

## Présentation du jeu :

Lorsque le jeu démarre, une fenêtre d'accueil s'affiche avec un bouton Play au milieu. En appuyant sur play, l'utilisateur arrive sur la fenêtre des niveaux (3) et peut en sélectionner un en appuyant sur celui-ci. L'ordre des niveaux allant du plus facile au plus compliqué. Plus le niveau augmente, plus les obstacles sont nombreux et la vitesse de défilement du background est rapide.

En cliquant sur un niveau, le jeu démarre. Pour sauter au-dessus d'un obstacle, il faut appuyer sur la touche espace. Pour sauter en continu, il suffit de rester appuyé dessus. Si collision avec un obstacle, game over.

## Analyse :

### Tâches prévues → tâches réalisées :

- Faire défiler le même fond tout le long de la partie : (3 jours)  
→ Réalisé grâce au « parallax scrolling ». Deux fonds utilisés (même image), un qui prend la largeur de la fenêtre et l'autre qui est à droite de la fenêtre (non visible). On fait défiler l'écran vers la gauche et quand un des fonds est égal à (-) la taille de la fenêtre, on le replace à droite (position en x = la largeur de la fenêtre + la largeur du fond)
- Faire sauter le cube : (2h)  
→ Association de la touche espace au saut. Diminution progressive de la position en y du cube (qui fait monter le cube car l'axe des y est dirigé vers le bas), puis on le fait retomber grâce à la gravité.
- Ajout des sprites des obstacles sur le fond défilant : (5j)  
→ Réalisé en stockant les différents types d'obstacles dans un tableau (chacun avec une texture et un type).  
Quand le fond défile, une boucle for lit les obstacles et les placent sur la fenêtre en fonction de la position définie pour chacun d'entre eux et le type appelé. (1. Bloc / 2. Carre / 3. Triangle....)
- Collision lorsque le cube touche un obstacle triangle et bloc en y (côté gauche) (1semaine)  
→ on définit l'action au moment où le cube et l'obstacle se croisent (collision de 2 sdl\_rect ou hitbox), on définit une action quand ils se rencontrent (game over quand on touche un triangle, nouveau sol si on touche la surface horizontal d'un carré...)
- Fenêtre de win lorsque le niveau est complété à un moment décidé de la partie  
→ Pas réalisé par manque de temps. Il aurait fallu définir un moment de contact avec un obstacle à 10 000px par exemple et le qualifier de fin du

niveau. Temps pris élevé car il faut gérer ce qui se passe après la fenêtre de win et vers quelle page elle redirige après (menu ? page d'accueil ?)

- Fenêtre de game over lorsqu'il y a collision  
→ pas réalisé par manque de temps également mais un message dans la console s'affiche (game over), et l'utilisateur revient sur la page d'accueil.

## Problèmes rencontrés / Bugs :

### Problèmes mineurs :

- Écran saccadé, peut-être dû à une mauvaise organisation du code. Car problème résolu du cube qui devenait transparent juste en changeant l'ordre d'appel de mes fonctions
- Complication pour dépasser la création d'un niveau. Dès qu'on gère un menu avec plusieurs niveaux, le code est divisé en d'autant plus de parties et on se perd très vite dans l'organisation du code et les fonctions s'entremêlent. Le nom des fonctions et leur appel doit être très précis pour ne pas les confondre.
- Manque de versionnage. Il aurait fallu garder plusieurs versions de mon code pour ne pas tout perdre quand je modifiais des choses.

### Problèmes majeurs :

- Free des tableaux au fur et à mesure. Segmentation fault (11). Avec leaks sur mac, on voit les blocs alloués qui n'ont pas été libérés mais on n'a pas de détails sur la partie du code auxquels ils correspondent (fonction/fichier) ni sur les seg fault 11. Il a fallu faire des free à plusieurs endroits et recompiler afin de ne plus avoir de fuites de mémoire.
- Association du saut du cube avec la position des obstacles. Il aurait fallu faire les collisions avant pour pouvoir être sûr de l'emplacement des obstacles en fonction du cube mais cela rend la pose des obstacles compliquée car à chaque aperçu de jeu, nous sommes obligés de gagner pour voir l'avancement (plus le niveau avance et plus cela devient un problème)
- Gérer la distance entre chaque obstacle pour être sûr que le cube puisse parcourir la distance entre ceux-ci en prenant en compte la vitesse de défilement et la durée de saut. Impossible de tester si nos obstacles sont fonctionnels avant de les avoir tous posés (les tester au même temps qu'on les pose nous oblige à jouer le jeu sans se faire éliminer jusqu'au bout du niveau).

## Problèmes bloquants :

- Gerer la taille des hitbox. Hitbox trop grandes, elles rendent le jeu impossible, si le moment de saut n'est pas parfaitement calculé, game over.