

Corrigé de l'examen, Programmation I
Sciences et Technologies du Vivant, Semestre 1

Mercredi 19 décembre 2007

Question 1 - Commandes Unix

- (a) `mkdir exercices slides`
`mkdir exercices/C exercices/java`
- (b) `cp ../lepetit/cours-C-?.pdf slides`
ou
`cp /home/lepetit/cours-C*.pdf slides`
- (c) `ls -l exercices/C/*.cpp`
- (d) `cd exercices/C`
- (e) `rm conversion.cpp palindrome.cpp`
- (f) `mv complation.cpp compilation.cpp`
- (g) `chmod o+r pointeurs.cpp`

Question 2 - Pointeurs

- (a) 1) 5, 14
2) 5, 14
3) 5, 14
- (b) 1) 1 2 4 8
2) 1 4
3) 2
4) 2 2
5) 2 2
6) 8 2
- (c) 1) 1 2 4 8 16 32
2) 2, 4
3) 18
4) 6 16 8 4 2 5
5) 5 4

Question 3 - Flock of Birds

- ```
(a) Essaim * alloue_essaim(int n)
 {
 Essaim *nouveau = new Essaim;
 nouveau->nb_oiseaux = n;
 nouveau->oiseaux = new Oiseau[n];
 return nouveau;
 }

(b) void initialise_oiseau(Oiseau * o, int longueur, int hauteur)
 {
 o->x = rand() % longueur;
 o->y = rand() % hauteur;
 float angle = float(rand()) / RAND_MAX * 2 * M_PI;

 o->dx = 2 * cos(angle);
 o->dy = 2 * sin(angle);
 }

(c) void initialise_essaim(Essaim * e, int longueur, int hauteur)
 {
 for (int i=0; i<e->nb_oiseaux; i++)
 initialise_oiseau(e->oiseaux + i, longueur, hauteur);
 }

(d) void regarde_voisinage(Essaim * e, Oiseau * o, int * nb_voisins,
 float * x_moyen, float * y_moyen)
 {
 *x_moyen = 0;
 *y_moyen = 0;
 *nb_voisins = 0;

 for (int i=0; i<e->nb_oiseaux; i++) {
 // On peut stocker le pointeur sur l'oiseau i dans une variable
 // supplémentaire pour éviter de répéter e->oiseaux[i] :
 Oiseau * o2 = e->oiseaux + i;

 // Évite de compter 'o' parmi ses propres voisins:
 if (o2 != o) {
 float distance = sqrt((o2->x - o->x) * (o2->x - o->x) +
 (o2->y - o->y) * (o2->y - o->y));

 if (distance < 5) {
 (*nb_voisins)++;
 }
 }
 }
 }
```

```

 *x_moyen += o2->x;
 *y_moyen += o2->y;
 }
}

if (*nb_voisins > 0) {
 *x_moyen /= *nb_voisins;
 *y_moyen /= *nb_voisins;
}
}

(e) void bouge_oiseau(Essaim * e, Oiseau * o)
{
 int nb_voisins;
 float x_moyen, y_moyen, norme;

 regarde_voisinage(e, o, &nb_voisins, &x_moyen, &y_moyen);

 if (nb_voisins > 0) {
 o->dx = o->x - x_moyen;
 o->dy = o->y - y_moyen;

 norme = sqrt(o->dx*o->dx + o->dy*o->dy);

 if (nb_voisins < 5) {
 o->dx *= -2 / norme;
 o->dy *= -2 / norme;
 } else {
 o->dx *= 2 / norme;
 o->dy *= 2 / norme;
 }
 }

 o->x += o->dx;
 o->y += o->dy;
}

(f) void affiche_essaim(SimpleWindow * window, Essaim * e)
{
 for (int i=0; i<e->nb_oiseaux; i++)
 window->drawPoint(int(e->oiseaux[i].x), int(e->oiseaux[i].y));
}

```