Le casse-tête mathématique de

Il paraît qu'en ville, on n'est jamais à plus de quelques mètres d'un rat. Mais de nos jours, il est encore plus probable que l'on ne soit jamais à plus de quelques mètres de quelqu'un qui joue à Candy Crush Saga. C'est actuellement le jeu le plus populaire sur Facebook. Il a été téléchargé et installé sur des téléphones, des tablettes et des ordinateurs plus d'un demi-milliard de fois. Essentiellement sur la base de ce succès, son développeur Global King a récemment été introduit à la Bourse de New York avec une valorisation initiale de plusieurs milliards de dollars. Pas mal pour un petit jeu consistant simplement à échanger des bonbons virtuels pour former des chaînes d'au moins trois pièces identiques!

Une grande partie de l'attrait de Candy Crush pour les joueurs est liée à la complexité qui sous-tend ce passe-temps apparemment si simple. De façon surprenante, le jeu est aussi intéressant pour les chercheurs: il apporte un éclairage original sur l'un des problèmes ouverts les plus importants des mathématiques, ainsi que sur la sécurité des systèmes informatiques.

J'ai récemment démontré que Candy Crush est, sur le plan calculatoire, un casse-tête difficile à résoudre *(voir la bibliographie)*. Pour ce faire, j'ai dû recourir à l'un des concepts les plus beaux et les plus importants de l'informatique: l'idée de réduction de problème. Il s'agit de transformer un problème en un autre, ou, comme les informaticiens aiment à dire, réduire un problème à un autre. Essentiellement, ce concept découle de la polyvalence du code informatique: on peut utiliser le même type de code pour résoudre plus d'un problème, même si les variables diffèrent. Si le problème initial est difficile, alors le problème auquel on le réduit est au moins aussi difficile. Le second problème ne peut pas être plus facile, parce que tout programme informatique qui le résout résout aussi le premier. Et si l'on peut prouver l'inverse, c'est-à-dire que le second problème peut également être réduit au premier, alors les deux problèmes sont d'une certaine manière aussi difficiles l'un que l'autre, et prennent autant de temps à résoudre.

Déterminer la difficulté d'un problème est fondamental. Si l'on peut classer un problème en fonction de la difficulté à le résoudre, on sait avec quelle puissance de calcul l'attaquer, et on peut éventuellement déterminer que ce n'est même pas la peine d'essayer de le résoudre. À certains égards, au moins pour les mathématiciens, se pencher sur Candy Crush en tant que problème mathématique peut être aussi addictif que d'y jouer.

Candy Crush

Derrière ce jeu tout simple en apparence se dissimulent des problèmes calculatoires difficiles. C'est probablement pourquoi Candy Crush est aussi addictif.

Toby Walsh

L'ESSENTIEL

- Candy Crush est un jeu électronique pratiqué par des centaines de millions de personnes.
- L'auteur a proposé en 2014 une preuve que ce jeu appartient à une classe de problèmes difficiles sur le plan calculatoire, les problèmes « NP-difficiles ».
- Pour ce faire, Toby Walsh a établi l'équivalence de Candy Crush avec un problème de logique connu pour être de cette classe.
- La complexité de Candy Crush contribue probablement à l'attrait de ce jeu et à son caractère addictif.

Dans notre analyse de Candy Crush, mes collègues et moi sommes partis de la classe la plus célèbre de problèmes présentant des défis calculatoires, les problèmes dits NP (pour *Nondeterministic Polynomial time*, « temps polynomial non déterministe »), le « temps » se référant à la durée requise pour la résolution.

Solutions difficiles, vérifications faciles

Par définition, la classe NP contient tous les problèmes pour lesquels, si une solution est proposée, on peut rapidement vérifier si elle est correcte, en un temps qui est une fonction polynomiale de la taille des données du problème (par exemple, un temps proportionnel à N^{12}, où N est la taille des données). En revanche, l'étape consistant à trouver la solution d'un problème de classe NP est parfois un défi calculatoire.

Appartiennent à cette classe de nombreux problèmes mathématiques célèbres, tels que déterminer si une formule logique complexe peut être satisfaite, ou si un graphe peut être colorié de telle façon que des nœuds voisins aient des couleurs différentes.

Au-dessous de la classe NP, en termes de complexité, on a la classe P (incluse dans la classe NP) de problèmes « faciles » sur le plan du calcul. Dans ce cas, le P signifie simplement « polynomial ». La classe P contient des problèmes tels qu'ordonner une liste ou trouver un item dans une base de données. Le temps que mettra un programme informatique efficace à résoudre de tels problèmes est court, même dans les pires cas. Plus précisément, le temps de résolution d'un problème de classe P est un polynôme en N, la taille des données du problème. Par exemple, un célèbre algorithme de tri, le tri à bulles, fait remonter progressivement les plus grands éléments d'une liste par une succession de comparaisons deux à deux. Ce processus prend un temps qui augmente comme le carré du nombre d'éléments de la liste. Même si l'on doublait la longueur de la liste, l'algorithme mettrait dans le pire des cas quatre fois plus longtemps (ce « pire cas » se produit quand la liste est dans l'ordre inverse).

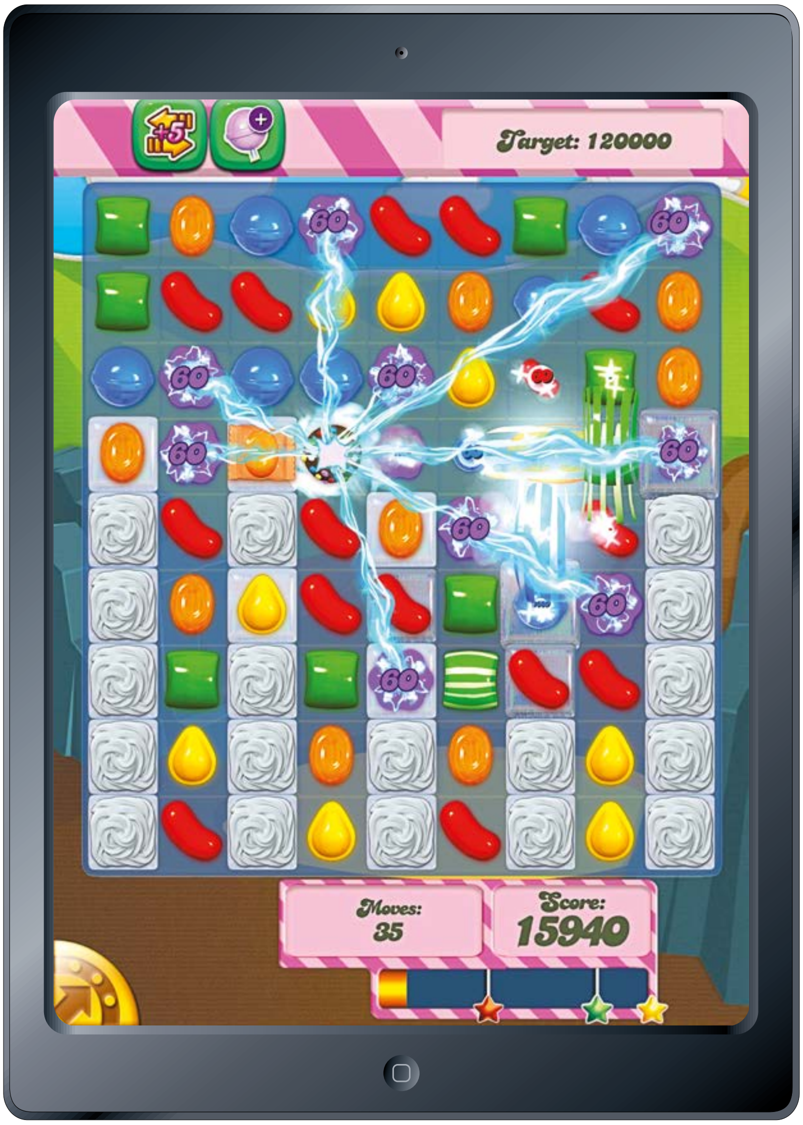
Au-dessus du niveau NP de complexité figurent les problèmes extrêmement difficiles sur le plan calculatoire. Il existe même des problèmes pour lesquels notre modèle de calcul standard, celui que tous les ordinateurs implémentent, est inadéquat. Pour de tels problèmes, il n'existe pas de programme informatique dont on ait l'assurance qu'il va s'arrêter de tourner et donner un résultat. Ces exemples s'inscrivent dans la classe des problèmes dits indécidables. Cette classe inclut des questions telles que décider si un programme informatique va s'arrêter plutôt que de continuer à tourner indéfiniment en boucle, ce que les informaticiens nomment le « problème de l'arrêt ». Alan Turing, pionnier de l'informatique, a prouvé que le problème de l'arrêt est indécidable : il n'existe aucun programme informatique qui puisse, pour tout programme, indiquer s'il s'arrêtera ou non.

La classe NP se situe juste à la frontière entre « facile » et « difficile ». On y trouve, notamment, de nombreux problèmes représentant un défi, tels que l'organisation des tournées de livraison de colis, du tableau de service pour un hôpital ou de l'emploi du temps d'un lycée. Il s'avère que gagner à Candy Crush s'inscrit dans ce sous-ensemble de NP. L'un quelconque de ces problèmes peut être réduit à n'importe quel autre d'entre eux. En ce sens, ils sont tous d'égale difficulté.

■ L'AUTEUR

Toby WALSH est chercheur au NICTA (National Information Communications Technology Australia) et professeur d'intelligence artificielle à l'université de Nouvelle-Galles du Sud, en Australie.

Article publié avec l'aimable autorisation de la revue *American Scientist*.



Avec l'aimable autorisation de Global King

UNE BOMBE DE COULEUR DÉTRUIT TOUS LES BONBONS VIOLETS de la grille sur cette capture d'écran d'une partie de Candy Crush Saga. L'une des raisons pour lesquelles le jeu est tellement prenant est qu'il est en fait vraiment difficile, au sens mathématique du terme.

Un calcul plus long que l'âge de l'Univers ?

Malheureusement, les meilleurs programmes informatiques dont nous disposions pour les problèmes NP ont un temps d'exécution qui augmente énormément avec la taille des données du problème. Sur mon ordinateur de bureau, j'ai un programme qui met quelques heures à trouver les tournées optimales pour $N = 10$ camionnettes de livraison et à démontrer que cette solution est la meilleure possible. Mais pour $N = 100$ camionnettes, le même programme devrait tourner pendant plus de temps que l'âge de l'Univers. En fait, le temps d'exécution de mon programme est une fonction exponentielle de la taille N du problème, et les exponentielles ont une croissance extrêmement rapide.

Bien que les informaticiens soient généralement d'accord avec moi pour dire que les problèmes NP sont à la limite entre facile et difficile, il n'y a aucun moyen de savoir avec certitude, pour un problème spécifique, de quel côté il se situe. Les meilleurs programmes informatiques actuellement à notre disposition mettent un temps exponentiellement long à résoudre

Photographie reproduite avec l'aimable autorisation de CBS

***ELEMENTARY*, une série télévisée formant une adaptation moderne de Sherlock Holmes, illustre comment, bien au-delà de Candy Crush, les problèmes NP se sont fait une place dans la culture populaire. Dans l'épisode dont cette image est extraite, deux mathématiciens ont caché à leurs rivaux leurs recherches sur la question « P = NP ? » en faisant leurs calculs avec des marqueurs ultraviolets, comme le découvre Holmes *(ci-dessus)* après leur assassinat lié à leurs résultats pionniers.**

les problèmes de la classe NP. Mais nous ignorons s'il existe un algorithme (à découvrir) qui saurait résoudre les problèmes NP efficacement, en un temps polynomial. Les mathématiciens expriment cette question en la résumant : « P est-il égal à NP ? » En fait, c'est l'un des problèmes ouverts les plus importants des mathématiques d'aujourd'hui. L'institut Clay de mathématiques a même promis, depuis 2000, une récompense d'un million de dollars à quiconque le résoudra.

Dans le dernier sondage interrogeant les informaticiens pour savoir s'ils pensaient que P = NP est vrai, 83 % déclaraient que non. Autrement dit, ils pensent qu'il n'existe pas d'algorithme efficace pour résoudre les problèmes NP et qu'il n'y en aura jamais. On a également demandé à des informaticiens par quel vocable il convenait de désigner les problèmes qui sont aussi difficiles à résoudre que ceux de NP, qu'ils soient ou non dans cette classe. Le mot qui a finalement remporté les suffrages est le plutôt prosaïque « NP-difficile ». Mais la consultation a également mis en évidence des suggestions plus hautes en couleur telles que NP-impractical (NP-pas pratique), NP-tricky (NP-épineux) et NP-hard-ass (NP-dur à cuire)...

L'idée de la réduction de problème est centrale à la question « P = NP ? ». Si nous trouvions un algorithme capable de résoudre efficacement un problème NP-difficile, alors nous pourrions aussi résoudre efficacement tous les autres problèmes de la classe NP. Le monde serait assez différent si ce résultat était un jour atteint.

Du côté des avantages, nous pourrions mener nos vies en gérant plus efficacement notre temps, tout serait magnifiquement optimisé, depuis les tournées de livraison jusqu'aux horaires de vols en passant par les emplois du temps des personnels hospitaliers (et l'on gagnerait régulièrement en jouant à Candy Crush).

La complexité calculatoire est parfois un bienfait

En revanche, il est important que certaines tâches restent difficiles, par exemple le déchiffrement des messages cryptés, si nous voulons que nos mots de passe et nos comptes bancaires restent sûrs. La complexité calculatoire peut être un bienfait autant qu'un fléau. Nous voulons rendre difficile (d'une difficulté prouvable) le décryptage des messages par des pirates informatiques. Et tout autant, nous devons être capables de crypter ces messages facilement. Cet exemple rappelle la définition de la classe NP-difficile : des problèmes dont la solution est facile à vérifier, mais difficile à trouver.

Pour montrer que Candy Crush est un problème mathématiquement difficile, nous pouvions procéder en partant de n'importe quel problème NP-difficile et en le réduisant à Candy Crush. Pour nous simplifier la

▪ BIBLIOGRAPHIE

T. Walsh, **Candy Crush is NP-hard,** prépublication arXiv du 11 mars 2014 (http://arxiv.org/abs/1403.1911).

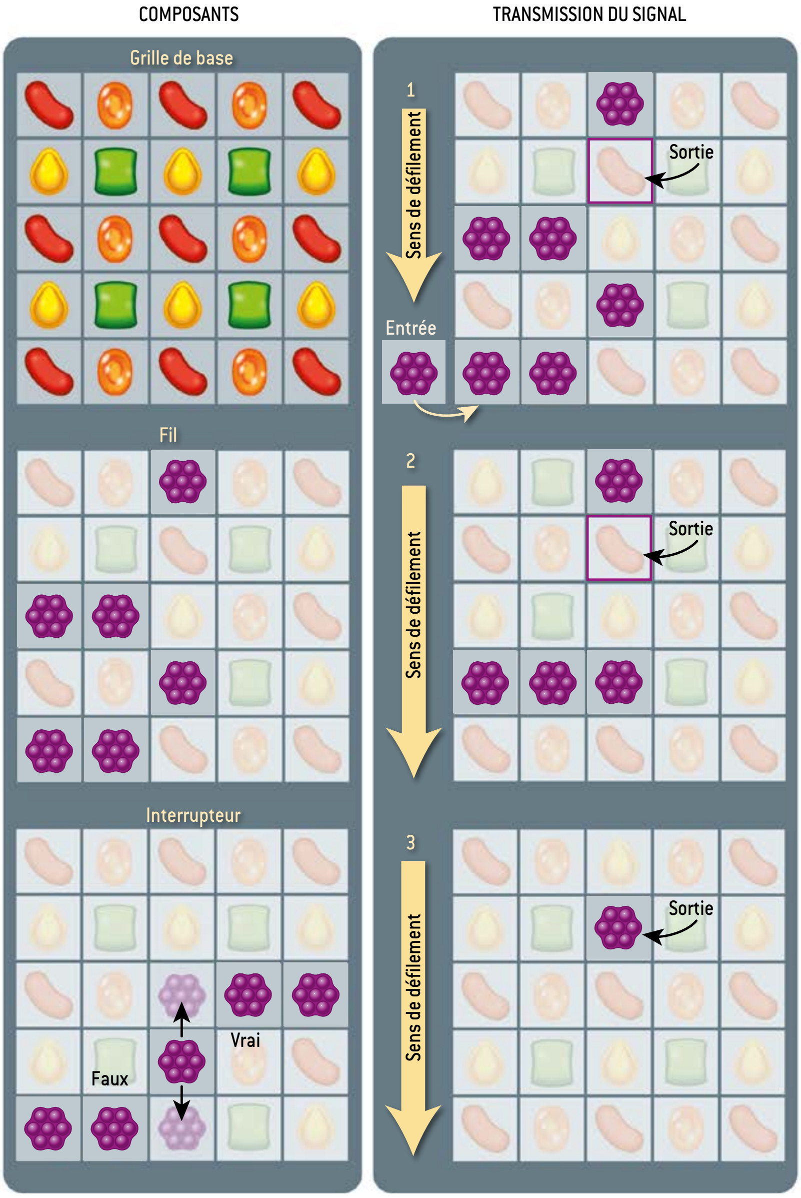
G. J. Woeginger, **The P-versus-NP page** (www.win.tue.nl/ffgwoegi/P-versus-NP.htm).

J.-P. Delahaye, **Complexités,** Belin/Pour la Science, 2006.

IMITER DES CIRCUITS ÉLECTRIQUES AVEC DES BONBONS

Afin de prouver que Candy Crush appartient à la classe des problèmes NP-difficiles, Toby Walsh a traduit ce jeu en un problème de logique appartenant à cette même classe, en concevant un circuit électrique constitué de bonbons. Le principe consiste à partir d'une grille de base *(à gauche en haut)* appropriée, dans laquelle on place des bonbons violets en fonction du composant électrique que l'on cherche à imiter.

On peut former ainsi un « fil » *(à gauche au milieu)* qui transmet un signal, un « interrupteur » *(à gauche en bas)* qui détermine quel fil sera utilisé, etc. Dans le « fil », par exemple, la propagation du signal *(dessins de droite)* commence par le placement d'un bonbon violet sur une case « Entrée » *(1)*. Cela crée un alignement de trois bonbons identiques, qui est effacé. Les bonbons situés au-dessus se décalent vers le bas, créant un nouvel alignement de trois bonbons *(2)*. Quand cet alignement est effacé, un bonbon violet tombe dans la case de sortie : le signal a été transmis *(3)*.



Barbara Aulicino

tâche, mes collègues et moi sommes partis de l'ancêtre de tous les problèmes NP-difficiles, à savoir trouver une solution à une formule logique. C'est ce qu'on appelle le « problème de la satisfaisabilité ».

Vous avez forcément résolu un tel problème si vous vous êtes un jour attaqué à un casse-tête logique. Vous devez décider quelles propositions rendre vraies, et lesquelles rendre fausses, afin de satisfaire un ensemble de formules logiques. Par exemple, sachant que « l'Anglais vit dans la maison rouge », « l'Espagnol possède le chien » et « le Norvégien habite à côté de la maison bleue », la proposition « l'Espagnol possède le zèbre » doit-elle être déclarée vraie ou fausse ?

Traduire Candy Crush en un problème de logique

Pour réduire un casse-tête logique à un problème de Candy Crush, nous exploitons le lien étroit qui existe entre la logique et les circuits électriques. On peut en effet représenter n'importe quelle formule logique par un circuit électrique. Les ordinateurs ne sont, après tout, qu'un grand ensemble de portes logiques (ET, OU, NON, etc.) connectées par des conducteurs électriques. Par conséquent, tout ce que nous avons besoin de faire est de montrer que l'on pourrait construire un circuit électrique dans un jeu de Candy Crush.

Tout d'abord, nous avons besoin d'une grille sur laquelle le circuit va être construit. La grille doit être un motif neutre de bonbons où l'ordre des types de bonbons les uns par rapport aux autres ne change jamais *(voir l'encadré ci-contre)*. La configuration choisie ressemble à des feux de circulation : dans les colonnes paires, nous alternons les bonbons haricots rouges et les bonbons au citron jaunes, alors que dans les colonnes impaires nous alternons les pastilles orange et les gommes vertes. De cette façon, même en déplaçant des colonnes vers le haut ou vers le bas, nous ne créerons jamais une chaîne de trois bonbons identiques.

Dans ce cadre, nous insérons les composants électriques, qui sont constitués de bonbons violets à la mûre. Les bonbons violets décalent les autres bonbons, ils ne les effacent pas. En groupant des bonbons violets, on crée des « fils » qui acheminent le signal dans le circuit *(voir l'encadré ci-contre)*, ou des configurations plus complexes selon les besoins.

Par exemple, lorsqu'on place un bonbon violet sur l'entrée du « fil » en bas à gauche de la grille de départ représentée dans l'encadré, on crée une chaîne de trois bonbons violets. Cette chaîne est effacée, conformément aux principes de base du jeu, ce qui déplace vers le bas les bonbons des deux colonnes concernées. Et ainsi de suite, le signal se propage et est transmis : à la fin, un bonbon violet apparaît dans la case de sortie.

Nous avons également besoin d'interrupteurs que l'utilisateur peut actionner pour décider quels fils sont actifs. Ces interrupteurs représentent le choix des valeurs de vérité (vrai ou faux) d'une proposition logique. L'utilisateur déplace un bonbon violet soit vers le haut, soit vers le bas, ce qui va envoyer un signal soit vers la gauche, soit vers la droite *(voir l'encadré)*.

Enfin, on peut construire des portes logiques telles que ET, OU et NON à partir d'autres bonbons violets en utilisant ces composants de base. Il nous reste alors simplement à connecter les interrupteurs à ces portes logiques avec des fils suffisamment longs pour obtenir un circuit électrique qui simule notre formule logique. Le circuit électrique a alors une sortie qui représente la valeur de vérité de la formule logique.

En termes de circuits logiques électriques, jouer à Candy Crush consiste à décider judicieusement sur quels interrupteurs agir de façon que les portes logiques s'activent convenablement et que la sortie soit « vrai ». De cette manière, le problème consistant à satisfaire une formule logique est réduit à résoudre un problème dans Candy Crush. Et puisque satisfaire une formule logique est un problème difficile, il en va de même de la résolution d'une grille de Candy Crush.

On peut également montrer l'inverse. En d'autres termes, on peut réduire un problème Candy Crush à celui de la satisfaction d'une formule logique. Il faut simplement écrire une séquence de formules qui représentent le jeu sur une grille de Candy Crush. Or pour l'essentiel, une description logique de Candy Crush se trouve dans n'importe quel programme qui y joue. Par conséquent, Candy Crush n'est pas plus difficile que l'un ou l'autre des problèmes NP-difficiles, et le jeu est exactement aussi difficile qu'il est difficile de résoudre tous les autres problèmes de la classe NP.

Si nous avions une manière efficace de jouer à Candy Crush, nous aurions un moyen efficace (et prouvable) d'organiser les tournées de livraison, de faire les tableaux de service des hôpitaux et les emplois du temps des lycées. Inversement, si nous avions un moyen efficace de préparer les tournées, faire les tableaux de service et les emplois du temps, alors nous disposerions d'un moyen efficace de jouer à Candy Crush. C'est toute la puissance de la réduction de problème.

Pourquoi ne pas profiter de Candy Crush pour que les joueurs effectuent des calculs longs et utiles ?

La prochaine fois que vous calez pour résoudre une grille de Candy Crush avec le nombre de coups impartis, vous pourrez vous consoler en sachant que c'était un problème mathématiquement difficile à résoudre. De fait, cette caractéristique participe sans doute à l'attrait du jeu et à son caractère addictif. S'il était aussi facile à résoudre que le morpion, par exemple, il ne serait sans doute pas aussi captivant.

Au cœur de tout cela, il y a la belle notion de réduction de problème, qui a permis aux informaticiens de simplifier le labyrinthe des différents problèmes calculatoires en un nombre plus petit de classes fondamentales telles que P et NP, que les informaticiens appellent le zoo de la complexité. Actuellement, ce zoo contient 500 classes de problèmes.

Dans l'hypothèse improbable où l'on démontrerait que P = NP, le nombre de classes distinctes du zoo de la complexité chuterait brutalement. De nombreuses classes se révéleraient être identiques. En revanche, si P ≠ NP, comme le croient la plupart des informaticiens, alors le zoo contient effectivement de nombreuses classes distinctes de problèmes. En fait, le zoo continue à s'élargir. Les zoologistes de la complexité ont récemment introduit de nouvelles classes pour décrire la complexité des problèmes que pourraient résoudre de futurs ordinateurs quantiques.

L'idée de réduction de problème offre une possibilité intéressante pour les mordus de Candy Crush. Peut-être pourrait-on tirer parti des millions d'heures passées par l'humanité à résoudre des grilles de Candy Crush ? En exploitant l'idée de réduction de problème, nous pourrions dissimuler des problèmes calculatoires utiles dans ces casse-tête.

D'autres problèmes calculatoires bénéficient de telles interactions : chaque fois que vous prouvez à un site web que vous êtes une personne et non un robot en résolvant un Captcha (acronyme de *Completely Automated Public Turing test to tell Computers and Humans Apart*, « test de Turing complètement automatisé pour distinguer les ordinateurs des humains » – un tel test vous demande de taper une séquence de lettres et chiffres apparaissant déformés à l'écran), la réponse aide Google à numériser les anciens ouvrages et journaux. Peut-être devrions-nous atteler les casse-tête de Candy Crush à des tâches non moins importantes !

Nos études de Candy Crush nous ont inspiré un profond respect pour ce passe-temps apparemment anodin. Il offre un éclairage original sur l'une des questions ouvertes les plus importantes des mathématiques d'aujourd'hui, et les implications de cette question s'étendent à de nombreuses applications telles que les algorithmes de chiffrement utilisés pour garantir la sécurité des comptes bancaires. Vous pourrez ainsi expliquer ces enjeux supérieurs à votre patron la prochaine fois qu'il vous surprend à essayer de vous hisser un niveau plus haut ! ■