

# **DOSSIER DE CONCEPTION TUNNELS AND TROLLS**

## **SCENARIO :**

Tunnels & Trolls va mettre le joueur dans la peau d'un personnage cherchant à faire fortune. Pour cela, il faudra piller une grotte infestée de monstre et aussi de trésors. Le plus gros trésor étant gardé par un puissant dragon.

## **PERSONNAGES :**

Le joueur pourra choisir d'incarner 1 personnage parmi 4 types :

- L'archer
- Le magicien
- Le voleur
- Le guerrier

## **SYSTEME DE JEU :**

Compétence des personnages :

Chaque personnage possède une et une seule vie. n points de santé maximum, un certain nombre de « point de combat ». (munition, durée de vie de l'arme), un niveau de dégât et une vitesse.

- Au départ, l'archer possède un carquois avec 20 flèches. 2 Points de dégâts par coup. Vitesse = 2. 100 points de vie max.
- Le magicien a mémorisé 20 sorts d'attaques qu'il peut lancer. 3 Points de dégâts par coup. 70 points de vie Max. Vitesse=2.
- Le voleur possède 20 dagues qu'ils pourra lancer. 1 Point de dégât par coup. Vitesse = 3. 100 points de vie max.
- Le guerrier possède une épée qui pourra utiliser 40 fois avant qu'elle ne se casse (corps à corps). 120 points max. 3 Point de dégât par coup, Vitesse = 1 (lent)

## **Le labyrinthe :**

Le héros devra trouver la salle du dragon dans un labyrinthe composé de 16\*16 salles (256 salles).  
Les salles sont de 3 types :

- Labyrinthe (12 différentes)



- Salle de garde / Salle de trésor / Cachot



Cette salle est obligatoirement gardée par un ou plusieurs troll et contient de l'or ou des bonus.

- Salle du dragon (1 seule et unique) : Dans cette salle contrairement aux autres salles, les bonus d'armes « respawn » régulièrement. Dans cette salle il y a forcément un dragon.

## **Le bestiaire qu'on peut tuer :**

Les monstres attaquent au corps à corps, et font perdre n point de vie par tour ou ils sont en contact avec le joueur.

- Trolls : Ces êtres répugnant se trouvent surtout dans des salles de garde. Il possèdent 3 points de vies, et font 2 points de dégâts par tour.
- Serpents : Ils sont lent, par groupe de 8, démarrent du centre de la pièce et vont chacun dans une direction et « rebondissent » sur les murs dès qu'ils en rencontre un. (1 point de vie, 1 point de dégât)
- Dragon : Boss final du jeu. 100 points de vie. Lance des boules de feu en direction du joueur qui font 20 points de dégâts. Si le joueur a trouvé la potion anti-flamme les dégâts seront limité à 5 points, à 0 points pour le chevalier uniquement.
- Dynamite : Objet à utiliser par le joueur. Si celui ci est trop prêt d'elle quand elle explose : 50 points de dégâts. (C'est pourquoi elle est considérée comme bestiaire).
- Chauves-souris

## **Le bestiaire qui respawn (8 maxi) :**

- Squelettes : Contrairement aux autres monstres, une fois éliminés ceux ci se transforme en tas d'os. Si le joueur revient dans la salle/labyrinthe, ils seront de

nouveaux recomposés et prêt à attaquer le joueur. 2 point de vie, 1 point de dégât. Ils sont toujours par groupe de 2.

- Araignée rouge : Les araignées rouges ne bougent pas et n'attaque pas le joueur. Elles crachent des toiles d'araignées qui paralyse le joueur durant un certain laps de temps. (2 points de vie)
- Araignée verte : Les araignées vertes ne bougent pas mais attaquent le joueurs avec un salve de poison qui fait 10 points de dégâts. (2 Points de vie)

### **Les bonus :**

- Recharge d'arme : Remet +5 aux armes des joueurs.
- Clé : Rouge
- Clé : Verte
- Anneau de force : Objet unique, double les dégâts
- Coffre : Augmente le nombre de pièces d'or trouvé
- Potion anti flamme : Objet unique pour le combat final
- Potion de soin partielle : + 20 points de vie.
- Dynamite

### **Les obstacles :**

- Portes : Rouge, Vert → Nécessite une clé de couleur identique pour l'ouvrir. Une fois ouverte, la clé disparaît de l'inventaire.
- Porte Noire : Une clé, quelle quelle soit fait l'affaire.
- Rocher : Bloque une sortie. Nécessite de la dynamite pour passer.
- Trappes pièges : Invisible au joueur, elles s'activent quand le joueur passe dessus.
  - → Lancé de flèches (évitable)
  - → Trappe ouverte sur sol de pointe (1 points de dégâts par tour ou le joueur reste dessus)

### **Combat Final :**

Le combat final à lieu dans l'ancre du Dragon. Celui-ci se déplace latéralement et tire des boules de feux sur le héros.

En dessous du dragon 4 ou 5 rochers derrières lesquelles ont pourra se cacher pour éviter les tirs. Et régulièrement les armes seront respawneés. Le combat devrait être plus accessible pour les personnages qui peuvent tirer à distance.

**La potion anti flamme devra être obligatoire pour le chevalier pour ne pas mourir.**

## **CONCEPTION TECHNIQUE :**

Ce chapitre propose un début de réflexion sur la conception technique.

Le labyrinthe sera codé en ROM. Les portions de labyrinthe seront des «salles» précalculés (soit stocké en rom).

Les salles sont numérotés ainsi :

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,  
17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,  
32,33,34,...

La structure globale du labyrinthe sera ainsi en mémoire :

Byte laby[] = {Type de salle pour 0 et 1, Type de salle pour 2 et 3, ...}

2 salles sont codées sur 1 seul octet. Taille totale = 128 octets

Byte contenance[] = {contenance salle 1, contenance salle 2, ...}

Les 4 premiers octets sont le bestiaires de la salle, les 4 suivants les objets de la salle. Selon le type de salle chaque byte représente un autre monstre/objet. Taille totale = 256 octets.

C'est le programme qui déterminera, selon la salle à afficher , ou mettre les bonus/monstres/trésors, grace à un tableau. (Voir détail ci dessous)

## **HEROS, MONSTRES ET OBJETS :**

Les héros et monstres seront toujours vu de coté (direction droite ou gauche), et occuperont 2 sprites monochromes (16\*32 maxi). (Voir les mouvement de Dungeon and Dragon Intellivision)

Certains objets seront dessinés en caractères (portes, rochers, trappes), d'autres en sprites de 16\*16 (potions, bonus d'armes etc etc ...).

## **ZONE DE JEU, SALLES et TILES:**

La zone de jeu, dans laquelle sera affichée le décor est de 28\*20 caractères. Les 4 autres caractères restant en Y seront pour la zone de sécurité, afficher la vie, l'or, et les objets dans l'inventaire.

Pour donner une certaines variété au labyrinthe il va falloir stocker en mémoire un certains nombres de salle labyrinthe. Une salle sera composée de tiles. Chaque tiles faisant 32\*32 pixels (4\*4caractères) , une plaque sera donc composée de 7\*5 tiles.

1 salle « classique » occupera donc 35 octets en mémoire. Le labyrinthe total devra être composé de 12 salle = 12\*35 octets = 420 octets.

Dans cette salle, il faut pouvoir positionner les monstres, les objets etc etc ...

Considérons la salle composée de tile ainsi :

t0,t1,t2,t3,t4,t5,t6,t7  
t8,t9,[...]

On met en place une table des objets autorisés sur cette plaque et de leur position.

Objet sur salle labyrinthe :  
Araignée Rouge, t13  
Serpent,t10  
Potion,t16

Sur cette plaque seul les araignées et les serpents sont autorisé et une potion est affichée respectivement aux position 10,13,16.

Donc pour cette plaque, en mémoire, coûtera  $4 + 6 \text{ octets} = 10 \text{ octets}$ .

(A AFFINER)

### **PSEUDO ALGO :**

#### **Variables :**

```
const byte gfxTile1[] = {c1,c2,c3,c4,...} // 16 octets pour une tile
};

const byte gfxTile2[] = {c1,c2,c3,c4,...} // 16 octets pour une tile
};

const byte const byte *tiles[] = { gfxTile1, gfxTile2 };

const byte salleNormale1[] = {0,0,0,0,1,1,...}; Contient les ID des Tiles qui constitue la salle

const byte const byte *salles[] = {salleNormale1,salleNormale2,...};

const char autoriseSalle0[] = {TROLL,11,-1
}

const byte autoriseGlobal[] = {*autoriseSalle0,*autoriseSalle1,...};

const byte LabyGlobal[] = {
    0,0,1,1, // Id des salles
};

const byte contenance[] = {
    // Ce qu'il y a dans les salles du laby global
}

void copieLaby2Ram()
{
    disable_nmi();
    screen_off();
    // Copie les informations de contenance en RAM - 256 Octets
    screen_on();
    enable_nmi();
}

void GénèreSalle(int zoneLabyNo)
{ [...] }
```