

Mini projet Arduino : Donkey Kong



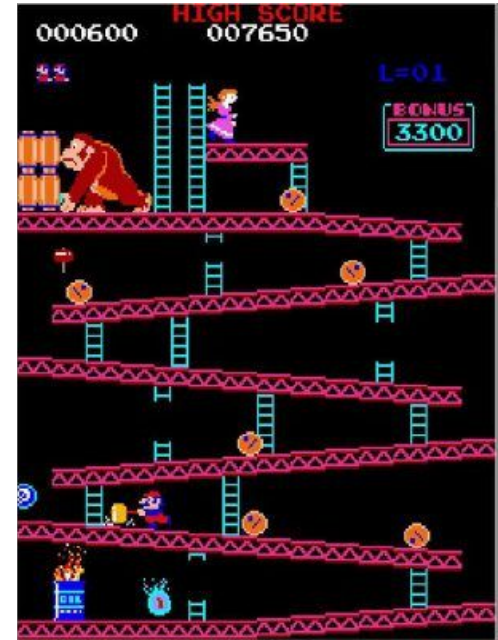
LECONTE Eugène



B1

Présentation du projet

- Inspiré du jeu DONKEY KONG des années 80'
- On dirige Luigi pour arriver en haut d'une tour
- Attention aux tonneaux!

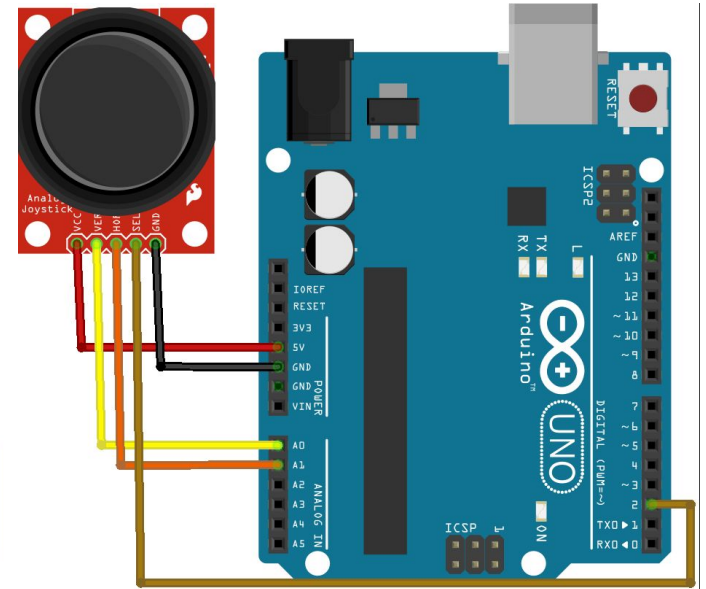
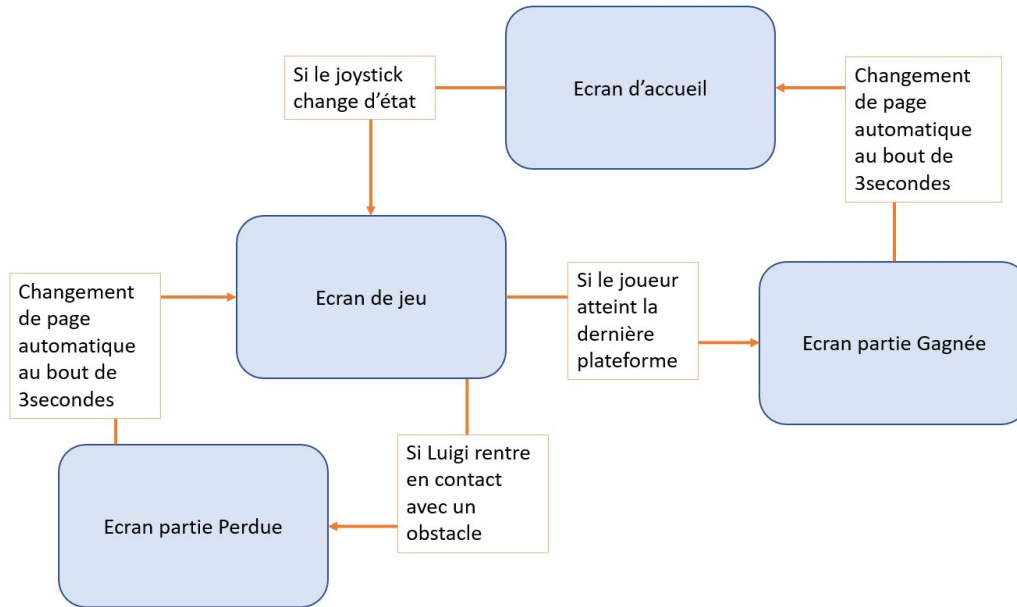


Sommaire :

- Cahier des charges
- Présentation des éléments clefs :
 - Gestion du décors
 - Gestions des contacts entre le personnage et les échelles et le sols
 - Gestions des obstacles aléatoires
- Points difficiles
- Perspectives et conclusion
- Annexes

Cahier des charges

Graphe d'état :



Matériel :

- Une carte arduino
- Une matrice de diodes 32x16
- 1 joystick

Gestion des éléments clefs

Gestion des décors

- Utilisation de tableau pour répertorier les sols et échelles
- Changement de coordonnées entre le jeu et la matrice
- Effacement des anciens décors et affichages des nouveaux

```
// Position des Echelles sur l'ensemble du niveau 1
T_echelle tableauEchelles[NB_ECHELLES] = {
    {10, 1, AFFICHE},
    {18, 7, AFFICHE},
    {17, 13, AFFICHE},
    {27, 13, AFFICHE},
    {9, 19, AFFICHE},
    {2, 19, AFFICHE},
    {6, 25, AFFICHE},
    {2, 31, AFFICHE},
    {29, 31, AFFICHE},
    {9, 37, AFFICHE},
    {30, 37, AFFICHE},
    {5, 43, AFFICHE},
};
```

Utilisation de tableau pour répertorier les sols et échelles

tableauEchelle de type T_Echelle

| Coordonnée en X | Coordonnée en Y |
|-----------------|-----------------|
| 10 | 1 |

tableauSol de type T_Sol

| | |
|----|--|
| 15 | |
| 14 | |
| 13 | |
| 12 | |
| 11 | |
| 10 | |

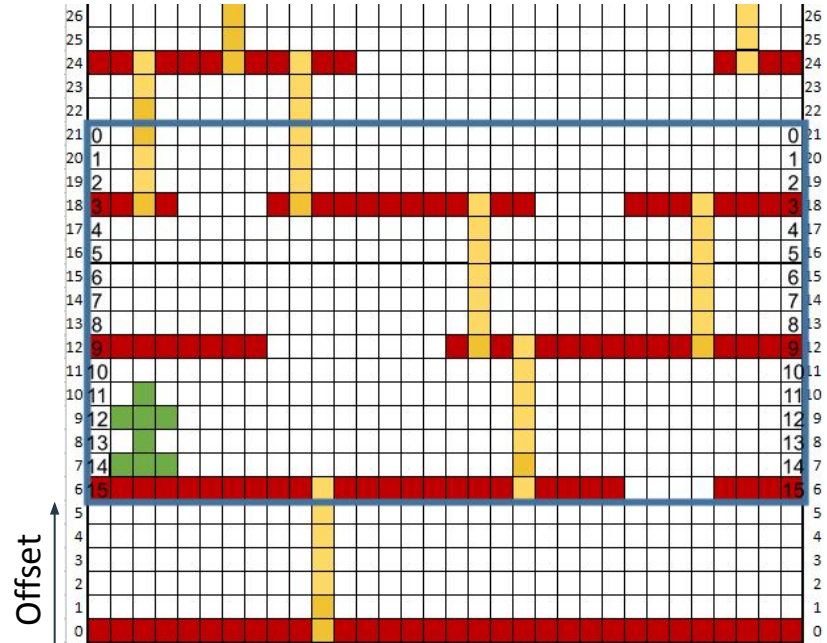
| Y | Pos1 | Pos2 | Pos3 | Pos4 | Pos5 | Pos6 | Pos7 | Pos8 |
|----|---------|---------|-------|---------|-------|---------|---------|---------|
| 12 | affiche | affiche | cache | affiche | cache | affiche | affiche | affiche |



Changement de coordonnées entre le jeu et la matrice

```
char getY(byte Y) {  
    char Ymatrice;  
    byte Ycadre;  
    Ycadre = Y - offset; //Per  
    Ymatrice = (15 - Ycadre);  
    return Ymatrice;  
}
```

Pour l'échelle avec en Yjeu = 6
Ycadre = 6 - Offset (ici 6)
Ymatrice = (15 - 0)



Gestion des obstacles aléatoire

Fonction random pseudo aléatoire donc utilisation de `"randomSeed(analogRead(0));"`

Gestion de 4 obstacles max sur la matrice

Deux fonctions de gestions :

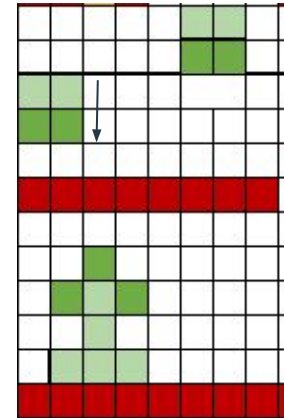
- la première permettant de faire chuter
- La seconde pour remplacer un obstacle si sa valeur en Y dépasse 16

Pour gérer les collisions entre les tonneaux et le joueur, on test les coordonnées des tonneaux avec ceux des deux bras et de la tête du joueur

Les points forcés représentent les points de test de contact

T_Tonneau

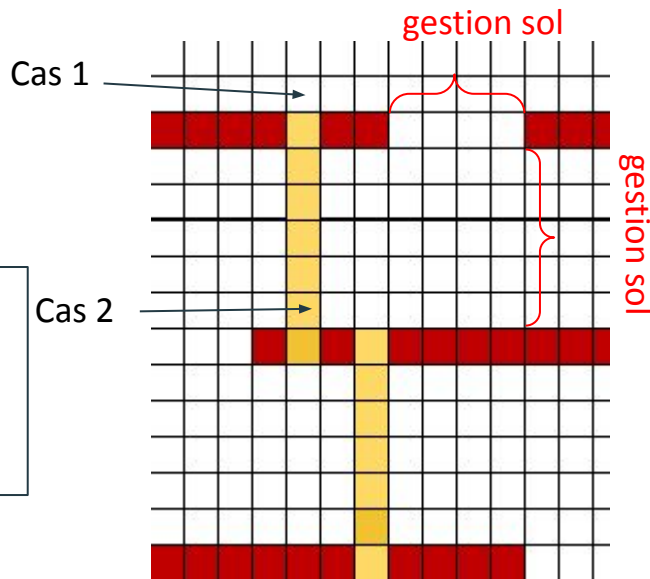
| posX | PosY |
|------------------------|------|
| aléatoire de 0 à 30 | 0 |



Gestion des contacts entre le personnage et les échelles et le sols

Pour une échelle :
Comparaison des coordonnées
échelles/personnage

Pour un sol :
Test présence sol
Comparaison coordonnées
sol/personnage



Dans chaque cas, on ne modifie pas la position Y du personnage mais la valeur de l'offset faisant défiler vers le haut ou le bas le décors

Points difficiles

- Établir la gravité du personnage
- Gérer les décors dans des tableaux
- Le contact entre le personnage et les autres éléments
- Gestion des différents écrans

Perspective et conclusion

- Amélioration de l'affichage
- Ajout de différent niveau
- Difficulté ajustable
- Niveaux plus ambitieux



Amélioration

- Axer la première partie du module sur le Mini-Projet;
- Meilleure gestion de l'emploi du temps.

Annexes Jeu

Écran d'accueil



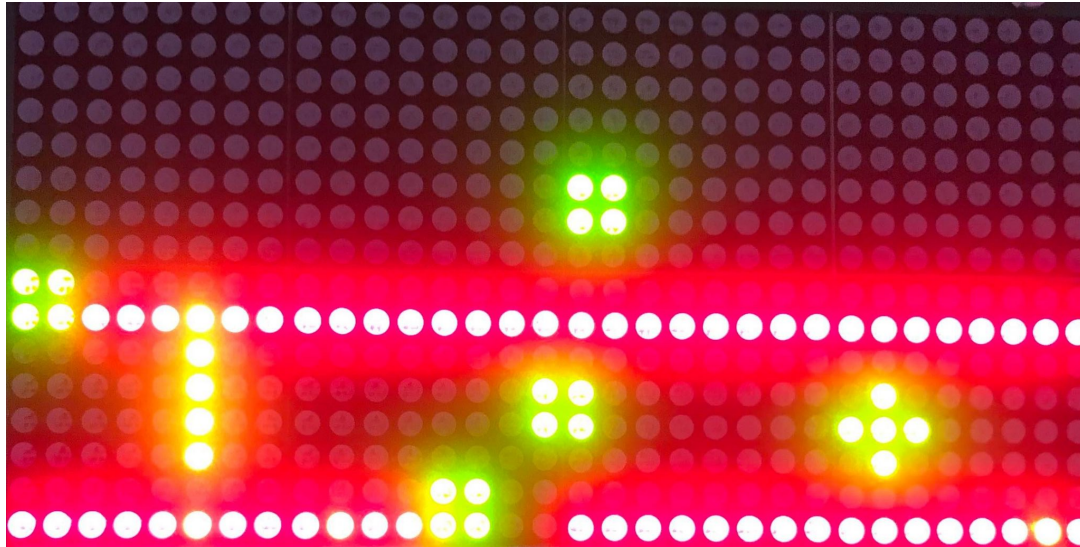
Partie en cours



Partie Perdue



Fin du niveau



Partie gagnée

