

# PROJET 1: PROGRAMMATION DU JEU DE *LODE RUNNER*

---

## 1. Le but du projet

Le but de ce projet est de programmer le 1<sup>er</sup> tableau du jeu de *Lode Runner*.

**Ce travail doit OBLIGATOIREMENT être réalisé en équipe de deux.**

## 2. Le code Javascript

Étant donné que ce travail est en équipe, je vous suggère fortement de placer votre code **Javascript** dans plusieurs fichiers **.js** afin de bien répartir la tâche entre chaque étudiant (mais ce n'est pas obligatoire).

## 3. La remise du projet

- La date de remise est **Lundi, le 14 mars 2024 à minuit**.
- Dans le titre de la page Web, vous devez écrire *Lode Runner* suivi de vos noms.
- Remettre **votre jeu complet zippé** sur LÉA (avec les images s'il y en a ainsi que les sons et la musique).
- Ce projet comptera pour 13% de la session.
- Tout projet remis en retard sera pénalisé de 10% par jour de retard. Maximum 5 jours.
- **Si je m'aperçois que vous avez utilisé un jeu de *Lode Runner* déjà programmé soit à partir d'Internet, soit à partir d'une autre équipe, vous obtiendrez automatiquement la note de 0%.**

## 4. L'origine du jeu<sup>1</sup>

En 1982, **Dough Smith** développa le premier prototype du jeu de *Lode Runner* pour l'**Apple II+**. À cette époque, le jeu portait le nom de *Miner*. À la fin des années 1982, il soumit une première version de son jeu à l'éditeur **Brøderbund** dans le but de le commercialiser. Ce fut un refus. Par la suite, il raffina son jeu et l'offrit à 4 éditeurs différents. Ces 4 éditeurs acceptèrent son offre mais **Dough Smith** signa avec l'éditeur **Brøderbund**.

Au milieu de l'année 1983, ce jeu fut renommé *Lode Runner* et fut commercialisé sur les micro-ordinateurs de l'époque : **Apple II**, **Atari**, **VIC-20**, **Commodore 64** et **IBM PC**. À l'époque, le jeu avait 150 niveaux (tableaux) différents mais l'originalité de ce jeu était que l'utilisateur pouvait créer ses niveaux personnalisés grâce à un éditeur de niveau. Ce fut d'ailleurs l'un des premiers jeux à offrir un éditeur de niveau.



LODE RUNNER SUR APPLE II

## 5. Le but du jeu (du 1<sup>er</sup> tableau)<sup>2</sup>

Le but du 1<sup>er</sup> tableau est de ramasser les six lingots d'or situés sur des passerelles tout en évitant les gardes qui essaient d'attraper le joueur (*Lode Runner*). Lorsque les six lingots ont été ramassés, une échelle apparaît. Cette échelle donne la possibilité au joueur de « quitter le décor » et ainsi passer au prochain tableau (niveau).

*Lode Runner* a la possibilité de se déplacer horizontalement sur des passerelles et sur des barres de franchissement. Il peut également se déplacer verticalement sur des échelles. Finalement, il a la possibilité de creuser des trous temporaires dans la passerelle. *Lode Runner* utilise ces trous pour descendre plus rapidement et/ou pour piéger les gardes pendant un certain laps de temps.

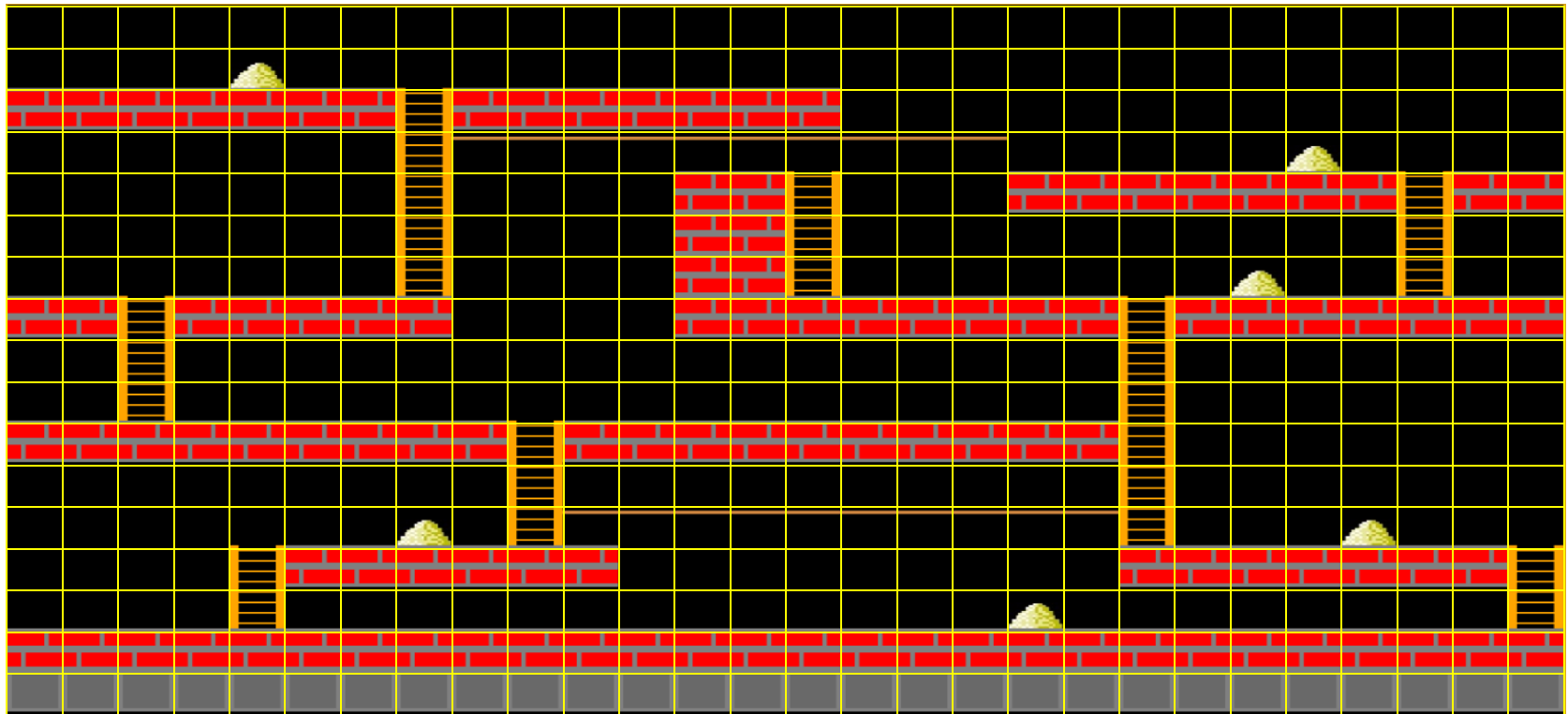
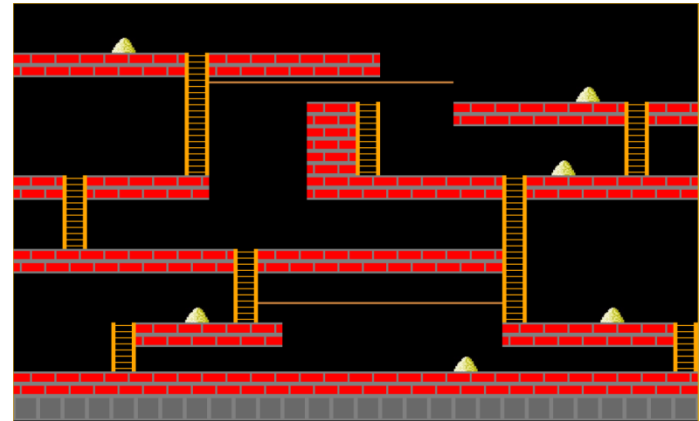
<sup>1</sup> Référence : [https://en.wikipedia.org/wiki/Lode\\_Runner](https://en.wikipedia.org/wiki/Lode_Runner)

<sup>2</sup> Référence : [https://fr.wikipedia.org/wiki/Lode\\_Runner](https://fr.wikipedia.org/wiki/Lode_Runner)

## 6. Les objets fixes du 1<sup>er</sup> tableau

Voici les objets fixes que vous aurez à utiliser et à dessiner dans le 1<sup>er</sup> tableau du jeu de *Lode Runner* : les passerelles (et les murs), les échelles, les barres de franchissement et les lingots d'or.

Je vous suggère fortement de découper le 1<sup>er</sup> tableau en plusieurs cellules et de dessiner ce tableau cellule par cellule. Dans ce tableau, il y a 28X17 cellules. Par exemple, ici, le lingot d'or le plus haut est situé sur la 2<sup>ème</sup> ligne et sur la 5<sup>ème</sup> colonne. Par exemple, ici, la première échelle en bas à gauche est située sur la 5<sup>ème</sup> colonne et sur les 14<sup>ème</sup> et 15<sup>ème</sup> lignes. Il n'est pas nécessaire que vos dessins soient identiques aux miens.



## 7. Les objets en mouvement du 1<sup>er</sup> tableau

Vous aurez également à dessiner le joueur (*Lode Runner*) ainsi que chacun des gardes. Ces objets vont se déplacer tout au long du jeu. Ils n'ont pas de position fixe.

### 8. *Lode Runner*



#### 8.1 Son emplacement initial

Au début du niveau, *Lode Runner* est situé au milieu de la passerelle en brique la plus basse.

#### 8.2 Se déplacer horizontalement

*Lode Runner* se déplace horizontalement sur les passerelles ainsi que sur les barres de franchissement. *Lode Runner* ne peut pas se déplacer horizontalement lorsqu'il frappe une passerelle ou un mur ou lorsqu'il atteint les limites horizontales. Lorsque *Lode Runner* atteint le vide, il chute mais ne meurt pas.

Pour déplacer *Lode Runner* horizontalement, l'utilisateur doit appuyer sur la flèche à gauche ou à droite. Dès que l'utilisateur relâche cette flèche, *Lode Runner* ne se déplace plus.

Voir **Démo 8.2.1 Déplacement horizontal 1 Lode Runner.wmv**.

Voir **Démo 8.2.2 Déplacement horizontal 2 Lode Runner.wmv**.

#### 8.3 Se déplacer verticalement

*Lode Runner* se déplace verticalement sur les échelles. *Lode Runner* ne peut pas se déplacer verticalement lorsqu'il n'y a pas d'échelle ou lorsqu'il atteint les limites verticales.

Pour déplacer *Lode Runner* verticalement, l'utilisateur doit appuyer sur la flèche en haut ou en bas. Dès que l'utilisateur relâche cette flèche, *Lode Runner* ne se déplace plus.

Voir **Démo 8.3 Déplacement vertical Lode Runner.wmv**.

## 8.4 Creuser des trous

*Lode Runner* a la possibilité de creuser des trous sur les passerelles en brique. *Lode Runner* ne peut pas creuser de trous si directement au-dessus de cette portion de passerelle, il y a un autre objet fixe (un lingot d'or, une échelle, une barre de franchissement ou un mur). La passerelle doit être complètement libre de toute entrave. De plus, il n'est pas possible de creuser un trou si cette passerelle est en béton.

Le trou n'est pas permanent. La portion creusée se remplit automatiquement au bout de 8 secondes.

Dans la version originale, les trous sont noirs mais dans ma version, les trous sont grisés (vous avez le choix).

Pour creuser un trou à la droite de *Lode Runner*, l'utilisateur doit appuyer sur **X** et pour creuser un trou à la gauche de *Lode Runner*, l'utilisateur doit appuyer sur **Z**.

Voir **Démo 8.4 Trouer passerelle Lode Runner.wmv**.

## 8.5 Chuter

Lorsque *Lode Runner* se déplace horizontalement à la fin d'une passerelle ou à partir d'une échelle, s'il n'y a aucun objet pour le soutenir, il tombe dans le vide (il chute) jusqu'à ce qu'il atteigne une passerelle ou une barre de franchissement à laquelle il s'accroche. S'il y a un trou dans la passerelle, *Lode Runner* passe à travers ce trou.

Il est également possible de faire chuter *Lode Runner* à partir d'une barre de franchissement. Dans ce cas, l'utilisateur doit appuyer sur la flèche en bas.

Prenez note que la chute est automatique. L'utilisateur perd le contrôle sur *Lode Runner* tout au long de sa chute (*Lode Runner* ne réagit plus aux touches du clavier). Il reprend le contrôle à la fin de la chute.

Voir **Démo 8.5.1 Chute dans le vide 1 Lode Runner.wmv**.

Voir **Démo 8.5.2 Chute dans le vide 2 Lode Runner.wmv**.

Voir **Démo 8.5.3 Chute dans un trou 1 Lode Runner.wmv**.

Voir **Démo 8.5.4 Chute dans un trou 2 Lode Runner.wmv**.

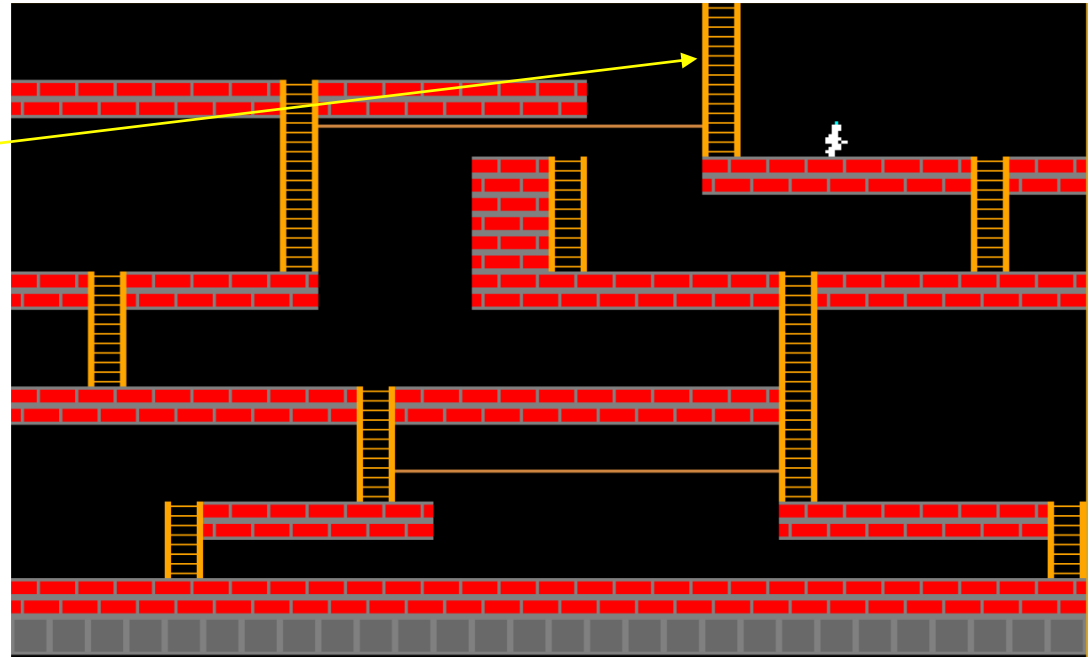
Voir **Démo 8.5.5 Chute de la barre Lode Runner.wmv**. (On appuie sur la flèche en bas)

## 8.6 Ramasser des lingots d'or

Dès que *Lode Runner* passe sur un lingot d'or, il le ramasse (le lingot disparaît de l'écran). Dès que le sixième lingot a été ramassé, une échelle s'ajoute automatiquement sur l'écran. C'est grâce à cette échelle que *Lode Runner* va avoir la possibilité de quitter le tableau et ainsi passer au niveau suivant.

Voir **Démo 8.6.1 Ramasser lingots Lode Runner.wmv**.

Voir **Démo 8.6.2 Ramasser sixième lingot Lode Runner.wmv**.



## 8.7 Sa vitesse

*Lode Runner* ne doit pas se déplacer à une vitesse ni trop rapide ni trop lente. Faites des tests.

## 8.8 Mourir

*Lode Runner* meurt lorsqu'il tombe dans un trou et que le trou se remplit sur lui. Au 1<sup>er</sup> niveau, cela se produit seulement dans un trou sur la passerelle du bas qui est en brique. Il ne peut plus en ressortir car la passerelle grise est en béton et *Lode Runner* ne peut pas faire de trou.

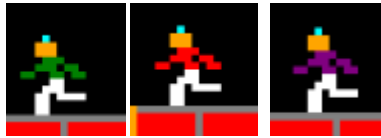
Voir **Démo 8.8.1 Mort de Lode Runner trou 1.wmv**.

Voir **Démo 8.8.2 Mort de Lode Runner trou 2.wmv**.

## 9. Les gardes

### 9.1 Leur apparence

Dans ce tableau, il y a 3 gardes. Les gardes ont exactement la même apparence que *Lode Runner*. Dans le jeu original, *Lode Runner* porte un chandail blanc tandis que les gardes portent un chandail rouge. Dans ma version, *Lode Runner* porte un chandail blanc tandis que chacun des gardes porte un chandail de couleur différente mais vous pouvez faire comme vous le voulez. L'important c'est de bien faire la différence entre un garde et *Lode Runner*.



### 9.2 Leur emplacement initial

Dans le jeu original, au point de départ, chacun des gardes est situé à un emplacement bien précis. Dans votre version du jeu, chacun des gardes doit être placé au hasard sur une passerelle. Chacun des gardes ne doit pas être placé sur la même passerelle que *Lode Runner* (c'est-à-dire celle du bas) ni à la même position qu'un lingot d'or.

**Attention** : ne placez pas les gardes sur une échelle ni sur une barre de franchissement ni dans un mur ni dans une passerelle ni dans le vide. De plus, initialement, aucun garde ne doit occuper la même position qu'un autre garde.

Voir **Démo 9.2.1 Emplacement initial gardes hasard 1.wmv**.

Voir **Démo 9.2.2 Emplacement initial gardes hasard 2.wmv**.

Voir **Démo 9.2.3 Emplacement initial gardes hasard 3.wmv**.

### 9.3 Leur déplacement

Les gardes se déplacent exactement de la même manière que *Lode Runner* et sont soumis aux mêmes règles. La différence c'est que leurs déplacements sont automatiques. Il ne faut pas que les déplacements des gardes soient « trop intelligents » car cela serait beaucoup trop difficile à jouer.

Dans ma version du jeu, chacun des gardes possède chacun son propre algorithme de déplacement mais vous pouvez faire comme vous le voulez, c'est-à-dire que vous pouvez utiliser le même algorithme de déplacement pour tous les gardes. Par contre, dans votre algorithme, évitez que vos gardes fassent trop de surplace. Un bon algorithme de déplacement est de le déplacer de manière à ce qu'il s'approche de plus en plus de *Lode Runner*.

Dans la version originale du jeu et dans ma version, aucun garde n'entre en collision avec un autre garde. Dès qu'une collision est sur le point de se produire, un des gardes change de direction ou reste stationnaire s'il ne peut pas changer de direction. Mais vous pouvez faire comme vous le voulez car cela est relativement difficile à programmer.

Voir **Démo 9.3.1 Déplacements gardes 1.wmv**.

Voir **Démo 9.3.2 Déplacements gardes 2.wmv**.

Voir **Démo 9.3.3 Déplacements gardes 3.wmv**.

### 9.4 Tomber dans un trou

Lorsqu'un des gardes tombe à l'intérieur d'un trou, celui-ci reste coincé à l'intérieur de ce trou. Cela a comme effet de boucher le trou. Par la suite, si *Lode Runner* ou un autre garde se déplace sur le trou, celui-ci ne tombe pas dans le trou (comme s'il n'y avait pas de trou).

Le garde qui est tombé dans un trou demeure dans le trou pendant 4 secondes. Au bout de 4 secondes, le garde sort du trou et poursuit sa route. Si, pendant ces 4 secondes, le trou s'est rebouché alors le garde meurt puis ressuscite sur une des cellules de la 2<sup>ème</sup> ligne du tableau (au hasard).

Voir **Démo 9.4.1 Garde tombe dans le trou en en ressort.wmv**.

Voir **Démo 9.4.2 Mort et renaissance du garde.wmv**.



## 9.5 Leurs chutes

Tout comme *Lode Runner*, lorsqu'un garde se déplace horizontalement, il est possible qu'il chute car il n'y a aucun objet pour le soutenir. Dans ce cas, il tombe dans le vide jusqu'à ce qu'il atteigne une passerelle ou une barre de franchissement à laquelle il s'accroche. Par contre, s'il y a un trou, le garde ne passe pas à travers ce trou.

Il est également possible qu'un garde chute à partir d'une barre de franchissement.

Voir **Démo 9.5.1 Chute dans le vide de la barre garde.wmv**.

Voir **Démo 9.5.2 Chute dans le trou garde.wmv**.

## 9.6 Leurs lingots d'or

Dès qu'un garde passe sur un lingot d'or, il le ramasse de manière automatique (le lingot disparaît de l'écran). Étant donné que le garde a ce lingot d'or en sa possession, *Lode Runner* ne peut plus le ramasser. Le garde qui a ramassé un lingot d'or peut le relâcher sur une passerelle à tout moment (au hasard mais pas trop souvent). Dans ce cas, *Lode Runner* (ou un autre garde) peut ramasser le lingot qui a été relâché.

Une autre manière pour *Lode Runner* de ramasser un lingot d'or qu'un garde a en sa possession, c'est de le faire tomber dans un trou. Lorsqu'il tombe dans un trou, le garde relâche automatiquement le lingot d'or juste au-dessus de lui.

Dans la version originale, on ignore si un garde a un lingot d'or en sa possession ou non. Dans la version que j'ai programmée, lorsqu'un garde a un lingot d'or en sa possession, ses pantalons sont dorés. J'aimerais que vous programmiez ma version c'est-à-dire que vous indiquiez visuellement si un garde possède un lingot d'or. Dans la version originale, un garde ne peut pas ramasser plus qu'un lingot.

Dans la version originale, un garde ne peut pas avoir plus qu'un lingot d'or en sa possession. Vous pouvez, si vous voulez, donner la possibilité à un garde de ramasser plus qu'un lingot.

Voir **Démo 9.6.1 Ramasser lingot garde.wmv**.

Voir **Démo 9.6.2 Relâcher lingot garde tombe dans le trou.wmv**.

## 9.7 Leur vitesse

Les gardes doivent se déplacer à une vitesse légèrement inférieure à celle de *Lode Runner*.

## 9.8 La mort de *Lode Runner*

*Lode Runner* meurt lorsqu'il entre en collision avec un ennemi.

Voir **Démo 9.8.1 Mort de Lode Runner collision 1.wmv**.

Voir **Démo 9.8.2 Mort de Lode Runner collision 2.wmv**.

Voir **Démo 9.8.3 Mort de Lode Runner collision 3.wmv**.

# 10. Les niveaux

Dans le jeu original, dès que *Lode Runner* quitte un tableau, il passe au niveau suivant et un nouveau tableau se dessine. Étant donné que vous ne devez programmer qu'un seul tableau, lorsque *Lode Runner* quitte un tableau, augmenter de un le nombre de gardes. Par exemple, au niveau 1, il doit y avoir 3 gardes; au niveau 2, il doit y avoir 4 gardes; au niveau 3, il doit y avoir 5 gardes. Ne programmez pas plus de 10 niveaux. Il est pratiquement impossible de passer au niveau suivant lorsqu'il y a 13 gardes.

Au début de chaque niveau, vous devez attendre que l'utilisateur appuie sur une touche avant de démarrer le jeu.

À tout moment, le niveau actuel ainsi que la durée écoulée doivent être dessinés.

Voir **Démo 10.1 Changement de niveau de 1 à 2 (de 3 gardes à 4 gardes).wmv**.

Voir **Démo 10.2 Changement de niveau de 2 à 3 (de 4 gardes à 5 gardes).wmv**.

## 11. Le score

**Lode Runner** ramasse un lingot : +250 points

Un garde tombe dans le trou : +75 pts

Un garde meurt (et ressuscite) : +75 pts

**Lode Runner** a réussi le niveau : +1500 pts

À tout moment, le score doit être dessiné.

## 12. Les vies de *Lode Runner*

Lorsque *Lode Runner* meurt, il perd une vie, perd tous les points qu'il a accumulés dans ce niveau et recommence le niveau à zéro. Par contre, il ne perd pas les points qu'il a accumulés dans les niveaux précédents. *Lode Runner* a 5 vies. Lorsqu'il a dépensé ses 5 vies, le jeu s'arrête (**game over**).

À tout moment, le nombre de vies doit être dessiné.

Voir **Démo 12.1 Lode Runner perd une vie (de 5 vies à 4 vies).wmv**.

Voir **Démo 12.2 Game Over.wmv**.

## 13. Les sons

Dans ce jeu, il doit y avoir un minimum de 9 sons différents (ou musique).

- Lorsque *Lode Runner* ramasse un lingot d'or
- Lorsque *Lode Runner* fait une chute
- Lorsque *Lode Runner* creuse un trou dans la passerelle
- Lorsqu'un trou se remplit automatiquement (après 8 secondes)
- Lorsque *Lode Runner* perd une vie
- Lorsque *Lode Runner* perd toutes ses vies (**Game over**)
- Lorsque *Lode Runner* a terminé un niveau pour passer au niveau suivant
- Lorsqu'un garde tombe dans un trou
- Lorsqu'un garde meurt

## 14. Les contraintes de programmation

- Vous devez programmer votre jeu à l'aide de cycles d'animation (comme cela a été fait dans le jeu du vortex).
- Vous devez programmer chacun des éléments du jeu dans des objets distincts (par exemple, le tableau des objets fixes, *Lode Runner*, les gardes, les sons, etc. sont des éléments du jeu qui doivent être programmés dans des objets distincts).
- Chacun de vos objets doit être complet. Les propriétés doivent décrire son état à n'importe quel moment du jeu.

Par exemple, dans l'objet de type *Lode Runner*, vous pouvez mettre sa position (x,y) sur l'écran, sa direction, sa vitesse, le nombre de pas qu'il a fait dans une direction donnée, le nombre de pixels à laquelle il se déplace, un indicateur qui indique l'aspect de *Lode Runner* (cela va être utilisé pour dessiner *Lode Runner* si vous l'animez), s'il fait une chute ou non, etc.

- Prévoyez des fonctions pour créer et initialiser chacun des objets.
- Prévoyez des fonctions pour dessiner chacun des objets dessinables. CETTE FONCTION NE DOIT JAMAIS MODIFIER L'ÉTAT DE CET OBJET NI DES AUTRES OBJETS. ELLE DOIT DESSINER L'OBJET. C'EST TOUT.
- L'état des objets doit être modifié à deux endroits spécifiques : dans l'événement du clavier et dans la fonction qui met à jour l'animation.
- À part des circonstances exceptionnelles, n'appellez JAMAIS directement la fonction qui dessine un objet ni celle qui met à jour l'animation. C'est le cycle d'animation qui doit s'en occuper.

## 15. La correction

Voici la pondération pour la correction :

- Le bon déroulement du jeu ainsi que le respect des règles du jeu : 80%
- L'interface : Bonne vitesse, fluidité des animations, qualité des dessins: 20%
- Non respect des contraintes de programmation : jusqu'à -20%