| **Module : Projet 1A** | **Atelier Accélération** |  |
| --- | --- | --- |

**Introduction**

L’objectif de cet atelier est de vous familiariser avec quelques notions de mathématiques et de physique qui vous seront utiles afin d’assurer un déplacement réaliste de vos entités.

En effet, le calcul du pas de déplacement d’une entité doit être fait en fonction de sa vitesse.

Nous présentons, dans ce qui suit, un rappel de quelques notions sur la cinématique qui vous seront indispensables pour le déplacement de vos objets.

**Méthodologie**

La cinématique est l’étude du mouvementde 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 du personnage principal dans notre jeu.

Le personnage est en mouvement unidimensionnel selon une trajectoire rectiligne (droite).

L’accélération est le taux de variation instantanée de la vitesse.

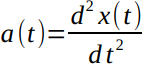
Une accélération positive indique une vitesse croissante, une accélération nulle indique une vitesse constanteet une accélération négative indique une vitesse décroissante.

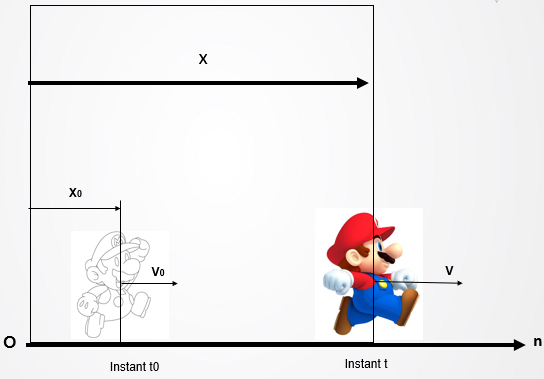
L’accélération est la dérivée de la vitesse, et donc la dérivée seconde de la distance exprimée en m/s2.



L’accélération moyenned’un objet Mentre les instants t1et t2de la position M1à la position M2en variant sa vitesse de V1à V2 est :



A un instant tquelconque, l'accélération instantanéecorrespond à la limite du rapport  lorsque **∆**ttends vers 0: , or, d'où: 



**Figure 1 : Accélération du Mouvement**

Dans ce cas de figure (*F*igure 1), l’accélération reste constante au cours du temps.

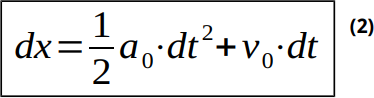
On peut distinguer trois mouvements :

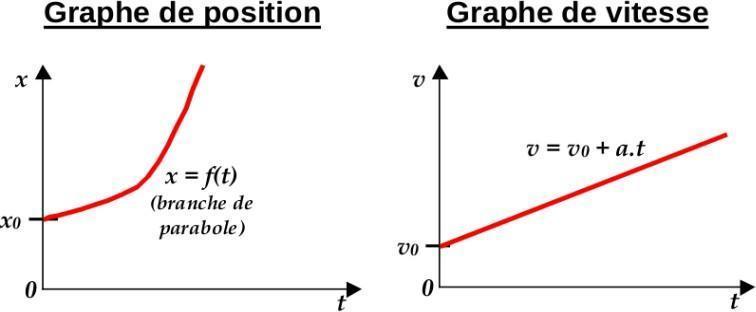
* Mouvement accéléré (accélération>0)
* Mouvement décéléré (accélération<0)
* Mouvement à pas constant (acceleration=0)

Notons :

* t0: instant initial t0=0s
* x0: l'abscisse initial à t=t0
* a0: l'accélération initiale
* v0: la vitesse initiale à t=t0
* x(t): l'abscisse à l'instant t
* v(t): la vitesse à l'instant t

Pour une entité en mouvement rectiligne uniformément accéléré, l’équation relative à un petit déplacement dx est :





**Figure 2 : Graphes de Position et de Vitesse (cas vitesse croissante)**

**Application**

**Objectifs :**

* Accélérer le déplacement du personnage en appuyant sur une touchedu clavier
* Si vous lâchez votre doigt de la touche, le personnage avance avec une vitesse décroissante jusqu'à s’arrêter sous l'effet de la décélérationcausée par un certain facteur de frottement
* Décélérer l'avancement du personnage en appuyant sur une touche du clavier

**Description :**

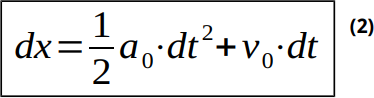
1. Ajouter les champs vitesse et accélération dans la structure personnage :

**double vitesse, acceleration;**

1. Initialiser la vitesse du personnage à une valeur max et son accélération à 0 dans la fonction **void initialiserPerso(perso\* p)**:

**p->vitesse=5;**

**p->acceleration=0;**

1. Implémenter la fonction de déplacement du personnage **void movePerso(Perso \*p,Uint32 dt)** moyennant la relation entre dx et dt en utilisant l’équation de calcul du pas d’avancement suivante: 
2. Déclarer les variables t\_prev et dt de type Uint32 dans la fonction **main()**:

**Uint32 dt, t\_prev;**

1. Au niveau de la boucle de jeu:

* Calculer la duréede chaque itération en utilisant la fonction **SDL\_GetTicks()** qui retourne

le temps écoulé depuis l’initialisation de SDL:

**t\_prev=SDL\_GetTicks(); //au début de la boucle de jeu**

**dt=SDL\_GetTicks()-t\_prev; //à la fin de la boucle de jeu**

* Programmer l’événement de l’accélération:

Si appui sur la touche accélération alors exécution d’un cumul de la constante **p.acceleration+=0.005;**

* Programmer l'événement de décélération:

Si appui sur la touche décélération alors exécution d’un cumul de la constante **p.acceleration-=0.01;**

* Ajouter une décélération avec un cumul de la constante:

**p.acceleration-=0.001; //à chaque itération et sans événement préalable**

1. Gérer les différents inputs (appui touche clavier), mettre à jour l'accélération et appeler la fonction **void movePerso(Perso \*p,Uint32 dt)**