**Contrôle qualité**

Précieux Guide d’intégration pour l’Épopée fantastique

Symboles et classes de base

* Dans ce projet, les symboles dont la mécanique est contrôlée par une classe possèdent une classe générée dynamiquement (ex. TabVillage), mais ils doivent avoir une classe de base liée à un script réel (ex. Tableau).
* C’est leur classe de base qui leur donne leurs fonctionnalités. Il est donc important de ne pas utiliser la classe de base habituelle «flash.display.MovieClip» pour créer la *liaison* *ActionScript* d’un élément.
* Les noms des symboles liés à une classe de base possèdent tous un préfixe pour *rappeler* leur fonctionnement (ex. ObsRoche est un Obstacle ; TabVillage est un Tableau).

Création de tableaux et de liens entre les tableaux

* La création d’un nouveau tableau exige l’ajout de son nom dans une des listes des tableaux du script Jeu.as (\_tTableauxPacifiques ou \_tTableauxDangereux). Sans cet ajout, le tableau n’est pas activé.
* L’accès à un tableau est rendu possible par la présence d’un duo porte et téléport. Une porte permet de quitter un tableau, tandis qu’un téléport permet d’y arriver.
* Pour créer un nouveau lien, une porte *menant* vers un nouveau tableau doit être placée dans un tableau *existant*, et un téléport doit être placé dans le *nouveau* tableau. C’est le téléport qui détermine à quel endroit le personnage sera «téléporté» à son arrivée dans le tableau.

Noms des instances

* De nombreuses mécaniques du jeu sont basées sur les noms d’instances des éléments tels que les portes, téléports, PNJ et objets (ex. patates).
* Les conventions pour nommer les instances des portes et téléports sont particulièrement importantes. Le nom d’une instance de porte *doit* être porte\_*Destination*\_mc (où *Destination* est le nom du tableau à atteindre). Le nom d’une instance de téléport *doit* être teleport\_*Provenance*\_mc (où *Provenance* est le nom du tableau d’où arrive le joueur).
* Les noms d’instance des PNJ et objets interactifs doivent correspondre aux noms prévus dans les scripts PNJ.as et Objet.as. À noter : si un même nom est donné à plusieurs instances dans des tableaux différents, ils seront gérés comme si c’était un seul et même élément (ex. si une pièce d’or dans le village s’appelle «piece1\_mc» et qu’une autre pièce d’or dans le château porte le même nom, le ramassage de l’une des pièces provoquera la disparition des deux!)
* Les autres obstacles peuvent être nommés automatiquement par *Animate*, car leurs noms ne sont pas utiles au fonctionnement (ex. maisons, arbres, roches, clôtures)

Éléments interactifs et collisions

* Il suffit de placer des obstacles dans un tableau pour qu’ils bloquent le passage (ex. clôtures, arbres). L’ajout d’un objet ou PNJ doit généralement être accompagné d’un ajout de code dans leur classe.
* Tous les éléments interactifs des tableaux (ex. patates, clôtures, maisons, roches, arbres, portes, PNJ) doivent avoir une classe de base Obstacle ou une classe qui en dérive (ex. ObsPorte).
* La forme de chaque élément est considérée pour la détection des collisions, sauf si le symbole contient une instance d’un clip qui possède la classe de base Collision (cercleDeCollision\_clip ou rectDeCollision\_clip). Dans ce cas, c’est le sous-clip qui sert à la détection de la collision. La visibilité de ce sous-clip est gérée automatiquement à l’exécution du jeu. L’utilisation d’un sous-clip de collision permet d’ajuster les zones de collision pour que l’expérience utilisateur soit plus agréable.
* La position du *point centre* d’un clip utilisé dans les tableaux est très importante, car c’est la position y de chaque clip qui est utilisée pour gérer la profondeur des éléments sur la scène (c’est ce qui permet au personnage de passer devant et derrière d’autres éléments). Ne pas oublier que la profondeur dans les tableaux est gérée à l’exécution (donc l’ordre z des éléments en mode auteur à peu d’importance).

Monstres et personnages

* Les monstres et les personnages doivent avoir une classe de base Monstre ou Perso.
* Les symboles des monstres et personnages doivent minimalement posséder des étiquettes «Vivant» et «Mort». D’autres étiquettes pourraient s’avérer utiles et nécessaires au fil du projet.
* Les monstres et personnages sont toujours placés par programmation. Ils ne doivent *jamais* être placés *manuellement* dans un tableau ou une autre interface, car ils causeraient des bogues. Au besoin, une «copie désactivée» peut être placée dans un PNJ (pour désactiver la classe sur cette copie, l’instance doit être au mode symbole *graphique* avec l’option «image unique»).
* Rappel : un PNJ est un obstacle, ce n’est pas un Être/Monstre/Perso.

Mode de débogage et tests automatisés

* Pour accéder au mode de débogage, il faut d’abord appuyer sur CTRL+SHIFT+D («D» pour débogage…), puis il faut entrer le mot de passe : «TIM». En cas d’erreur lors de l’entrée du code secret, il faut relancer du début le processus d’accès au débogage.
* Une fois le débogage activé, il est possible d’utiliser les raccourcis clavier du débogage. Ainsi, pour lancer les tests automatisés des tableaux et de leurs composantes, il faut utiliser la combinaison CTRL+SHIFT+T («T» pour tests!)
* Les tests automatisés traversent tous les tableaux et vérifient les noms des instances importantes ainsi que la présence des duos porte et téléport. À la fin du test, un rapport indique les erreurs dans la fenêtre de sortie de *Animate*.