

8

Conception de menus

Afin de créer un exemple de jeu complet, vous allez voir au long de ce chapitre comment créer une scène distincte de la scène de l'île qui servira de menu. Plusieurs méthodes existent pour concevoir des menus dans Unity, qui combinent des comportements intégrés et des textures 2D.

Les composants GUI TEXTURE s'utilisent pour les écrans de démarrage incluant les logos des développeurs ou les écrans de chargement. Par contre, pour les menus interactifs, deux manières différentes sont possibles. La première repose sur l'utilisation des composants GUI TEXTURE – que vous avez étudiés au Chapitre 6, "Instanciation et corps rigides", pour ajouter un viseur, et au chapitre précédent, pour afficher la boîte d'allumettes. L'autre méthode se base sur les classes UnityGUI et sur les ressources GUI skin pour créer une interface graphique personnalisée.

Au cours de ce chapitre, vous apprendrez à :

- concevoir une interface de deux manières différentes ;
- contrôler les composants GUI TEXTURE avec des événements souris scriptés ;
- rédiger un script simple UnityGUI ;

- paramétrer les ressources GUI skin ;
- charger les scènes, les menus et le niveau de jeu.

Vous découvrirez deux méthodes pour ajouter des menus interactifs :

- **Méthode 1** : composants GUI TEXTURE et contrôle des événements souris par le script. Cette méthode consiste à créer des objets GUI TEXTURE puis à gérer l’affichage de leurs textures à l’aide de scripts basés sur les événements souris – survol du menu, clic, bouton relâché. Elle nécessite moins de code pour créer les boutons eux-mêmes, mais toutes les actions doivent être contrôlées par des scripts.
- **Méthode 2** : classe UnityGUI et ressources GUI skin. Avec cette méthode, la totalité du menu est créée à l’aide de scripts, contrairement à la précédente dans laquelle on crée des objets de jeu.

La création des éléments de menu demande donc plus de code au début, mais des ressources GUI skin peuvent être créées pour définir l’apparence et le comportement des éléments dans le panneau INSPECTOR.

Les développeurs préfèrent généralement la seconde méthode pour créer des menus de jeu complet en raison de sa plus grande flexibilité. Les éléments du menu eux-mêmes sont créés dans le script, mais leur aspect est défini à l’aide des composants GUI. Cela est comparable à l’utilisation du code HTML et des CSS lors de la conception de pages web – les CSS (*Cascading Style Sheets*, feuilles de styles en cascade) contrôlent l’apparence tandis que le code HTML fournit le contenu.

Vous verrez que les composants GUI appliquent certaines conventions des CSS, pour les marges, le padding et la justification, ce qui vous sera utile si vous avez une expérience du développement web. Cependant, ne vous inquiétez pas si ce n’est pas le cas ; à l’instar de la plupart des fonctionnalités de Unity, les paramètres des composants GUI sont conçus pour qu’un minimum de connaissances préalables soit nécessaire.

Les interfaces et les menus

Les menus servent le plus fréquemment pour définir les contrôles et ajuster les paramètres du jeu – les options graphiques et sonores – ou pour charger ou sauvegarder une partie. Quel que soit le jeu, il est essentiel que le menu qui l’accompagne ne gêne pas l’accès au jeu ou à l’un de ses paramètres. C’est avant tout du jeu lui-même dont on se souvient, à moins que les menus ne soient particulièrement réussis ou mal conçus.

Les développeurs cherchent souvent à relier le menu du jeu au thème ou au principe du jeu. Par exemple, dans l’excellent *World of Goo* de 2D Boy, le curseur prend l’apparence d’une boule de Goo (les petites créatures rondes du jeu) suivie d’une traînée lumineuse dans les

menus afin de créer un lien visuel entre le concept du jeu et son interface. Ainsi, la navigation dans le menu d'ouverture permet déjà au joueur de s'amuser.

Dans *LittleBigPlanet* de Media Molecule, ce concept est poussé encore plus loin, puisque le joueur doit apprendre à contrôler le personnage jouable pour pouvoir naviguer dans le menu.

Comme pour toute création, l'important est de rester cohérent. Dans notre exemple, vous devez veiller à utiliser des couleurs et une typographie en harmonie avec le contenu du jeu. Vous avez peut-être déjà rencontré des jeux mal conçus et qui utilisent trop de polices de caractères différentes ou trop de couleurs qui jurent. Or, le menu étant le premier élément que le joueur découvre, ces erreurs créent d'emblée un sentiment peu agréable et nuisent à la viabilité commerciale du jeu.

Les textures employées pour créer le menu de notre exemple de jeu sont disponibles sur la page web consacrée à cet ouvrage (www.pearson.fr). Décompressez si nécessaire le fichier nommé `Menu.unitypackage` puis revenez dans Unity. Cliquez sur `ASSETS > IMPORT PACKAGE`, parcourez votre disque dur jusqu'à l'emplacement de ces ressources et importez-les.

Une fois l'importation terminée, un dossier `Menu` s'affiche dans le panneau `PROJECT`. Il contient les éléments suivants :

- des textures pour le menu principal du jeu et trois boutons : `PLAY`, `INSTRUCTIONS` et `QUIT` ;
- une séquence audio (un bip pour les boutons de l'interface).

Créer le menu principal

À cette section, vous utiliserez l'île elle-même comme toile de fond de vos menus afin de donner aux joueurs un avant-goût de l'environnement qu'ils exploreront et de fixer le cadre visuel du jeu.

Vous allez dupliquer l'île existante, l'afficher en plaçant une caméra à une certaine distance, puis vous superposerez des éléments d'interface 2D en utilisant les deux méthodes que nous avons mentionnées au début de ce chapitre.

Créer la scène

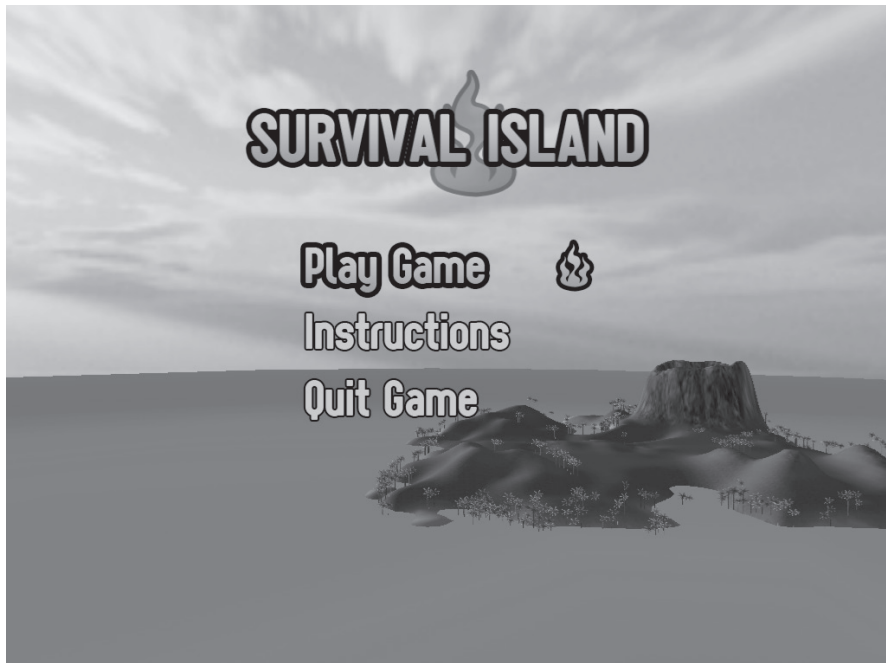
Pour ces menus, l'environnement insulaire que vous avez créé va vous servir. En plaçant l'île à l'arrière-plan dans le menu, vous éveillerez la curiosité du joueur en lui présentant l'environnement qu'il pourra explorer lors du lancement de la partie. Ce type d'incitation

visuelle peut sembler anodin, mais cela encourage le joueur à tester le jeu de manière quasi subliminale.

Un exemple visuel

La Figure 8.1 montre l'aspect qu'aura votre menu une fois terminé. Cet exemple est obtenu en suivant la méthode 1 ; les éléments de menu créés avec la méthode 2 auront un aspect différent.

Figure 8.1



Dupliquer l'île

Pour commencer, vous réutiliserez la scène ISLAND LEVEL que vous avez créée aux chapitres précédents. Pour faciliter la gestion des éléments, vous regrouperez les ressources essentielles de cet environnement afin de les distinguer des objets dont vous n'avez pas besoin, comme l'avant-poste et les piles. De cette façon, il vous suffira ensuite de tout supprimer à l'exception du groupe que vous allez créer dans la copie du niveau. Il est aussi facile de regrouper des éléments dans Unity que d'imbriquer des objets enfants dans un objet parent vide.

Regrouper les objets de l'environnement

Cliquez sur GAME OBJECT > CREATE EMPTY.

Un nouvel objet nommé GAMEOBJECT est alors créé. Renommez-le *Environment*.

Dans le panneau HIERARCHY, faites glisser les objets DIRECTIONAL LIGHT, DAYLIGHT SIMPLE WATER et TERRAIN sur l'objet vide ENVIRONMENT. Vous êtes maintenant prêts à dupliquer ce niveau pour créer le menu dans une scène distincte.

Dupliquer la scène

Suivez les étapes suivantes pour dupliquer la scène :

1. Cliquez sur FILE > SAVE SCENE pour vous assurer que la scène ISLAND LEVEL est enregistrée, puis sélectionnez la ressource ISLAND LEVEL dans le panneau PROJECT.
2. Cliquez sur EDIT > DUPLICATE ou utilisez les raccourcis clavier Cmd/Ctrl+D. Lors de la duplication, Unity ajoute simplement un numéro au nom de l'objet ou de la ressource, si bien que cette copie est baptisée *Island Level 1*. Assurez-vous qu'elle est sélectionnée dans le panneau PROJECT puis renommez-la *Menu*.
3. Double-cliquez sur cette scène dans le panneau PROJECT pour la charger et commencer à la modifier.
4. La scène MENU étant ouverte, vous pouvez maintenant supprimer tous les objets inutiles. Appuyez sur Cmd/Ctrl puis cliquez sur tous les objets dans le panneau HIERARCHY à l'exception du groupe ENVIRONMENT pour les sélectionner.
5. Appuyez sur Cmd+Retour arrière (Mac OS) ou Maj+Suppr (Windows) pour les supprimer de cette scène.

Comme l'illustre la Figure 8.1, l'île apparaîtra vue de loin et dans le coin inférieur de l'écran dans le menu, tandis que le titre du jeu et le menu s'afficheront en surimpression du ciel et de la mer. Pour reproduire cela, vous devez ajouter une nouvelle caméra dans la scène, car la seule caméra existante était celle attachée à l'objet FIRST PERSON CONTROLLER. Il n'y a donc aucune caméra dans la scène actuelle, rien qui permette de visualiser le monde 3D, si bien que le point de vue dans le panneau GAME est maintenant complètement vide.

Pour créer une caméra, cliquez sur GAMEOBJECT > CREATE OTHER > CAMERA. Assurez-vous que ce nouvel objet CAMERA est sélectionné dans le panneau HIERARCHY, puis entrez les valeurs de POSITION suivantes dans le composant TRANSFORM du panneau INSPECTOR : (150, 250, -650).

Annuler le mip mapping

Le *mip mapping* est une technique permettant de générer de plus petites versions de textures distantes dans un moteur de jeu, ce qui améliore les performances (jusqu'à 33 % dans le moteur de Unity). Toutefois, il ne s'applique qu'aux textures intégrées dans le monde en 3D et non aux textures 2D comme celles que vous allez utiliser pour le titre et les trois boutons de menu. Vous devez donc désactiver cette fonction dans les paramètres d'importation de chaque ressource.

Commencez par sélectionner le fichier de texture `MAINTITLE` dans le dossier `Menu` du panneau `PROJECT` pour afficher les paramètres d'importation de cette texture dans le panneau `INSPECTOR`. Désactivez ensuite l'option `GENERATE MIP MAPS`.

Répétez cette étape pour les textures suivantes dans le dossier `Menu` :

- **InstructionsBtn**
- **InstructionsBtnOver**
- **PlayBtn**
- **PlayBtnOver**
- **QuitBtn**
- **QuitBtnOver**

Ajouter le titre

Maintenant, vous avez besoin d'un logo pour le jeu. Le plus simple consiste à partir d'une texture que vous avez conçue et à la définir comme un objet `GUI TEXTURE` dans Unity.

Les formats des objets GUI Textures

Sélectionnez la texture `MAINTITLE` dans le dossier `Menu` du panneau `PROJECT`. Cette texture créée dans Photoshop est enregistrée au format `TIFF` (*Tagged Image File Format*), un format qui convient pour toutes les textures que vous comptez utiliser pour vos interfaces graphiques. Il prend en charge la transparence en haute qualité non compressée, ce qui permet d'éviter certains problèmes avec les contours blancs que présentent les formats `GIF` (*Graphics Interchange Format*) ou `PNG` (*Portable Network Graphics*) notamment.

Créer les objets

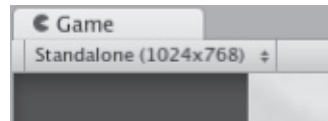
Cette texture toujours sélectionnée, cliquez sur `GAMEOBJECT > CREATE OTHER > GUI TEXTURE` pour créer un nouvel objet `GUI TEXTURE` basé sur cette texture. Nous l'avons

déjà vu, les dimensions de la texture sélectionnée sont automatiquement définies. Un nouvel objet, `MAINTITLE`, s'affiche dans le panneau `HIERARCHY` (l'objet prend le nom du fichier sélectionné lors de sa création).

Positionnement

La plupart des ordinateurs actuels sont capables de gérer des résolutions égales ou supérieures à 1024×768 pixels. Vous allez donc utiliser cet objet comme une norme pour tester votre menu. Vous définirez sa position de façon dynamique avec la classe `Screen`, afin que cela fonctionne dans d'autres résolutions. Le menu déroulant situé dans le coin supérieur gauche du panneau `GAME` (voir Figure 8.2) permet de spécifier différents ratios d'écran et résolutions. Cliquez dessus et sélectionnez `STANDALONE (1024 x 768)` pour afficher un aperçu de cette résolution.

Figure 8.2



Gardez à l'esprit que si la résolution de votre propre ordinateur est proche de celle que vous sélectionnez, le panneau `GAME` n'affichera pas un aperçu exact mais une version plus petite de l'image, car l'interface de Unity occupe une petite partie de l'écran.

Pour afficher le panneau `GAME` en mode plein écran (ainsi que tous les panneaux de l'interface de Unity) et le réduire, placez le curseur dessus, puis appuyez sur la barre d'espacement. Lorsque vous positionnez des éléments d'interface graphique, cette méthode vous permet de prévisualiser l'apparence de cette interface dans le jeu fini.

Par défaut, tous les objets `GUI TEXTURE` se placent à $(0.5, 0.5, 0)$, ce qui correspond au milieu de l'écran. On se souvient que les éléments 2D utilisent les coordonnées de l'écran comprises entre 0 et 1.

Définissez la position de l'objet `MAINTITLE` à $(0.5, 0.8, 0)$ dans le composant `TRANSFORM` du panneau `INSPECTOR` pour placer le logo du titre principal du jeu en haut et au centre de l'écran. Maintenant, vous allez ajouter trois boutons à l'aide d'objets `GUI TEXTURE` supplémentaires.

La création du menu – méthode 1

Avec cette première méthode, vous créez un menu avec une texture de fond transparente comme GUI Texture, de la même manière que vous venez de le faire avec le logo du titre principal.

Vous aurez ensuite besoin d'écrire un script afin que la texture puisse recevoir les événements souris (lorsque le curseur survole ou quitte le bouton et lorsque le joueur clique ou relâche le bouton de la souris).

Ajouter le bouton Play/Jouer

Sélectionnez la texture PLAYBTN dans le dossier Menu du panneau PROJECT, puis cliquez sur GAMEOBJECT > CREATE OTHER > GUI TEXTURE. Sélectionnez l'objet PLAYBTN que vous venez de créer dans le panneau HIERARCHY et réglez sa position dans le composant TRANSFORM du panneau INSPECTOR à (0.5, 0.6, 0).

Le script du bouton *GUI Texture*

Comme il s'agit du premier des trois boutons que vous allez créer et qu'ils auront tous des fonctions en commun, ce script pourra être utilisé pour les trois boutons. Pour cela, vous choisirez des variables publiques pour définir les différents paramètres. Ainsi, chaque bouton doit :

- lancer la lecture d'un son lors du clic ;
- charger un autre niveau (ou une *scène*, selon la terminologie de Unity) lors du clic ;
- changer de texture lorsque la souris le survole pour être mis en évidence.

Sélectionnez le dossier Scripts dans le panneau PROJECT, cliquez sur le bouton CREATE et sélectionnez JAVASCRIPT dans le menu déroulant. Renommez le script *MainMenuBtns*, puis double-cliquez sur son icône pour l'ouvrir dans l'éditeur de script.

Commencez par déclarer les quatre variables publiques suivantes au début du script :

```
var levelToLoad : String;  
var normalTexture : Texture2D;  
var rolloverTexture : Texture2D;  
var beep : AudioClip;
```

La première variable stockera le nom du niveau à charger lorsque le joueur cliquera sur le bouton sur lequel ce script est appliqué. Le fait de placer ces informations dans une variable

permet d'appliquer ce script à tous les boutons, puisque le niveau à charger est défini par le nom de la variable.

Les deuxième et troisième variables sont déclarées comme des variables de type `Texture2D`. Aucune texture en particulier n'est définie afin qu'elles puissent être assignées ensuite par glisser-déposer dans le panneau `INSPECTOR`.

Enfin, la variable de type `AudioClip` permet de lancer la lecture d'un son au clic de la souris.

Ajoutez maintenant la fonction suivante pour définir la texture utilisée par le composant `GUI_TEXTURE` lorsque le curseur survole la zone que la texture occupe (on appelle généralement cet état *rollover* ou *survolé*) :

```
function OnMouseEnter(){
    guiTexture.texture = rolloverTexture;
}
```

Le paquet de ressources `Menu` que vous avez importé contient une texture pour l'état normal et l'état survolé de chaque bouton. Cette fonction `OnMouseEnter()` définit simplement que la texture utilisée dans le composant correspond à la valeur attribuée à la variable publique `rolloverTexture` dans le panneau `INSPECTOR`. Afin de savoir quand le curseur s'éloigne ou sort de la limite de cette texture, ajoutez la fonction suivante :

```
function OnMouseExit(){
    guiTexture.texture = normalTexture;
}
```

Sans cette seconde fonction, la texture `rolloverTexture` resterait affichée. Le joueur verrait alors cette option de menu toujours en surbrillance.

Pour gérer le son et le chargement de la scène appropriée, ajoutez la fonction suivante :

```
function OnMouseUp(){
    audio.PlayOneShot(beep);
    yield new WaitForSeconds(0.35);
    Application.LoadLevel(levelToLoad);
}
```

La première commande concerne le son. Elle permet d'être sûr qu'il se lancera avant le chargement de la scène suivante et qu'il ne sera pas coupé. La commande `yield` placée entre le déclenchement du son et le chargement de la scène suivante indique au script de faire une pause du nombre défini de secondes. Vous pouvez ainsi créer rapidement un délai sans devoir utiliser une commande `timer`.

La commande `yield` sert ici à créer un retard, mais elle peut également servir à exécuter tout un ensemble d'instructions avant la ligne de code suivante. En programmation, on

appelle cette unité de traitement une *coroutine* pour la distinguer des autres procédures – les *routines*.

Ensuite, la commande `Application.LoadLevel()` charge la scène dans le jeu, en utilisant la valeur de la variable publique `levelToLoad` indiquée dans le panneau INSPECTOR pour trouver le fichier approprié.



Pour les interfaces, il est généralement conseillé de déclencher les actions lorsque le bouton de la souris est relâché (mouse up) plutôt qu'au moment du clic (mouse down). Cela permet au joueur de déplacer le curseur sans valider lorsqu'il s'est trompé et a sélectionné le mauvais élément.

Comme vous lancez la lecture d'un son, vous devez vous assurer que l'objet possède un composant `AudioSource` en ajoutant la ligne `RequireComponent` suivante à la fin du script :

```
@script RequireComponent(AudioSource)
```

Cliquez sur `FILE > SAVE` dans l'éditeur de script puis revenez dans Unity. Sélectionnez l'objet `PLAYBTN` dans le panneau `HIERARCHY`, puis cliquez sur `COMPONENT > SCRIPTS > MAIN MENU BTNS` pour lui appliquer le script que vous venez d'écrire. Le script doit alors apparaître sur la liste des composants de l'objet `PLAYBTN` dans le panneau `INSPECTOR`. En raison de la ligne `RequireComponent`, l'objet dispose également d'un composant `AUDIO SOURCE`.

Assigner des variables publiques

Les variables publiques doivent être assignées avant que le script ne puisse fonctionner. Entrez le nom *Island Level* dans le champ de la variable `LEVEL TO LOAD`, afin que ce niveau se charge lorsqu'on clique sur le bouton. Faites ensuite glisser les textures `PLAYBTN` et `PLAYBTN OVER` depuis le dossier `Menu` du panneau `PROJECT` sur les variables `NORMAL TEXTURE` et `ROLL OVER TEXTURE` respectivement. Enfin, faites glisser la séquence audio `MENU_BEEP` du même dossier sur la variable `BEEP`. Une fois terminé, le composant doit être identique à celui de la Figure 8.3.

Figure 8.3

