

L'objectif de cet atelier est de vous familiariser avec quelques notions de mathématiques et de physiques qui vous seront utiles afin d'assurer un déplacement réaliste de vos entités.  
En effet le calcul du pas du déplacement d'une entité doit être fait en fonction de sa vitesse.

## 1) Rappel :

Nous présentons, dans ce qui suit, un rappel de quelques notions sur la cinématique qui vous seront indispensables pour faire l'animation de vos objets. En effet, la cinématique est l'étude du **mouvement** de la particule ou des systèmes de particules en introduisant le temps et sans rechercher les causes de ce mouvement car ils font partie de l'étude de la dynamique des objets. Nous traitons le cas des objets en mouvement unidimensionnel dont la trajectoire est une droite.

a- La vitesse:

$$v = \frac{\Delta x}{\Delta t}$$

A vitesse constante, la distance  $\Delta x$ , parcourue entre deux instants quelconques  $t_1$  et  $t_2$  est proportionnelle à la durée, d'où :

$$\Delta x = v \cdot \Delta t$$

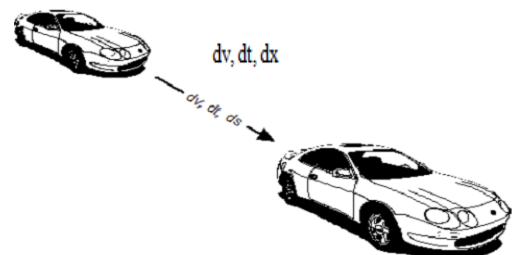
, avec:

$$\begin{aligned}\Delta x &= x(t + \Delta t) - x(t) \\ \Delta t &= t_2 - t_1\end{aligned}$$

$v$  : la vitesse du mobile.

$x(t)$ : l'abscisse du mobile à l'instant  $t$ .

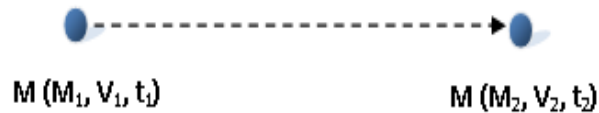
$x(t + \Delta t)$  : l'abscisse du mobile à l'instant  $t + dt$ .



## b- L'accélération:

L'accélération est le taux de variation instantanée de la vitesse. Une accélération positive indique une vitesse **croissante**, une accélération nulle indique une vitesse **constante** et une accélération négative indique une vitesse **décroissante**.

L'accélération est la dérivée de la vitesse, et donc la dérivée seconde de la distance. Elle s'exprime en **m/s<sup>2</sup>**.



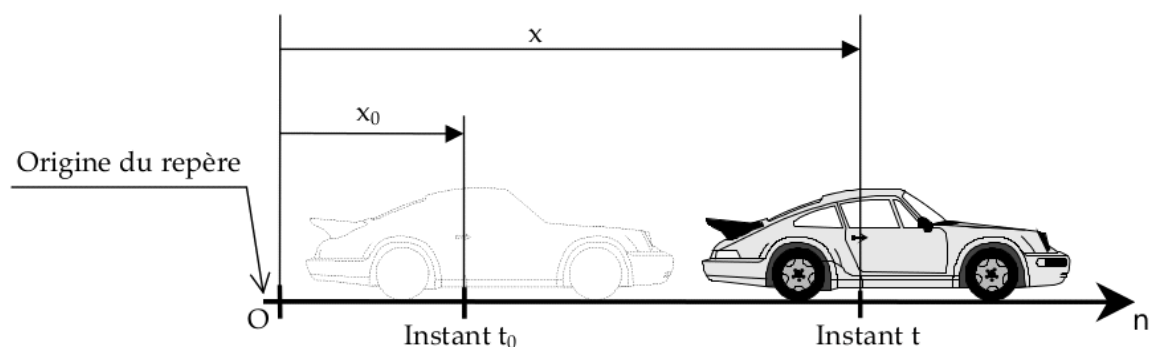
L'accélération **moyenne** de l'objet **M** entre les instants **t<sub>1</sub>** et **t<sub>2</sub>** de la position **M<sub>1</sub>** à la position **M<sub>2</sub>** en variant sa vitesse de **V<sub>1</sub>** à **V<sub>2</sub>**, est :

$$a_{\text{moy}} = \frac{v_2 - v_1}{t_2 - t_1} = \frac{\Delta v}{\Delta t}$$

A un instant **t** quelconque, l'accélération **instantanée** correspond à la limite du rapport  $\frac{\Delta v}{\Delta t}$  lorsque  $\Delta t$  tends vers 0:  $a(t) = \lim_{\Delta t \rightarrow 0} \frac{\Delta v}{\Delta t} = \frac{dv}{dt}$ , or  $v(t) = \frac{dx}{dt}$ , d'où :  $a(t) = \frac{d^2 x(t)}{dt^2}$

## 2) Etude de cas:

### a- Mouvement rectiligne uniforme:



Dans ce cas de figure la voiture avance sans que le conducteur accélère ( $a=0$ ) d'où la vitesse de mouvement est constante par rapport au temps. (Tout en supposant que les forces de frottements sont nulles)

Notons:  $t_0$ : instant initial:  $t_0=0$  s

$x_0$ : le déplacement initial, à  $t=t_0$

$v_0$ : la vitesse initiale, à  $t=t_0$

$x(t)$ : le déplacement à l'instant  $t$

$v(t)$ : la vitesse à l'instant  $t$

Pour une entité en mouvement **rectiligne uniforme**:

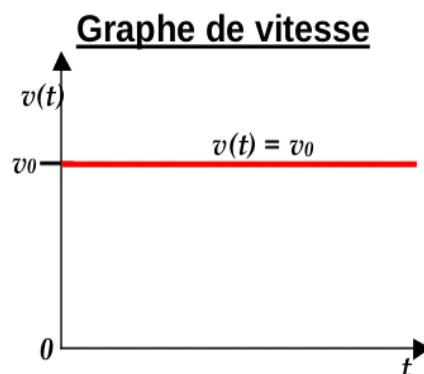
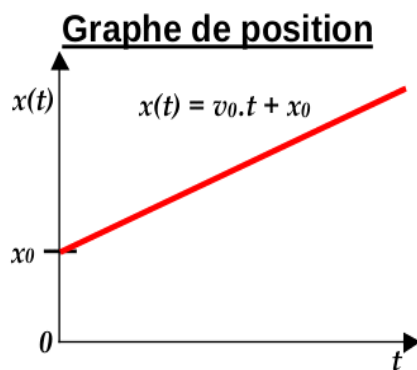
L'équation relative à l'accélération instantanée est:  $a(t)=0$

L'équation relative à la vitesse instantanée est:  $v(t)=v_0=constante$

L'équation relative à la position instantanée est:  $x(t)=\int v(t) \cdot dt$  d'où  $x(t)=v_0 \cdot t + x_0$

L'équation relative à un petit déplacement  $dx$ , est:

$$dx = v_0 \cdot dt \quad (1)$$



### ➤ Application:

1. Récupérer le code source du TP (dossier nommé TPMathPhysics/Part1/).
2. Lancer un terminal, accéder au dossier en question, ensuite, compiler le code source.
3. Exécuter le binaire résultant.
4. Dans la boucle du jeu, ajouter le code permettant de calculer la valeur de **dt** à chaque itération.

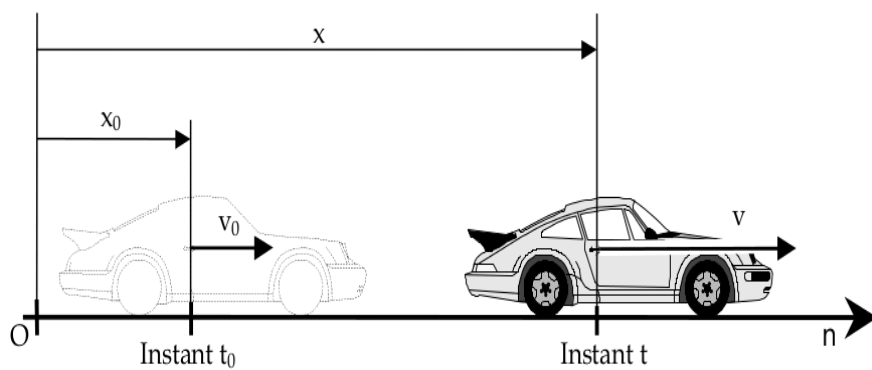
```
uint32 t_prev, dt=1; //dans la partie déclaration
while(!done) {
    t_prev = SDL_GetTicks(); //au début de la game loop
    ...
    dt=SDL_GetTicks()-t_prev; //à la fin de la game loop
    if(1000/FPS > dt)
        SDL_Delay(1000/FPS - dt); //pour avoir un FPS constant
}
```

5. A chaque nouvelle itération de la boucle du jeu, la position  $x_n = x_{n-1} + dx$ , implémenter, donc, le déplacement de la voiture moyennant la relation entre  $dx$  et  $dt$  (voir (1)).
6. Recompiler le code source et ré-exécuter le nouveau binaire résultant.

## b- Mouvement rectiligne uniformément accéléré/décéléré:

Dans ce cas de figure l'accélération reste constante au cours du temps. On peut distinguer deux mouvements:

- Mouvement accéléré ( $a > 0$ )
- Mouvement décéléré ( $a < 0$ )



Notons:

- $t_0$ : instant initial:  $t_0 = 0s$
- $x_0$ : le déplacement initial, à  $t = t_0$
- $a_0$ : l'accélération initiale
- $v_0$ : la vitesse initiale à  $t = t_0$
- $x(t)$ : le déplacement à l'instant  $t$
- $v(t)$ : la vitesse à l'instant  $t$

Pour une entité en mouvement **rectiligne uniformément accéléré**:

L'équation relative à l'accélération instantanée est:  $a(t) = a_0 = \text{constante}$

L'équation relative à la vitesse instantanée est:  $v(t) = a_0 \cdot t + v_0$  et  $x(t) = \frac{1}{2} a_0 \cdot t^2 + v_0 \cdot t + x_0$

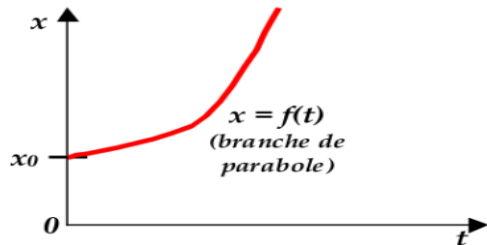
L'équation relative à la position instantanée est:

$$x(t) = \int (v(t) \cdot dt) = \int (a_0 \cdot t + v_0) \cdot dt = \frac{1}{2} a_0 t^2 + v_0 \cdot t + x_0$$

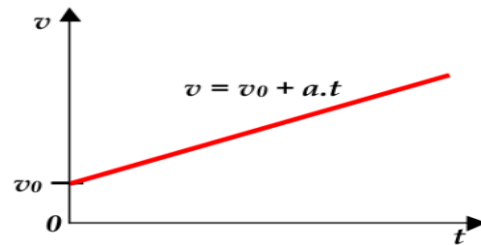
L'équation relative à un petit déplacement  $dx$ , est:

$$dx = \frac{1}{2} a_0 \cdot dt^2 + v_0 \cdot dt \quad (2)$$

Graphe de position



Graphe de vitesse



➤ Application:

1. Lancer l'exécution du binaire « prog » (déjà existant dans le dossier Part2).
2. Tester l'exécution du binaire en question:
  - Accélérer l'avancement de la voiture en appuyant sur la touche « **espace** » du clavier.
  - Laisser la voiture avancer toute seule sans l'appui sur la touche d'accélération, la voiture s'arrêtera toute seule sous l'effet de la décélération causée par un certain facteur de frottement.
  - Décélérer l'avancement de la voiture en appuyant sur la touche « **retour chariot** ».
3. Copier le contenu du dossier Part1 dans le dossier Part2
4. Au niveau de la boucle du jeu :
  - Programmer l'événement de l'accélération (lors de l'appui sur la touche « **espace** ») avec un cumul d'une constante **car.acceleration+=0.005** lors de l'appui sur « **espace** ».
  - Programmer l'événement de décélération (lors de l'appui sur la touche « **retour chariot** ») avec un cumul d'une constante **car.acceleration-=0.01** lors de l'appui sur « **espace** ».
  - Ajouter une décélération avec un cumul de la constante **car.acceleration-=0.001** (A chaque itération et sans événement préalable).

5. Tout en sachant qu'à chaque nouvelle itération du boucle du jeu:  $x_n = x_{n-1} + dx$ , implémenter le déplacement de la voiture (fonction « **moveVoiture** ») moyennant la valeur de l'accélération et la variation de la vitesse (voir **(2)**).
6. Compiler à nouveau, ensuite exécuter le binaire résultant.