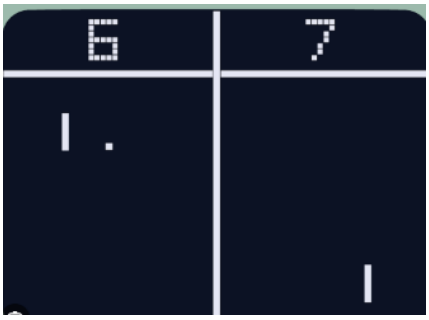
ANNEXE 1 : PROGRAMMATION EVENEMENTIELLE



jeu Pong (1972)

jeu Tetris (1984)

I. Rappel sur la programmation séquentielle

Les programmes que vous avez étudiés ou réalisés jusqu'ici sont dits « séquentiels ». L'ordre dans lequel sont exécutées les instructions (flux de contrôle) est imposé par le programme. Bien sûr, l'utilisateur peut interagir avec le programme via par exemple des demandes de saisies clavier. Mais le moment où l'utilisateur doit effectuer ces saisies est fixé dans le programme. L'utilisateur ne peut pas intervenir n'importe quand pour changer son déroulement. Vous avez d'ailleurs remarqué que la fonction *scanf* est « bloquante » : le programme s'arrête et attend que l'utilisateur saisisse une valeur.

II. La programmation évènementielle

En programmation événementielle, le principe est différent :

- Le programme **tourne en continu dans une boucle principale** et continue à s'exécuter même s'il n'y a pas d'action de l'utilisateur.
- **L'utilisateur peut intervenir à tout moment** : le programme « surveille » si des **évènements** se produisent. Un événement correspond à un signal venant de l'extérieur du programme, par exemple l'appui sur une touche du clavier, un clic de souris …
- Lorsqu'un événement survient, **le programme réagit** en appelant une action prévue pour cet événement.

En mode console en C, on gère surtout les évènements clavier (la gestion des évènements souris est compliquée en mode console). On utilise pour cela principalement deux fonctions de la librairie ***conio : kbhit*** et ***getch.***

III. La fonction *kbhit*

- Prototype : ***int kbhit();***
- Valeur de retour :
  - 0 si aucune touche a été pressée
  - Une valeur différente de 0 si une touche a été pressée (la valeur retournée dépend du système)
- Comportement : non bloquante
- Test pour savoir si une touche a été pressée :

```
//si une touche a été pressée
if(kbhit()!=0){
    //ce qu'il faut faire
}
```

IV. La fonction *getch*

- Prototype : ***int getch();***
- Valeur de retour :
  - Un entier correspondant au code de la touche pressée.
- Rôle et comportement :
  - Lit la valeur d'une touche au clavier sans attendre une validation (L'utilisateur n'a pas besoin d'appuyer sur la touche Entrée pour valider).
  - La valeur de la touche n'est pas affichée à l'écran.
  - Bloquante : Le programme attend qu'une touche soit disponible pour poursuivre son exécution
  - ⚠ Pour certaines touches « spéciales » (flèches, FX…), *getch* retourne d'abord 0 ou 224. Il faut rappeler *getch* pour obtenir le code de la touche.

- Utilisation en programmation évènementielle :

  Pour ne pas bloquer le programme, il faut utiliser *getch* avec *kbhit* :

```
unsigned char touche;
//si une touche a été pressée
if(kbhit()){
    //on récupère la valeur de la touche
    touche=getch();
    //on peut orienter le programme différemment selon la valeur de la touche pressée
    if(touche=='z'){
        //ce qu'il faut faire si l'utilisateur a appuyé sur 'z'
    }
    else{
        if(touche=='w'){
            //ce qu'il faut faire si l'utilisateur a appuyé sur 'w'
        }
    }
}
```