

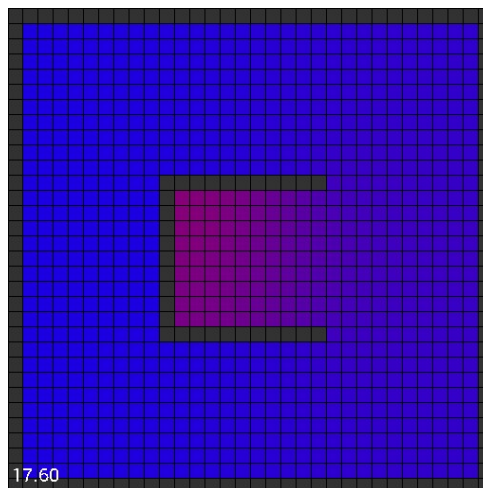
TP noté (1h30min)

Aucun document autorisé.

Récupérez le code départ dans TOMUSS dans la case « TPnoté/CodeInitial ». Vous coderez bien entre les balises sans toucher au reste. Créez un projet Grapic en l'ajoutant dans le fichier « .lua » ou codez directement dans le projet *Start* avec un copier/coller. Vous pouvez regarder tous les programmes Grapic (tutoriaux et exemples) pour trouver de l'inspiration ou comprendre une fonction mais rien d'autre.

A la fin du TP, vous déposerez sur TOMUSS le fichier `apps/lifami/main_VotreProjet.cpp` ou `apps/start/main_start.cpp`.

Nous souhaitons réaliser une application simulant le transfert de chaleur dans une grille régulière. Le monde affiché dans la fenêtre est découpé en un tableau 2D de Parcelle (cellule). Une parcelle est soit de l'air, soit un mur. La chaleur se propage uniquement entre les parcelles d'air. Chaque parcelle de type mur est un rectangle noir. Une parcelle de type air est un rectangle dont la couleur est un dégradé entre du bleu pour la température minimale et du rouge pour la température maximale. La chaleur se propage dans l'air d'une cellule vers ses voisines. Dans l'exemple ci-dessous, la source de chaleur est dans le coin en haut à gauche des 3 murs du milieu.



- 1) Ecrivez la définition de deux structures *Parcelle* et *World*.

2 pts

Une cellule *Parcelle* de la grille du monde comporte un entier *type* et un réel *t*. *type* ne peut prendre comme valeur que AIR ou WALL, deux constantes définies dans le code initial. *t* est la température de la Parcelle (réel).

Le monde *World* compte un tableau à 2 dimensions de *Parcelle* de taille MAX_HEATDIM et deux entiers *dx* et *dy* correspondant à la taille réelle de la simulation.

3 pts

- 2) Ecrivez la procédure *rectangleCell* qui remplit dans *World* un rectangle de *Parcelle* allant du coin en bas à gauche (*xm,ym*) au coin en haut à droite (*xM,yM*) avec le type *type* et la température *t*. Les bornes sont incluses dans le remplissage.

```
void rectangleCell(World& w, int xm, int ym, int xM, int yM, int type, float t)
```

2 pts

- 3) Ecrivez la procédure *init* qui initialise un monde *World* de taille 32x32 et le remplit d'air de température 0 degré. Initialisez un petit rectangle quelque part avec une température de 20 degrés. Pour l'instant il n'y a pas de mur. Utilisez 2 fois la procédure de la question 2.

2 pts

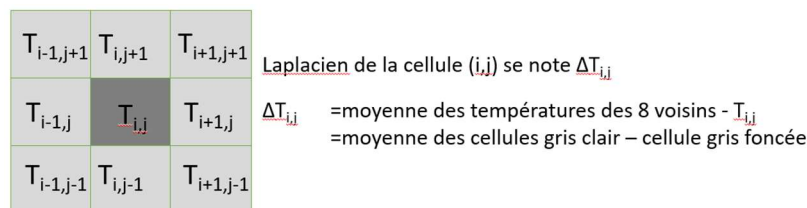
- 4) La fonction *temperatureToColor* renvoie une couleur *Color* à partir d'une température (réel). Complétez cette fonction pour qu'elle calcule l'interpolation entre du bleu pour la température *TEMP_MIN* et du rouge pour la température *TEMP_MAX*.

4 pts

- 5) Ecrivez la procédure *Draw* qui affiche toutes les *Parcelle* d'un monde *World* pris en paramètre. Le principe est comme pour le jeu de la vie, il faut calculer la largeur d'une colonne et la hauteur d'une ligne ; faire une double boucle sur les lignes et colonne ; puis dessiner un *rectangleFill* sur chaque case avec comme couleur l'appel de la fonction de la question 4.

3 pts

- 6) L'équation de la chaleur définie en physique indique que la température d'une *Parcelle* va tendre vers la température moyenne des *Parcelle* autour. On calcule donc la différence entre la température moyenne des *Parcelle* autour et la température de la parcelle du milieu. Ce calcul est appelé le *Laplacien* et se note ΔT_{ij} avec T_{ij} la température de la parcelle (i,j) .



Ecrivez la fonction *laplacien* qui renvoie le Laplacien d'une parcelle (i,j) . Attention au bord du monde.

```
float laplacien(World w, int i, int j)
```

4 pts

- 7) Ecrivez la fonction *update* qui calcule la mise à jour de la température. La mise à jour de la température se calcule par l'équation différentielle suivante

$$\frac{dT_{ij}(t)}{dt} = \Delta T_{ij}(t)$$

L'approximation des différences finies (même calcul que pour les particules) conduit au fait que la nouvelle température est égale à l'ancienne température plus le Laplacien multiplié par la constante dt :

$$\frac{T_{ij}(t+dt) - T_{ij}(t)}{dt} = \Delta T_{ij}(t)$$

$$\Rightarrow T_{ij}(t + dt) = T_{ij}(t) + dt \cdot \Delta T_{ij}(t)$$

Attention comme pour le jeu de la vie, il faut bien veillez à garder une copie de l'ancien monde pour calculer le Laplacien et changer la température sur le nouveau monde.

- 8) (BONUS qui donnera droit à toute notre admiration). Ajoutez la gestion des murs dans toutes les fonctions précédentes : initialisation, affichage, mise à jour de la température. Pour simplifier les murs ont une température constante et n'influent sur la température de l'air.