, le sous-espace vectoriel des (codes des) personnages du jeu est exactement le noyau de H (on a construit H pour cela).

7. **Que valent s_1, s_2 et s_3 si on ment à la première question et pas aux autres ?**
   Il va être important de réussir à faire comprendre que la réponse à cette question ne dépend pas du personnage choisi.
   Pour le calcul de s_1, on regarde ☐✗☐✗☐✗☐. On a vu que si on ne mentait pas, s_1 valait 0. Si on ment à la première question, on change la valeur du premier bit (0 si c'était 1, 1 si c'était 0) et pas des autres, et s_1 vaut donc 1.
   Pour le calcul de s_2, ✗☐☐✗☐☐✗. Si on ne ment pas, s_2 vaut 0. Si on ment à la première question, on change la valeur du premier bit (0 si c'était 1, 1 si c'était 0) et pas des autres, ce qui ne change pas s_2 qui vaut toujours 0.
   Pour s_3, comme pour s_2, une modification du premier bit ne change rien, on a toujours $s_3 = 0$.

   Explication en termes d'algèbre linéaire (sauf avec un expert, on évite bien entendu de parler de cette explication) : si le personnage choisi est $m \in \mathbf{F}_2^7$ et qu'on ment à la question i, le personnage fictif décrit est $m + e_i$ où e_i désigne le i-ième vecteur de la base canonique. Le syndrome obtenu est donc $H(m+e_i) = Hm + He_i = He_i$ qui est égal à la i-ième colonne de H. En particulier, si on ment à la première question, $s_1 = 1$, $s_2 = 0$, $s_3 = 0$.

8. **Et si on ment seulement à la cinquième question ?**
   Dans ce cas, (s_1, s_2, s_3) est égal à la cinquième colonne de H. On peut donner une explication du même type qu'à la question précédente, pour trouver $s_1 = 1$, $s_2 = 1$ et $s_3 = 1$.

9. **Comment le tableau ci-dessous permet-il de découvrir facilement à quelle question le joueur a menti ?**
   On calcule les syndromes. S'ils ne valent pas tous 0, on regarde le numéro de la colonne correspondant, il donne le numéro de la question à laquelle on a menti. Si les syndromes sont tous les trois nuls, le joueur n'a pas menti. On a construit une sorte de détecteur de mensonge !

	1	2	3	4	5	6	7
s_1	1	0	1	0	1	0	1
s_2	0	1	1	0	1	1	0
s_3	0	0	0	1	1	1	1

Le principe sur lequel repose ce jeu est en fait à la base d'applications très utiles, pour transmettre ou stocker des données. Disons que les données transmises sont des paquets de 0 et de 1. Lors de la transmission, que ce soit par ondes radio, par câble ou autre, des erreurs sont susceptibles de se produire et le message reçu n'est plus parfaitement conforme à celui qui a été envoyé (les erreurs jouent le rôle des mensonges de notre jeu). En envoyant, en plus de l'information de départ, une information supplémentaire

redondante, on va être capable de dire s'il y a eu ou non des erreurs lors de la transmission et de les corriger s'il n'y en a pas eu trop. C'est exactement le même principe que quand des pilotes d'avion disent « Alpha, Charlie, Tango » pour dire « A, C, T » mais avec plus d'algèbre !

Voici quelques informations supplémentaires sur ce qu'est un code correcteur. L'espace vectoriel $\mathbf{F}_2{}^n$ *est muni de la distance suivante (dite de Hamming) : la distance entre deux vecteurs est égale au nombre de composantes qui diffèrent entre ces deux vecteurs. Un code correcteur est un sous-espace vectoriel* $\mathscr{C}$ *de* $\mathbf{F}_2{}^n$ *(ici* $n = 7$ *et le sev est de dimension 4) et on définit sa* distance minimale d *comme la plus petite distance entre deux vecteurs distincts de* $\mathscr{C}$*. Les éléments du code* $\mathscr{C}$ *sont les messages que l'on peut envoyer (via un canal). Si* t *erreurs se produisent lors de la transmission, c'est-à-dire si* t *bits sont modifiés, le mot reçu est à distance* t *du mot qui a été envoyé. Si la distance minimale vérifie* $d \geqslant 2t + 1$*, les boules de centre les mots de* $\mathscr{C}$ *et de rayon* t *ne s'intersectent pas donc le mot reçu est dans une seule de ces boules et on peut déterminer quel mot a été envoyé. Dans notre exemple* $d = 3$ *et on peut corriger* $\lfloor \frac{3-1}{2} \rfloor = 1$ *erreur.*

10. Comment proposeriez-vous de modifier ce jeu pour autoriser jusqu'à deux mensonges ?

Cette question est très ouverte et pas facile. Des premiers éléments de réponse peu précis sont déjà valables : il faudrait ajouter des caractéristiques (combien ?). Il faudrait qu'entre chaque personnage du jeu il y ait au moins 5 différences. Aller jusqu'à construire le jeu est plus délicat et il y a de nombreuses manières de procéder. En voici une, toujours à l'aide d'un code de Hamming.

Dans le jeu précédent, les colonnes de la matrice H*, les différents syndromes possibles, étaient tous les vecteurs non nuls de* $\mathbf{F}_2{}^3$*. Ajoutons une ligne à notre matrice et mettons dedans, en colonnes, tous les vecteurs non nuls de* $\mathbf{F}_2{}^4$*, ce qui fait* $2^4 - 1 = 15$ *colonnes. Nous sommes ainsi en train de prévoir un jeu avec 15 questions, 15 caractéristiques par personnage (on pourrait leur ajouter des chaussettes, des gants, des chaussures, une ceinture etc. mais on va se contenter de numéroter les caractéristiques qui s'appelleront juste 1,2,. . .,15). Plusieurs questions se posent encore, par exemple :*

(a) Combien de personnages y aura-t-il dans notre jeu ?

(b) Quels personnages exactement faut-il dessiner, parmi les $2^{15} = 32\ 768$ *possibles ?*

(c) On sait quelles seront les colonnes de la matrice, mais peut-on les ranger dans n'importe quel ordre ?

Réponses dans le désordre :

(c) N'importe quel ordre convient pour les colonnes de H*. Le produit avec le* i*-ième vecteur de la base canonique donnera toujours la* i*-ième colonne, donc le syndrome permettra de déterminer le numéro de la question mensongère.*

(b) Les personnages qu'il faut dessiner dans le jeu dépendent de la matrice H*, ce sont exactement ceux qui sont dans son noyau.*

(a) La matrice H *est de rang 3, d'après le théorème du rang son noyau est de dimension* $15-4 = 11$*. Il y aura donc* $2^{11} = 2\ 048$ *personnages dans le jeu !*

Par exemple on peut prendre :

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

et les personnages de jeu sont tous les éléments du $\mathbf{F}_2$*-espace vectoriel dont une base est (par exemple) donnée par les 11 vecteurs suivants :*
(100000000001111)
(010000000001110)
(001000000001101)
(000100000001100)
(000010000001011)
(000001000001010)
(000000100001001)
(000000010000011)
(000000001000111)
(000000000100110)
(000000000010101).