## function seconde\_to\_temps(t: LongInt): String;

Transforme le temps (stocké comme un entier, en ms) en une chaine de caractère au format (m minutes, s secondes, ccc centièmes).

On obtient la valeur des minutes, secondes et centièmes via les opérateurs modulo et diviseur, et on assure le format de sortit via l’ajout éventuelle de 0 pour les minutes et secondes.

## procedure ajouter\_physique(var physique: T\_PHYSIQUE\_TABLEAU);

Ajoute un nouvel élément physique au tableau.

La procédure sauvegarde l’ancien pointeur, demande au système l’allocation d’un bloc de mémoire d’une taille supérieur de 1 au précédent, recopie via une boucle l’adresse des anciens éléments physique dans le nouveau bloc, demande et stocke l’adresse du nouvel élément physique dans la dernière case (la ‘nouvelle’ case) et initialise ces valeurs a 0. La procédure libère alors l’ancien bloc de pointeurs, et augmente la taille du tableau de 1.

## procedure ajouter\_enfant(var element: T\_UI\_ELEMENT);

Ajoute un enfant a un élément d’UI.

La procédure sauvegarde l’ancien pointeur, demande au système l’allocation d’un bloc de mémoire d’une taille supérieur de 1 au précédent, recopie via une boucle l’adresse des anciens éléments physique dans le nouveau bloc, demande et stocke l’adresse du nouvel enfant dans la dernière case (la ‘nouvelle’ case) et initialise ces valeurs. La procédure libère alors l’ancien bloc de pointeurs, et augmente le nombre d’enfants de 1.

## function isInElement(element: T\_UI\_ELEMENT; x, y: Integer): Boolean;

Teste si un élément d’UI se situe au coordonnées indiqué (référentiel fenêtre)

En premier, la procédure soustrait récursivement l’ensemble des coordonnées x,y des parents de l’élément au coordonnées des positions demandées, afin de pouvoir par la suite tester si celles-ci se situent dans le rectangle x,x+w,y,y+h de l’élement cible.

## function hitBox(surface: PSDL\_Surface; p: SDL\_Rect; a: Real; colors: array of TSDL\_Color): T\_HITBOX\_COLOR;

Compare les couleurs d’un rectangle de 12 points (cf types) , d’état p et d’angle a, d’une surface avec les couleurs fournit, renvoie un tableau liant chaque couleur trouvée avec son point.

Cette fonction est celle des collisions, en effet elle permet de tester sur quelle ‘sol’ est notre voiture : on teste la couleur des points sur l’image du circuit.

La fonction initialise d’abord

## function isSameColor(a: TSDL\_Color; b: TSDL\_Color): Boolean;

## function ZoomMin(a,b: Real): Double;

## procedure imageLoad(chemin: String; var surface: PSDL\_Surface; alpha: Boolean);

## procedure updatePseudo(k : TSDLKey; var pseudo: String);

## procedure scoreMaj(fichier: String; miseAJour: T\_SCORES);

## function min(liste : array of LongInt): LongInt;

## function test(): boolean;

## procedure freeUiElement(var element: P\_UI\_ELEMENT);

## procedure freeInfoPartie(var infoPartie: T\_GAMEPLAY);

## procedure getBestScore(var scores: T\_SCORES);

## procedure scoreLire(nomFichier: String; var scores: T\_SCORES);