

Exercice I.1 (Dessin)

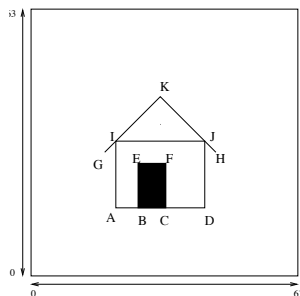
On souhaite effectuer des dessins sur une grille carrée comprenant 64×64 cases (ou pixels). Chaque point est repéré par un couple d'entiers (x, y) où x et y sont compris entre 0 et 63. On supposera que x et y sont écrits en binaire.

- Combien faut-il de bits pour coder un point ?

Solution

Il faut $2 \times \log_2(64) = 2 \times 6\text{bits} = 12\text{bits}$ pour coder un point.

On suppose que les dessins seront composés de trois types d'éléments : des segments, des rectangles dont les côtés sont parallèles aux axes et qui pourront être pleins ou vides. Ainsi, le dessin de la figure ci-dessous est composé d'un rectangle vide ADJI, d'un rectangle plein BCFE et de deux segments GK et KH.



- Un segment AB dont les extrémités ont pour coordonnées (x_A, y_A) et (x_B, y_B) sera représenté par 4 octets $01x_A 01y_A 01x_B 01y_B$.
- Un rectangle vide ABCD dont deux sommets opposés sont A et C sera représenté par 4 octets $10x_A 10y_A 10x_C 10y_C$.
- Un rectangle plein ABCD dont deux sommets opposés sont A et C sera représenté par les 4 octets $11x_A 11y_A 11x_C 11y_C$.
- Un dessin est représenté par la suite des représentations des éléments qui le compose.

On remarque donc qu'on peut savoir si un octet code un élément d'un segment s'il commence par 01, un élément d'un rectangle vide s'il commence par 10 et un élément d'un rectangle plein s'il commence par 11.

- Indiquez quel dessin est représenté par le codage suivant 10001111 10001111 10101111 10101111 01001111 01001111 01101111 01101111 01001111 01101111 01101111 01001111

Solution

Dans l'ordre :

- (a) Rectangle vide 10 : A(15,15) B(47,47)
- (b) Segment 01 : A(15,15) B(47,47)
- (c) Segment 01 : A(15,47) B(47,15)

Cela fait un carré vide avec les diagonales.

- Combien faut-il de bits pour représenter le dessin de la figure ?

Solution

il y a :

- un carré plein BEFC, 11001100 11001100 11011111 11011111
- un vide AIJD, 10001111 10001111 10101111 10101111
- un segment GK, 01001100 01101100 01011111 01111111

– un segment KH. 01011111 01111111 01110010 01101100

. Sachant qu'il y a 4 octets par figures simples. Soit $4 \times 4 = 16$ octets pour représenter la figure.

-
4. On suppose que les $64 \times 64 = 4096$ pixels de la grille sont numérotés selon un ordre conventionnel (par exemple de gauche à droite et de haut en bas). Dans une représentation bitmap, un dessin (en noir et blanc) est représenté par une suite de bits $b_1 \dots b_n$ dont le i -ième bit b_i est égal à 0 si le i -ème pixel du dessin est blanc et 1 s'il est noir.

Combien faut-il de bits pour représenter le dessin de la figure dans une représentation bitmap ?

Solution

La figure en noir et blanc sera de 4096 bits soit 512 octets.

-
5. On souhaite enrichir les codages possibles en représentant directement des lignes brisées $A_1A_2 \dots A_k$ (c'est-à-dire des réunions de segments $A_1A_2, A_2A_3, \dots, A_{k-1}A_k$).

Indiquez comment on pourrait étendre le codage ci-dessus pour représenter de telles lignes brisées ? Indication : les extrémités de la ligne pourront être repérées par 01 et les points intérieurs par 00. Combien faudrait-il alors de bits pour représenter le dessin de la figure ?

Solution

La solution est de remplacer les deux segments par une ligne brisée GKH avec la codage : 01001100 01101100 00011111 00111111 01110010 01101100

Ce qui fait économiser 2 octets. Donc 14 octets sont suffisants pour représenter la maison.

Exercice I.2 (Bonus)

En août 2011, Blizzard Entertainment, développeur du jeu vidéo World of Warcraft, a annoncé qu'ils allaient mettre en place un système permettant à chaque personnage du jeu de stocker 80 objets supplémentaires dans un emplacement spécial (une sorte de coffre en banque appelé void storage).

1. Sachant qu'il existe environ 80 000 objets différents dans le jeu, combien faut-il de bits pour coder un objet à l'aide d'un code unique ?

Solution

Puisqu'avec k bits on ne peut faire que 2^k codes différents, si l'on veut coder 80 000 objets, il faut au moins $\log_2(80000)$ bits. On a $16 \leq \log_2(80000) \leq 17$.

Il faut donc 17 bits pour représenter 80000 objets distincts.

-
2. Il y a en ce moment environ 10 millions d'abonnés à World of Warcraft, chaque abonné peut avoir jusqu'à 50 personnages différents. Estimez la quantité totale de mémoire nécessaire (en giga-octets) pour stocker les identifiants des 80 objets supplémentaires sur tous ces personnages.

Solution

Puisqu'il faut 17 bits pour chaque objet, qu'il y a 80 objets par personnage, 50 personnages par joueur et 10 millions de joueurs, il faut au total $17 \times 80 \times 50 \times 10^7$ bits = 68×10^{10} bits. Il y a 8 bits dans un octet et 10^9 octets dans un giga-octet. Il faut donc $\frac{68 \times 10^{10}}{8} = 85 \times 10^9 = 85\text{Go}$.

Après avoir présenté ce nouveau stockage, Blizzard a précisé que seuls les identifiants des objets seront stockés. Les modifications éventuelles que les joueurs peuvent apporter aux objets ne seront pas conservées. Les différentes modifications que l'on peut apporter à un objet sont les suivantes :

- le nom du créateur de l'objet (le nom d'un personnage, d'au plus 12 caractères, on considèrera qu'il faut un octet par caractère) ;
- un enchantement parmi environ 1000 possibilités ;
- au plus trois gemmes serties sur l'objet parmi environ 1000 possibilités pour chacune des gemmes.

La description complète d'un objet est donc la donnée du code désignant l'objet de base parmi les 80 000 possibles, le nom du créateur, un code désignant l'enchantement associé parmi les 1000 possibles et 3 codes désignant chaque gemme parmi les 1000 possibles.

3. Combien d'octets faut-il pour représenter entièrement la description d'un objet avec tous ses paramètres personnalisés ?

Solution

La taille nécessaire pour stocker un objet est la somme des tailles nécessaires pour stocker ses différentes parties :

- le code de l'objet : 17 bits (ou 3 octets si l'on veut un nombre entier d'octets) ;
- le nom du créateur : 12 octets (ou $12 \times 8 = 96$ bits si l'on veut faire le calcul en bits) ;
- l'enchantement : 1000 possibilités, soit $\log_2(1000) \approx 10$ bits ou 2 octets ;
- les gemmes : 1000 possibilités pour chacune, soit 10 bits ou 2 octets chacune et donc un total de 30 bits ou 6 octets pour les trois. Au final, il faut

$$17 + 96 + 10 + 30 \text{ bits} = 153 \text{ bits} \approx 20 \text{ octets}$$

ou bien (si l'on fait le calcul directement en octets avec des arrondis fait dans les évaluations de la taille de chaque partie)

$$3 + 12 + 2 + 6 \text{ octets} = 23 \text{ octets}$$

La valeur obtenue n'est pas exactement la même dans les deux cas car l'estimation en octets fait des arrondis vers le haut plus tôt dans le calcul donc on surestime un peu plus la taille minimale à utiliser. En pratique il est beaucoup plus probable que les différents éléments des objets soient enregistrés sur des octets séparés et donc l'estimation en arrondissant aux octets à chaque étape est probablement plus réaliste.

-
4. Combien de mémoire (en giga-octets) serait nécessaire pour pouvoir mémoriser la description complète de 80 objets sur chacun des personnages du jeu (50 personnages par joueur, 10 millions de joueurs) ? La raison officielle pour laquelle les modifications des objets ne seront pas mémorisées dans le void storage est que cela prendrait beaucoup trop de place.

Solution

Chaque objet nécessite 23 octets, il y a toujours $80 \times 50 \times 10^7$ objets au total soit une mémoire nécessaire de

$$23 \times 80 \times 50 \times 10^7 \text{ octets} = 92 \times 10^{10} \text{ octets} \approx 920 \text{ Go}$$

Pour 153 bits par objets :

$$153 \times 80 \times 50 \times 10^7 \text{ bits} = 6120 \times 10^9 \text{ bits} = 765 \times 10^9 \text{ octets} = 765 \text{ Go}$$

-
5. En comparant la mémoire nécessaire pour stocker entièrement les objets supplémentaires (question précédente) à l'ordre de grandeur de la capacité d'un disque dur couramment

trouvé dans un ordinateur actuel, que pensez-vous de ce commentaire venant d'une entreprise ayant des dizaines de serveurs dédiés à un jeu qui leur rapporte environ un milliard d'euros par an ?

Solution

Aujourd'hui, le disque dur d'un ordinateur personnel standard contient entre 200 Go et 2 To (2000 Go). L'espace total nécessaire pour stocker tous les nouveaux objets est donc de l'ordre de la taille d'un disque dur trouvé dans n'importe quel ordinateur. Sachant que cet espace nécessaire est réparti sur des dizaines de serveurs, la contrainte de place est assez négligeable.

Remarque : Il faut tout de même signaler que les estimations que l'on a faites ici considèrent la place minimale pour stocker des objets, en les représentant de manière minimaliste, les uns à la suite des autres dans la mémoire. En réalité, dans une base de données, il faut ajouter beaucoup d'autres informations (délimiteurs entre les descriptions d'objets, descriptions de certains champs, etc.) et la taille de l'ensemble peut donc augmenter significativement. Cependant, même en considérant une place nécessaire 100 fois plus importante on obtient une taille d'environ 100 To ce qui reste tout à fait acceptable pour une grande entreprise.
