

Jeu de plateforme

Le projet *jeu de plateforme* consiste à concevoir (algorithmique), développer (programmation), tester et documenter une application de jeu d'animation répondant à un certain nombre de spécifications et évaluée selon les critères présentés ci-dessous.

Le but du jeu consiste à faire arriver le personnage jusqu'à un certain objectif à travers un parcours muni de différents obstacles et bonus pouvant l'aider ou le défavoriser dans son but. Déplacer. Un exemple complet d'un tel jeu peut être consulté à l'adresse :

<http://www.gameslol.net/donkey-kong-365.html>



Règles du jeu (spécifications de base)

Les spécifications suivantes décrivent les objectifs à atteindre pour la version intermédiaire du projet dont le but est essentiellement d'implémenter un seul monde du jeu :

- ☐ La fenêtre initiale accueille le joueur en lui donnant la possibilité, via l'activation d'un bouton, de ...
 - ☐ ... lire les règles du jeu;
 - ☐ ... consulter les raccourcis clavier de l'application (stopper le jeu, mise en pause, déplacements, sauts du personnage, ...);
 - ☐ ... s'enregistrer en tant que nouvel utilisateur (nom, prénom, pseudo, mot de passe,...). Ces données seront sauvegardées dans un fichier externe ;
 - ☐ ... s'identifier (*pseudo+mot de passe*) ;
 - ☐ ... consulter les meilleurs scores (les siens ou ceux de tous les joueurs confondus) ;
 - ☐ ... commencer une nouvelle partie (un seul monde) ;
 - ☐ ... fermer définitivement l'application.
- ☐ Si un joueur est déjà identifié, il peut accéder à ses meilleurs scores (incluant la date ainsi que le nombre de points acquis pour la partie décrite). S'il n'est pas identifié, il peut consulter les meilleurs scores de tous les joueurs et parties confondus. Le pseudo du joueur associé à la partie décrite est alors donné.
- ☐ L'accès à une nouvelle partie ne peut pas se faire sans identification. Lors de l'identification, le mot de passe proposé par l'utilisateur doit être le même que celui associé au pseudo lors de l'enregistrement. Des messages d'erreur peuvent être proposés à l'utilisateur se trompant de pseudo ou de mot de passe.
- ☐ Une fois identifié, le joueur peut commencer une nouvelle partie se déroulant dans une fenêtre satellite de la fenêtre d'accueil du jeu ou dans la même fenêtre.
- ☐ En cours de partie, le joueur peut à tout moment mettre le jeu en pause ou arrêter définitivement la partie. Dans ce dernier cas, le joueur se retrouve dans la fenêtre d'accueil du jeu et son score n'est pas sauvegardé.
- ☐ Le plateau de jeu est formé d'éléments statiques comme des sols en pierre/terre, des échelles, des portails, ... et des éléments dynamiques comme le héros, éventuellement des ennemis, et des éléments de décors dynamiques comme des bonus qui disparaîtront quand le joueur sera passé dessus. Chaque élément est une image placée dans le canevas du jeu.
- ☐ Le plateau de jeu est décomposé en une grille virtuelle dont les cellules, identifiables à l'aide d'un système de coordonnées (ligne ; colonne), contiennent les différents éléments visuels du niveau : un espace, une brique, le personnage, une échelle, un bonus, ...
- ☐ L'état initial du niveau est encodé dans un fichier texte externe : chaque élément est représenté par un code arbitraire (par exemple, "0" = espace vide, "1" = brique, "2" = bonus, "3" = échelle, "4" = télé-portail, "5" = drapeau, "6" = pièce, "10" = personnage, "20" = ennemi,...). Lors du chargement du niveau, l'apparence visuelle du canevas est construite à partir de l'encodage du fichier à l'aide d'un analyseur de fichier (*parser*).

- ☐ Au début d'une partie, le joueur bénéficie de 3 vies et 0 point. A chaque reprise d'une partie après une pause, le jeu reprend au même état qu'il était avant la pause.
- ☐ Le personnage se déplace dans le niveau en courant et en sautant, se heurtant aux différents blocs présents dans le canevas : si ce bloc est un mur, le personnage s'arrête, si c'est un bonus ou un malus ou tout autre bloc, le personnage fera l'action souhaitée.
- ☐ A chaque changement de direction du personnage, son apparence est adaptée en fonction du sens de son déplacement.
- ☐ Un système d'étoiles bonus et de malus sera présent et permettra au joueur de gagner ou de perdre des points. Différents bonus et malus possédant d'autres fonctions que le nombre de points du joueur seront aussi présents, au nombre et fonctions que les créateurs choisiront.
- ☐ Si le personnage est heurté par un ennemi, il perd une de ses 3 vies et il devient pendant quelques secondes intouchables, lui permettant de se replacer. Le personnage pourra se débarrasser de ses ennemis en leur sautant sur la tête.
- ☐ Autour du champ de jeu sont disposées différentes informations relatives à la partie en cours :
 - ☐ Message d'accueil destiné au joueur (en fonction de son pseudo) ;
 - ☐ Affichage du score actuel du joueur ;
 - ☐ Affichage du nombre de vies encore à disposition du joueur ;
 - ☐ Affichage d'un timer au bout duquel, si le joueur n'a pas atteint l'objectif, la partie se termine et le joueur a perdu.
- ☐ Une fois la partie terminée, la fenêtre du jeu disparaît pour laisser place à une nouvelle fenêtre indiquant au joueur s'il a gagné ou perdu. De plus, la fenêtre affichera le score obtenu par l'utilisateur pour la partie jouée. Ce score sera automatiquement sauvegardé dans le fichier externe associé au joueur.

Gestion des niveaux et des scores (spécifications avancées)

Les spécifications suivantes décrivent les compléments à apporter à la version intermédiaire, compléments devant être atteints pour la version finale du projet :

- ☐ La fenêtre initiale accueille le joueur en lui donnant la possibilité, via l'activation d'un bouton, de...
 - ☐ ... lire le raccourci clavier de l'application permettant d'allumer-éteindre la musique associée à un niveau de jeu ;
 - ☐ ... s'enregistrer en tant que nouvel utilisateur (nom, prénom, pseudo, mot de passe,...). Ces données sont sauvegardées dans une base de données externe.
- ☐ Le joueur possède un niveau en fonction des scores qu'il aura réussi à atteindre. L'accès aux meilleurs scores indique le niveau atteint pour la partie décrite.
- ☐ Le jeu se compose de plusieurs mondes composés de plusieurs niveaux modélisés par autant de fichiers-texte. Lors du chargement d'un monde, l'apparence visuelle du canevas est construite à partir de l'encodage du fichier à l'aide d'un analyseur de fichier (*parser*).
- ☐ La valeur des différents bonus est décrite dans ce tableau :

Bonus	Points
Etoile jaune	100
Etoile multicolore	250
Etoile noire	-250
Etoile rouge	-100

- ☐ Au début d'une partie, le joueur se situe au niveau 1 du monde qu'il aura choisi et bénéficie de 0 pt et un timer est initialisé au temps choisi par les créateurs. Une fois arrivé à son objectif (un drapeau), le nombre de points est compté en fonction du temps et du nombre de bonus/malus pris et du nombre d'ennemis tués. L'état du score et des vies est ainsi mis à niveau après chaque partie, en fonction du résultat de cette dernière.
- ☐ Un joueur ne pourra pas accéder aux niveaux qu'il n'aura pas débloqués. Pour débloquent un niveau, il suffit d'avoir terminé celui d'avant. Le niveau et l'avancée du joueur seront donc sauvegardés.
- ☐ Chaque niveau est accompagné d'une musique pouvant être ou non écoutée par le joueur.
- ☐ Lors du passage du personnage sur les différents bonus/malus et ennemis tués, un son différent est émis.
- ☐ Autour du champ de jeu est affiché le niveau actuel du joueur ainsi que le monde et le niveau du monde dans lequel il se trouve.
- ☐ Les ennemis se déplaceront tout droit et seront soumis à la gravité comme le héros et ils changeront de direction dès qu'un mur sera touché. Selon le niveau, ils sortiront d'un portail et disparaîtront du monde au bout d'un chemin défini ou ils feront des aller-retours entre deux blocs qui les entravent.
- ☐ Une fois le niveau terminé, les résultats obtenus par le joueur (niveau atteint, nombre de points acquis, niveau actuel du joueur ainsi que le numéro du niveau terminé) sont sauvegardés dans la base de données externe.

Extensions possibles

La liste suivante décrit quelques extensions possibles du jeu. La qualité des extensions réalisées augmentera sensiblement la note finale de l'évaluation :

- ☐ Le joueur peut créer son propre niveau et le sauvegarder pour y jouer même après avoir quitté le jeu.
- ☐ Le joueur peut choisir son personnage dans une liste prédéfini (il serait enregistré avec le profil) et ce, jusqu'à la modification de ce paramètre dans les réglages (personnage valable pour tous les mondes).
- ☐ Le personnage peut tuer les ennemis en lançant ce qu'il reçoit en bonus.
- ☐ Le déplacement des ennemis se fait de manière intelligente, le but étant de piéger le héros.
- ☐ L'interface du jeu est programmée à l'aide du module *Pygame* (<http://www.pygame.org/news.html>), *PyQt* (<http://ogirardot.developpez.com/introduction-pyqt/>) ou *Panda 3D* (<https://www.panda3d.org/>).
- ☐ Toute autre idée ou innovation rendant le jeu plus attractif pour le joueur !