

Projet de programmation avancée



A faire en binôme

Faire un remake du jeu 'Tempest' de Atari (1981) qui sera le plus fidèle possible à la version originale. La programmation se fera en C++ *moderne* et l'affichage avec la librairie SDL2. Un code d'exemple C permettant de tracer des lignes sera fourni.

Les textes (menus, score, etc) seront tracés avec les fontes de Hershey

<http://paulbourke.net/dataformats/hershey/>

Vous trouverez les détails et règles du jeu sur <https://strategywiki.org/wiki/Tempest>. Pour trouver des vidéos sur youtube, cherchez « tempest atari ».

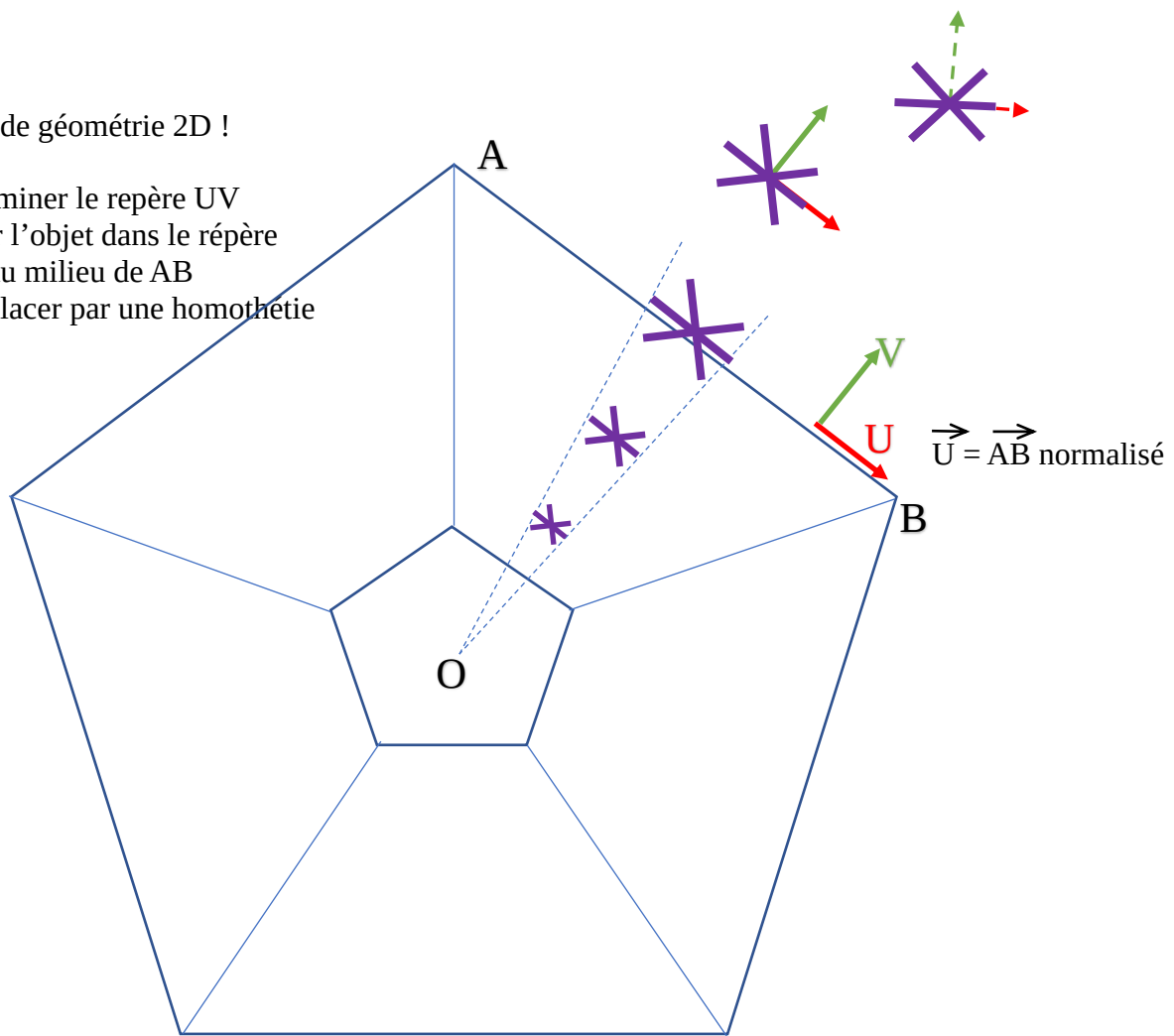
L'évaluation prendra en compte le contenu, mais surtout la modernité (pas de code à la C), la propreté, l'organisation et l'élégance du code.

Vous respecterez les critères suivants :

- Utilisez les possibilités du C++11/14/17/20 (lambda, for range, ...)
- Ecrivez des classes pour vos objets et encapsulation de la SDL2
- Utilisation de l'héritage et du polymorphisme.
- Utilisation de const et des références autant que possible.
- Pas de tableau ni statique ni dynamique, utilisez les containers de la STL
- Utilisation d'algorithmes de la STL
- Pas de pointeurs utilisez *unique*, *shared* et *weak_ptr*
- Pour les nombre aléatoires pas de rand()/srand() utiliser *std::uniform_real_distribution* et *std::uniform_int_distribution*
- Documentez votre code
- Utilisez une convention de codage

Un peu de géométrie 2D !

- 1 Déterminer le repère UV
- 2 Placer l'objet dans le repère
- 3 Puis au milieu de AB
- 4 le déplacer par une homothétie



Matrice M du repère (U,V)

$$\begin{pmatrix} U_x & V_x \\ U_y & V_y \end{pmatrix} = \begin{pmatrix} U_x & -U_y \\ U_y & U_x \end{pmatrix}$$

Mettre un point dans le repère $P' = M \cdot P$:

$$\begin{pmatrix} P_x \\ P_y \end{pmatrix} \begin{pmatrix} U_x & -U_y \\ U_y & U_x \end{pmatrix} = \begin{pmatrix} P'_x \\ P'_y \end{pmatrix}$$

$$\begin{aligned} U_x P_x - U_y P_y &= P'_x \\ U_y P_x + U_x P_y &= P'_y \end{aligned}$$

I suffit ensuite d'additionner la position du polygone (au milieu du segment AB)

Une homothétie (simple multiplication par un réel h) permet ensuite de déplacer chaque point P sur sa bande vers le fond.

Pour être précis il faudra adapter l'homothétie (vitesse 2D), plus lente au centre qu'à la périphérie due à la perspective. Le facteur d'homothétie devra donc évoluer suivant z^2 , avec $z=0 \rightarrow h=1$ (à la périphérie) et $z=d \rightarrow h=h_0$ (au centre) h_0 étant l'homothétie de la forme centrale / forme extérieur
La valeur de d étant la profondeur du tunnel. h_0 et d définissent le ratio de perspective (focale).
 $h = 1 - a \cdot z^2$ avec $a = (1-h_0)/d^2$. Si on fixe arbitrairement $d=1$ on a donc :
 $h = 1 - (1-h_0) \cdot z^2$ avec z dans $[0,1]$ pour le tube du (bord vers le fond).