

ActionScript

7.2 – Méthodologie 2

iode conseil
contact@iode-conseil.com



Contenu

- Essai de typologie des jeux
 - Jeux de cliquer-chasser
 - Attraper et éviter
 - Jeux de tir
 - Jeux de balles
 - Quiz et énigmes
 - Puzzles
 - Jeux de cartes
 - Jeux de casino
 - Jeux d'arcade
 - Jeux de Labyrinthe

Tous jeux récupérés sur le web (différents endroits)
- Partie suivante
 1. Outils pour le développement
 - Débogueur
 - Trucs et astuces
 - Boîte à outils
 - ...

Essai de typologie

- Plusieurs typologies possibles...
- Ici : pas exhaustif / conception du jeu
 - Quelques éclairages
 - Quelques exemples



Jeux de cliquer-chasser

- Objectif
 - Cliquer sur une cible pour marquer des points
- Principe général
 - Pouvoir marquer les cibles et les comptabiliser
 - Fin : quand toutes les cibles sont trouvées.
 - Ne pas oublier : écran d'explication et écran de fin
- Exemple
 - **FindaFox**
- Approche générale
 - Trois écrans : trois images Flash
 - Clips pour les cibles
 - Test à chaque clip
 - Pas vraiment de difficulté
- Variantes
 - Type de nombres ou de mots
 - Questionnaire géographique
 - Avec apparition aléatoire : **Whackafox**
 - Avec tir : **Shootafox**

Attraper et éviter

- Objectif
 - Le joueur contrôle un personnage/objet qui se déplace horizontalement ; des objets tombent qu'il faut éviter ou attraper
- Principe général
 - Contrôle par touches de direction (limites : bords)
 - Chute aléatoire continue
 - Tests de collision
 - Fin : nombre prédéfinis d'objets, score, temps limite
- Exemple
 - **Applecatch**
- Approche générale
 - Déplacement : objet `Key`
 - Création des objets : `attachMovie` + positionnement, vitesse et couleurs aléatoires
 - Test de collision en continu
 - Difficulté : bon moteur de gestion des événements aléatoires
- Variantes
 - Attraper ET éviter : **Applelettercatch**
 - Slalom : **Riverkayak**

Jeux de tir

- Objectif
 - Le joueur contrôle un personnage/objet qui se déplace horizontalement et tire vers des objets qui tombent
- Principe général
 - Contrôle par touches de direction (limites : bords)
 - Chute aléatoire continue
 - Tirs et tests de collision avec le projectile
 - Fin : nombre prédéfinis d'objets, score, temps limite
- Exemple
 - **Balloonshoot**
- Approche générale
 - Déplacement : objet `Key`
 - Tir : objet `Key` + test pour garantir un tir/seconde
 - Création des objets : `attachMovie` + positionnement, vitesse et couleurs aléatoires
 - Test de collision en continu (`hitTest` ou test de proximité)
 - Difficulté : gestion aléatoire et fréquence du tir
- Variantes
 - Viser et tirer : **Aimandshoot**

Jeux de balles

- Objectif
 - Le joueur contrôle une palette qui se déplace horizontalement ; une balle rebondit contre des murs et frappe des cibles (jeux dérivé du « Pong » - années 1970)
- Principe général
 - Contrôle par touches de direction (limites : bords)
 - Rebondissement
 - Frappe contre des cibles (et destruction)
 - Fin : nombre prédéfinis d'objets détruits, score, temps limite, nombre de balles
- Exemple
 - **Paddleball**
- Approche générale
 - Déplacement : objet `Key`
 - Difficulté : rebondissement sur murs et palette – attention réalisme du test
 - Test de présence palette ou mur : `hitTest()`
- Variantes
 - 3D : **Wallball**
 - Tennis : **Poong-tennis**
 - Casse-briques : **Paddlebricks**

Quiz et énigmes avec des mots

- Objectif
 - Le joueur doit répondre à des questions, découvrir des mots
- Principe général
 - Succession de pages présentant une question
 - Cases à cocher
 - Fin : fin des questions
- Exemple
 - **Flashquizz**
- Approche générale
 - Questions dans fichier extérieur format txt ou XML : objet XML
 - Test sur numéro de réponse cochée
 - Succession d'images `gotoAndStop()`
- Variantes
 - Questions ou mots aléatoires à partir d'une liste
 - Sauvegarde des résultats : client-serveur, Flash + PHP
 - Rapidité :
<http://zut.languageskills.co.uk/resource/zut/source/year12/triviaexpression/flashtrivia.swf>
 - Jeu du pendu : **Hangstan**

Puzzles

- Objectif
 - Le joueur doit assembler des pièces afin de reconstituer une image
- Principe général
 - Un ensemble de pièces déplaçables
 - Une grille vide
 - Fin : puzzle reconstitué
- Exemple
 - **Jigsawpuzzle**
- Approche générale
 - Déplacement des pièces : `startDrag` et `stopDrag`
 - Test sur nouvelle position lors du relâchement (mise en place si position proche de la position réelle – un truc : faire que si toutes les pièces sont en position, leurs points d’alignement sont confondus)
 - Une difficulté : création des pièces (utiliser un canevas de découpage)
- Variantes
 - Temps limité
 - Puzzle coulissant : **Slidingpuzzle**
 - Jeu des correspondances (**Memory**) éventuellement en temps limité et avec comptage

Jeux de casino

- Objectif
 - Souvent de purs jeux de chance
- Principe général
 - Jeu simple mais interface complexe
 - Action sur levier et attente du résultat...
 - Les roues tournent et s'arrêtent sur un élément aléatoire ; les trois éléments qui apparaissent déterminent le résultat
 - Le gain s'affiche
 - Fin...
- Exemple
 - Slotmachine : <http://www.les-meilleurs-jeux-de-casino.com/slots.html>
- Approche générale
 - La seule difficulté : donner l'impression que les roues tournent → une animation avec 3 ou 4 images en boucle, contenant quelques hachures qui donneront une impression de flou
- Variantes
 - Poker vidéo : [Videopoker](#)

Jeux de cartes

- Objectif
 - Le joueur joue contre le donneur de cartes qui suit un certain nombre de règles
- Principe général
 - Des cartes données aléatoirement au joueur et au donneur
 - Un ensemble de cartes déplaçables
 - Fin : selon règles
- Exemple
 - **Blackjack**
- Approche générale
 - Tirage aléatoire de cartes : un tableau aléatoire prédéfini
 - Cartes du joueur et du donneur : deux tableaux distincts
 - Test à chaque coup
 - Difficultés : graphisme des cartes, élaboration des règles qui déterminent l'action du donneur
- Variantes
 - Avec mise de fond

Jeux d'arcade

- Objectif
 - Le joueur doit viser et détruire des objets se dirigeant vers son vaisseau
- Principe général
 - Contrôle par touches de direction (limites : bords)
 - Arrivages aléatoires et continus
 - Tirs et tests de collision avec le projectile
 - Endommagement du vaisseau
 - Fin : nombre prédéfinis d'objets, score, temps limite
- Exemple
 - **Spacecombat**
- Approche générale
 - Éléments actifs : astéroïde et missiles
 - Tir : objet `Key` + test pour garantir un tir/seconde
 - Création des objets : `attachMovie` + positionnement, vitesse et couleurs aléatoires
 - Test de collision en continu (`hitTest` ou test de proximité)
 - La difficulté : rendu 3D
 - Les astéroïdes grossissent `_xscale` et `_yscale`
 - Les missiles se dirigent vers la souris
- Variantes
 - 2 directions et rotation : **Spacerocks**
 - Atterrissage : **Lunarlander**

Jeux de labyrinthe

- Objectif
 - Le joueur conduit un personnage poursuivi dans un labyrinthe et qui doit récupérer des objets
- Principe général
 - Un labyrinthe
 - Contrôle par touches de direction
 - Les poursuivants se dirigent vers le joueur
 - Fin : tous les objets récupérés, sortie du labyrinthe, personnage attrapé
- Exemple
 - **Mazechase**
- Approche générale
 - La difficulté : le labyrinthe est formé d'une double rangée de petits carrés (utiliser la grille de Flash) ; les murs constituent un même remplissage **avec bordure** de cellules de la grille
 - Objets : occurrences placées à la main seulement aux intersections qui se trouvent dans les deux rangées – clips de deux images (on passe à l'image 2 vide quand le personnage touche l'objet)
 - Déplacement du personnage : au milieu de la double rangée ; objet `key + hitTest` avec les murs
 - Déplacement du poursuivant : au milieu de la double rangée, examen des possibilités et choix au hasard parmi ces possibilités
- Variantes
 - Gain de points
 - Combat possible contre le poursuivant

Manquants

- Fantaisies et gadgets
- Jeux d'assemblage
- Casse-têtes
- ...



Quelques liens

- Site de Gary Rosenzweig
 - <http://garyrosenzweig.com/>
- Jeux à télécharger
 - <http://www.t45ol.com/>
 - <http://www.flash-games.net/>
 - Goggle : “jeux Flash”

