

Marques Da Silva Bruno
Kidnapillai Banugopan
Varathalingam Mayouran
Soulié Nicolas
Callegari Thomas
Lerendu Eric

Equipe : Praise the blood

1A S2 TOP EFREI P2022 Projet n°3, 4 : L'arène

Jeu : Tron

Objectif :

L'objectif de ce projet était de créer un jeu en 2D. Le choix du jeu ne nous a pas été imposé, nous avons donc choisi de créer un jeu à la fois divertissant et accessible à tous.

Tron nous est venu à l'esprit à la suite du visionnage des 2 films Tron ainsi que des jeux vidéo inspirés de celui-ci.

Ce jeu affronte plusieurs motos de couleurs différentes qui possèdent chacune une trainée qui leur permettent de tuer leurs adversaires. En effet, chaque trainée est mortelle ainsi que la limitation de la map.

Sommaire :

- Règles du jeu
 - Contrainte
 - Stratégie
- Code
 - Contextualisation
 - Fonction
 - Rendu
- SDL
 - Téléchargement
 - Application des fonctions



Règles du jeu :

Les règles sont simples, 32 motos représentatives d'une IA et à la fin il n'en restera qu'un.

Ces dernières circuleront sur une map de taille variable selon le nombre de motos au départ. Cette map est limitée par des bords car en effet une IA ne pourra pas sortir d'un côté de la map et en revenir de l'autre.

Aucun retour en arrière n'est possible car en effet comme, dit précédemment, la trainée est mortelle puisque si une moto traverse la trainée d'une autre moto ou encore la sienne, elle meurt à l'instant même. Cette dernière augmentera au fur et à mesure du temps qui est lui en tout par tour. Or ce tour par tour ne sera pas visible dû à une grande vitesse entre le temps d'action de chaque joueur. Ainsi nul ne s'en rendra compte.

Lorsqu'il ne restera que 2 motos en course et qu'une collision s'opère entre les deux au même moment nous avons une égalité. Pour cela nous avons mis en place un système pour surpasser ce problème. Le principe est de compter le nombre de kills d'une IA.

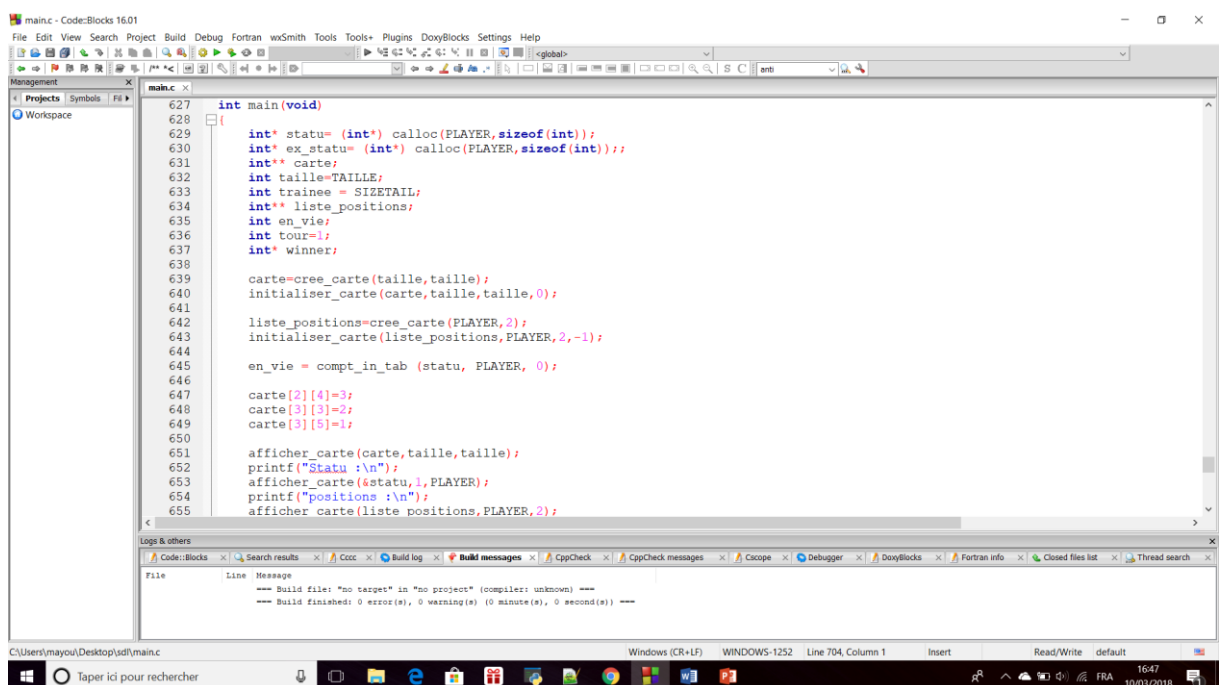
Plusieurs stratégies existent pour en arriver à cette fin :

- L'IA se place au milieu de la map et se déplace en escargot afin d'obtenir le maximum de map possible suite à la disparition de sa propre trainée.
- L'IA peut choisir d'effectuer cette même stratégie mais en commençant par longer les bords.
- Si une IA se trouve à proximité des bords, la stratégie pourrait être de la coincer grâce à notre trainée.

Code :

Dans notre code, la map est représentée par un tableau 2D rempli de 0. De là, les joueurs sont également par leurs chiffres allant de 1 à 32, soit leur numéro de joueur.

Leurs trainées, quant à elles, sont représentées par des chiffres qui augmentent de $+n$, n étant le nombre de joueurs, à chaque fois afin de ne pas confondre les trainées entre elles.



```
main.c - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DoxyBlocks Settings Help
Management
Projects Symbols File
Workspace
main.c
627 int main(void)
628 {
629     int* statu= (int*) calloc(PLAYER,sizeof(int));
630     int* ex_statu= (int*) calloc(PLAYER,sizeof(int));
631     int** Carte;
632     int taille=TAILLE;
633     int trainee = SIZETAIL;
634     int** liste_positions;
635     int en_vie;
636     int tour=1;
637     int winner;
638
639     carte=cree_carte(taille,taille);
640     initialiser_carte(carte,taille,taille,0);
641
642     liste_positions=cree_carte(PLAYER,2);
643     initialiser_carte(liste_positions,PLAYER,2,-1);
644
645     en_vie = compt_in_tab (statu, PLAYER, 0);
646
647     carte[2][4]=3;
648     carte[3][3]=2;
649     carte[3][5]=1;
650
651     afficher_carte(carte,taille,taille);
652     printf("Statu :\n");
653     afficher_carte(&statu,1,PLAYER);
654     printf("positions :\n");
655     afficher_carte(liste_positions,PLAYER,2);
656
657     return 0;
658 }
```

Log & others

Code::Blocks Search results Cccc Build log Build messages CppCheck CppCheck messages Cscope Debugger DoxyBlocks Fortran info Closed files list Thread search

File Line Message

==== Build file: "no target" in "no project" (compiler: unknown) ====

==== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ====

C:\Users\mayou\Desktop\sd\main.c Windows (CR+LF) WINDOWS-1252 Line 704, Column 1 Insert Read/Write default 1647 10/03/2018

Taper ici pour rechercher

main.c - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Symbols File

Workspace

```
654 printf("positions :\n");
655 afficher_carte(liste_positions,PLAYER,2);
656 printf("\n");
657
658 while (tour<=TOUR && en_vie>1)
659 {
660     printf("Tour n'%d\n",tour);
661     free(ex_statu);
662     ex_statu = copy_tab_1D (statu,PLAYER);
663     saisie_deplacement_ia (carte, taille, statu, liste_positions,tour);
664     collision_joueurs (liste_positions, statu);
665     gestion_trainee (carte,taille,trainee);
666     deplacement_ias (liste_positions, carte, taille, statu);
667
668     if (tour%REDUCTION==0 && taille>TAILLE_MIN)
669     {
670         ejectee_carte(liste_positions,taille,statu);
671         carte=reduction_carte(carte,taille);
672         taille=taille-2;
673     }
674
675     afficher_carte (carte,taille,taille);
676     printf("Statu :\n");
677     afficher_carte(&statu,1,PLAYER);
678     printf("Ex statu :\n");
679     afficher_carte(&ex_statu,1,PLAYER);
680     printf("positions :\n");
681     afficher_carte(liste_positions,PLAYER,2);
682 }
```

Log & others

Code::Blocks Search results CMake Build log Build messages CppCheck CppCheck messages Cscope Debugger DoxyBlocks Fortran info Closed files list Thread search

File Line Message

--- Build file: "no target" in "no project" (compiler: unknown) ---

--- Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ---

C:\Users\mayou\Desktop\ud\main.c Windows (CR+LF) WINDOWS-1252 Line 704, Column 1 Insert Read/Write default 1648 10/03/2018

Taper ici pour rechercher

main.c - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Symbols File

Workspace

```
678 printf("Ex statu :\n");
679 afficher_carte(&ex_statu,1,PLAYER);
680 printf("positions :\n");
681 afficher_carte(liste_positions,PLAYER,2);
682
683 tour++;
684 en_vie = compt_in_tab (statu, PLAYER, 0);
685 trainee = SIZETAIL + PLAYER - en_vie;
686 printf("%d joueurs restant\nTrainee de taille %d\n\n",en_vie,trainee);
687
688 if (en_vie<1)
689 {
690     winner = departager_vainqueur (statu, ex_statu);
691     egalite (winner);
692     printf("\nwinner :\n");
693     afficher_carte(&winner,1,winner[0]+1);
694 }
695 else
696 {
697     vainqueur (statu);
698 }
699
700
701
702 return 0;
703
704
705
706 }
```

Log & others

Code::Blocks Search results CMake Build log Build messages CppCheck CppCheck messages Cscope Debugger DoxyBlocks Fortran info Closed files list Thread search

File Line Message

--- Build file: "no target" in "no project" (compiler: unknown) ---

--- Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ---

C:\Users\mayou\Desktop\ud\main.c Windows (CR+LF) WINDOWS-1252 Line 704, Column 1 Insert Read/Write default 1648 10/03/2018

Taper ici pour rechercher

```
110 int** cree_carte(int taille1, int taille2) // Génère une carte de format taille1 x taille2 (sans control sur son contenu)
111 {
112     int** tableau;
113     int i;
114     tableau=(int**) malloc(taille1*sizeof(int*));
115     for (i=0;i<taille1;i++)
116     {
117         tableau[i]=(int*) malloc(taille2*sizeof(int));
118     }
119     return tableau;
120 }
121
122 void afficher_carte(int** carte,int taille1, int taille2) // Affiche une carte 2D (sans couleurs) en console // A modifier quand on aura SDL
123 {
124     int i,j;
125     for(i=0;i<taille1;i++)
126     {
127         for(j=0;j<taille2;j++)
128         {
129             printf(" %d",carte[i][j]);
130         }
131         printf("\n");
132     }
133 }
134
135 void initialiser_carte(int** carte,int hauteur,int largeur,int valeur) // Remplit une carte 2D avec la nombre "valeur"
136 {
137     int i,j;
138     for(i=0;i<hauteur;i++)
139     {
140         for(j=0;j<largeur;j++)
141         {
142             carte[i][j]=valeur;
143         }
144     }
145 }
146 }
```

Log & others

File	Line	Message
Code::Blocks		Build file: "no target" in "no project" (compiler: unknown) ---
		--- Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ---

Windows (CR+LF) WINDOWS-1252 Line 693, Column 25 Insert Read/Write default 17:07 10/03/2018

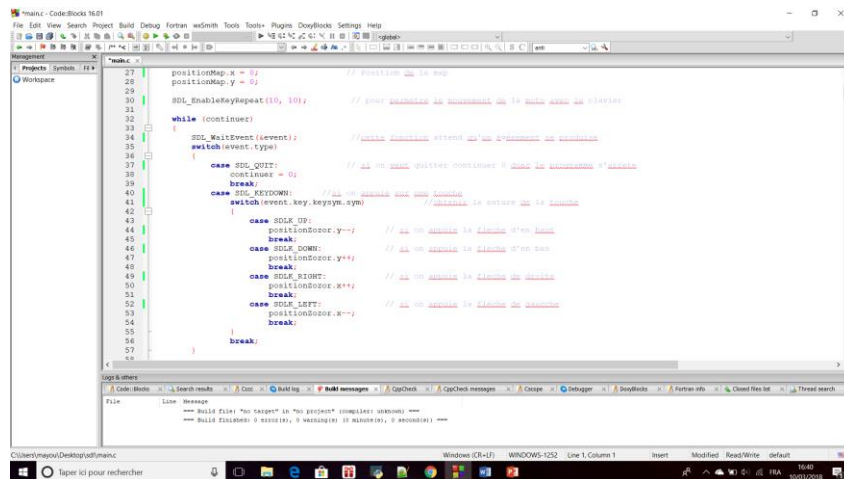
Nous avons créé plusieurs fonctions. Par exemple, la première fonction a pour but d'allouer notre map en prenant en argument le nombre de joueur ainsi que le nombre 2. Elle va donc nous permettre de créer une map à une adresse bien précise.

The screenshot shows the Code::Blocks IDE with the main.c file open. The code is in French and uses SDL for event handling. It features a switch statement for SDL_KEYDOWN events, handling keys like SDLK_UP, SDLK_DOWN, SDLK_RIGHT, and SDLK_LEFT. Comments in French explain the actions, such as moving a 'zozor' character and updating its position. The code also includes SDL_Init, SDL_SetVideoMode, and SDL_LoadBMP functions. The status bar at the bottom indicates the file path as C:\Users\mayou\Desktop\sd/main.c and the window title as 'main.c - Code::Blocks 16.01'.

```
40 case SDL_KEYDOWN: // si on appuie sur une touche
41     switch(event.key.keysym.sym) // ok pour la nature de la touche
42     {
43         case SDLK_UP: // si on appuie la touche d'en haut
44             positionZozor.y--;
45             break;
46         case SDLK_DOWN: // si on appuie la touche d'en bas
47             positionZozor.y++;
48             break;
49         case SDLK_RIGHT: // si on appuie la touche de droite
50             positionZozor.x++;
51             break;
52         case SDLK_LEFT: // si on appuie la touche de gauche
53             positionZozor.x--;
54             break;
55     }
56     break;
57 }
58
59 SDL_FillRect(ecran, NULL, SDL_MapRGB(ecran->format, 0, 255, 255)); // efface l'ecran
60 SDL_BlitSurface(map, NULL, ekran, &positionMap); // blit map
61 SDL_BlitSurface(zozor, NULL, ekran, &positionZozor); // blit le zozor
62 SDL_Flip(ecran); // met a jour l'ecran
63 }
64
65 SDL_FreeSurface(zozor); // free du aux allocation au debut
66 SDL_FreeSurface(map);
67 SDL_Quit();
68
69 return EXIT_SUCCESS;
70
71 }
```

The screenshot shows the Code::Blocks IDE with the main.c file open. The code is in French and uses SDL for initialization and loading. It includes headers for <stdlib.h>, <stdio.h>, and <SDL/SDL.h>. The main function initializes SDL, sets video mode, loads a background image (fond_02.bmp), and loads a zozor image (zozor.bmp). Comments in French explain the steps, such as initializing SDL, setting video mode, and loading images. The code also includes SDL_Init, SDL_SetVideoMode, SDL_LoadBMP, and SDL_SetColorKey functions. The status bar at the bottom indicates the file path as C:\Users\mayou\Desktop\sd/main.c and the window title as 'main.c - Code::Blocks 16.01'.

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <SDL/SDL.h>
4
5
6
7 int main(int argc, char *argv[])
8 {
9
10     SDL_Surface *ecran = NULL, *zozor = NULL, *map = NULL; // pour les pointeurs mettre a NULL les vars sont automatiquement
11     SDL_Rect positionZozor, positionMap;
12     SDL_Event event;
13     int continuer = 1;
14
15     SDL_Init(SDL_INIT_VIDEO); // initialisation de la video charge la video (pour des raisons)
16
17     ekran = SDL_SetVideoMode(800, 600, 32, SDL_HWSURFACE | SDL_DOUBLEBUF); // ouvre une video en 800x600 32 pixels, a ecran double
18     SDL_WM_SetCaption("TRON GAMES", NULL); // nom de la fenetre (define SDL_DOUBLEBUF 0x40000000)
19
20     zozor = SDL_LoadBMP("zozor.bmp"); // charge l'image pour "zozor" ici la map
21     SDL_SetColorKey(zozor, SDL_SRCCOLORKEY, SDL_MapRGB(zozor->format, 255, 255, 255)); // pour rendre transparent l'image de la map
22
23     positionZozor.x = ekran->w / 2 - zozor->w / 2; // position de "zozor" ici la map est la map
24     positionZozor.y = ekran->h / 2 - zozor->h / 2;
25
26     map = SDL_LoadBMP("fond_02.bmp"); // charge l'image de fond
27     positionMap.x = 0; // position de la map
28     positionMap.y = 0;
29
30     SDL_EnableKeyRepeat(10, 10); // pour permettre le mouvement de la map avec la clavier
31
32     while (continuer)
```



Ce bout de code permet d'afficher le rendu suivant :



Par ailleurs, ce programme permet de déplacer la moto en utilisant les flèches du clavier.

De plus, la moto ci-dessus a été dessinée par nos soins :



Pour conclure, ce projet nous a permis d'apprendre de nouveaux concepts en C ainsi qu'en SDL. Même si nous avons rencontré de nombreuses difficultés nous avons su faire à celle-ci notamment en changeant notre manière de travailler en groupe. Malgré notre avancement certains points sont encore à renforcer : Réduction de la taille de map en fonction du temps et quelques points à la finalisation notamment l'apparition de la trainée ainsi que des affichages claires et précises.