

# Document technique

# Aperçu du jeu

**Titre** : Mer'Veille

**Plateforme** : PC

**Cibles** : Tout public (âgé de 7ans et +)

**Éditeur** : Unity Technologies

## Description du jeu

Mer'Veille est une **animation immersive** qui plonge les visiteurs dans l'**univers aquatique** tout en fournissant des **informations** sur les **espèces présentes**. Utilisant un **écran de projection**, ce concept crée une **expérience visuelle captivante**. Les visiteurs sont **entourés de projections des environnements marins**, avec des **commentaires visuels** détaillant les **caractéristiques** de diverses **espèces**. Cette immersion totale vise à **émerveiller** et à **éduquer**, en laissant une **impression durable** et en **approfondissant la compréhension** et l'**appréciation** des visiteurs pour le monde sous-marin.

# Concept

Mer'Veille permet de faire découvrir aux visiteurs des **informations sur les espèces marines** présentes dans l'aquarium en utilisant la **technologie Kinect**. Il s'agit d'un jeu **ludique et éducatif** qui permet aux visiteurs de **comprendre le comportement** des poissons et d'**apprendre** des faits intéressants sur chaque espèce. Il y aura **deux modes de jeu en un seul**, pour le **premier** le visiteur sera **face à plusieurs espèces** de poisson où il devra **sélectionner le type de poisson** dont il veut l'information et pour **le deuxième** il sera plongé **au cœur de l'océan**, à l'aide de la caméra il pourra **se voir comme si il s'y trouvait vraiment**, et à travers un **quiz** il devra se **déplacer dans l'espace** pour choisir la bonne réponse.

## Arguments uniques

- En **apprendre plus sur les espèces** en montrant l'invisible
- Expérience **immersive** et **interactive**
- Permet d'**utiliser son corps pour jouer** au jeu
- **Éducatif** et **amusant**

# Exigence minimales de la plate-forme

## 1. Ordinateur (Configuration requise pour le lecteur Unity)

Cette section énumère les exigences minimales pour construire et exécuter le lecteur Unity. Les performances réelles et la qualité du rendu peuvent varier en fonction de la complexité du projet.

Operating system	Windows	Universal Windows Platform	macOS	Linux
Operating system version	Windows 7 (SP1+) , Windows 10 and Windows 11	Windows 10, Xbox One, HoloLens	High Sierra 10.13+	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	x86, x64 architecture with SSE2 instruction set support.	x86, x64 architecture with SSE2 instruction set support, ARM, ARM64.	Apple Silicon, x64 architecture with SSE2.	x64 architecture with SSE2 instruction set support.
Graphics API	DX10, DX11, DX12 capable.	DX10, DX11, DX12 capable GPUs.	Metal capable Intel and AMD GPUs	OpenGL 3.2+, Vulkan capable.
Additional requirements	Hardware vendor officially supported drivers. For development: IL2CPP scripting back-end requires Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Hardware vendor officially supported drivers. For development: Windows 10 (64-bit), Visual Studio 2015 with C++ Tools component or later and Windows 10 SDK.	Apple officially supported drivers. For development: IL2CPP scripting back-end requires Xcode. Targeting Apple Silicon with IL2CPP scripting back-end requires macOS Catalina 10.15.4 and Xcode 12.2 or newer.	Gnome desktop environment running on top of X11 windowing system Other configuration and user environment as provided stock with the supported distribution (such as Kernel or Compositor) Nvidia and AMD GPUs using Nvidia official proprietary graphics driver or AMD Mesa graphics driver.
For all operating systems, the Unity Player is supported on workstations, laptop or tablet form factors, running without emulation, container or compatibility layer.				

## 2. Configuration requise pour l'éditeur Unity

Il vous faudra télécharger le logiciel Unity et se créer un compte pour pouvoir modifier le projet. Cette section énumère la configuration minimale requise pour faire fonctionner l'éditeur Unity.

Minimum requirements	Windows	macOS	Linux (Support in Preview)
Operating system version	Windows 7 (SP1+), Windows 10 and Windows 11, 64-bit versions only.	High Sierra 10.13+	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support
Graphics API	DX10, DX11, and DX12-capable GPUs	Metal-capable Intel and AMD GPUs	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.
Additional requirements	Hardware vendor officially supported drivers	Apple officially supported drivers	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.)
For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer.			

### 3. Kinect

#### Installer le kit de développement SDK :

Le **kit de développement logiciel** (SDK) Kinect pour Windows 2.0 permet aux développeurs de **créer des applications** qui prennent en charge la **reconnaissance gestuelle et vocale**, en utilisant la technologie du **capteur Kinect** sur des ordinateurs fonctionnant sous Windows 8, Windows 8.1 et Windows Embedded Standard 8. Il vous permettra par la même occasion de vérifier si la Kinect est bien branchée à l'ordinateur.

<https://www.microsoft.com/en-us/download/details.aspx?id=44561>

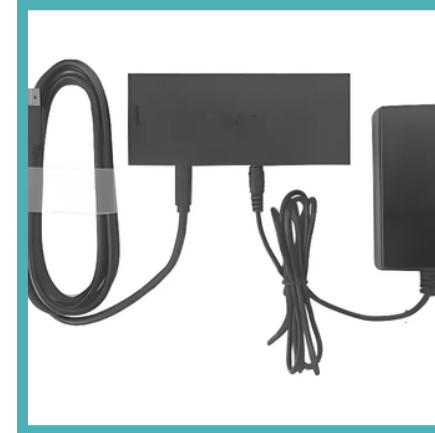
#### Télécharger les packages Unity Pro :

Installer le **kit de développement logiciel Kinect 2.0** et utiliser les **packages NuGet** pour **optimiser l'expérience** de développement .NET Framework. Vous pouvez également vous servir des **packages Unity Pro** pour **créer des applications Unity basées sur Kinect**.

<https://learn.microsoft.com/fr-fr/windows/apps/design/devices/kinect-for-windows>

### 3. Kinect

En plus de la **kinect V2**, il vous faudra un **adapteur avec une alimentation** pour pouvoir le **brancher sur votre ordinateur**.



### 4. Vidéo projecteur

**Tous types de vidéo projecteur** conviendra à notre jeu tant qu'il y a un **branchement avec une prise HDMI** possible, pour faire le **raccordement avec l'ordinateur**.

## Objectif du jeu

L'**objectif du jeu** est de faire **découvrir aux visiteurs des informations détaillées sur les espèces marines**. De manière **ludique**, il pourront **choisir les espèces** dont ils veulent **connaître les informations** et les **découvrir** sous forme de **quiz**.

## Règles du jeu

Le visiteur se retrouve **face à un écran**, où plusieurs espèces de poisson se déplacent sur un fond marin. Il doit, à l'aide de sa main, **choisir et attraper le poisson** pour lequel ils souhaitent être interrogé. **Un quiz apparaît** alors sur l'écran, avec des **questions** liées au poisson choisi précédemment et il a le choix entre **deux réponses**. Le visiteur se **déplace** à droite ou à gauche afin de choisir la réponse qui lui plaît.

## Structure du jeu



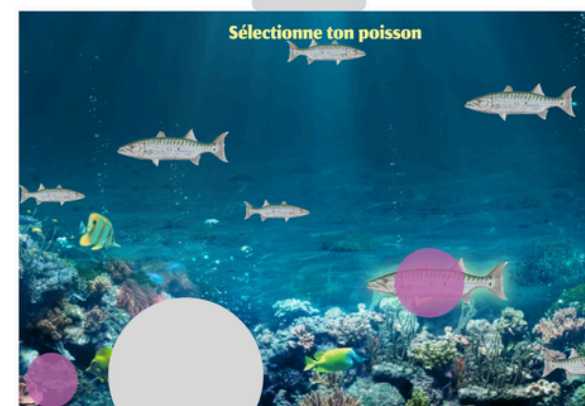


# Gameplay

Toutes **les espèces de poissons** se **déplacent de droite à gauche**. Le **curseur** de la main est **bougé grâce au visiteur** et dès qu'il arrive sur un poisson, on passe à une autre scène. **Le quiz** apparaît avec les **questions associées au poisson** choisis, la **caméra montre toutes les personnes présentes** dans la **zone de la Kinect**. **Deux réponses** apparaissent en même temps que la question et le visiteur a **10 secondes pour pouvoir y répondre**. Afin de répondre correctement à la question, le visiteur doit se trouver dans la bonne zone. Que ce soit une mauvaise ou une bonne réponse, un écran apparaît pour **représenter l'invisible du poisson avec une simple description** en lien avec la question du poisson. Seul le texte "mauvaise réponse et bonne réponse" va changer être modifié.

## 1. Contrôle du jeu

Pour certaines scènes (écran de démarrage / veille, scène de sélection de poisson), le joueur **utilise sa main**, représentée par un **curseur en forme de main** afin d'être plus **intuitif** et d'interagir avec les boutons ou les poissons. Dans une autre partie du jeu, le joueur doit **utiliser son corps** et **se déplacer** afin de **répondre aux différentes questions**.



## 2. Caméra de jeu

Le jeu se fait en **2D**, donc la **caméra ne bouge pas**, elle est de face **montrant le profil des poissons**. Tandis que pour la **scène des questions**, la caméra filme l'**espace du joueur** permettant de **montrer où se trouve le joueur**.

### HUD (Head up display) :

Pour la **première phase** du jeu, où l'on **choisit son poisson**, les différents éléments qui composent la scène sont le **curseur en forme de main** qui **suit les mouvements** de celle du joueur et une **bulle en haut de l'écran** avec la phrase "Attrape un des poissons à l'aide de ta main !" pour faire **comprendre** au joueur ce qu'il doit faire.

Pour la **seconde phase**, avec les questions, le HUD est composé d'**une bulle en haut de l'écran** avec la **question** et un **compte à rebours** juste en dessous qui indique au joueur combien de temps il lui reste pour se déplacer. Il y a également en bas de l'écran **deux choix de réponses** à la question, et enfin une **barre verticale au centre** de l'écran qui couvre toute sa hauteur, et qui permet de délimiter la partie gauche et droite de l'écran. Lorsque la réponse s'affiche, on voit centré sur l'écran le message "Bonne Réponse !" ou bien "Mauvaise Réponse !", puis en dessous, un **paragraphe explicatif** donnant un complément d'information, pour l'aspect éducatif.

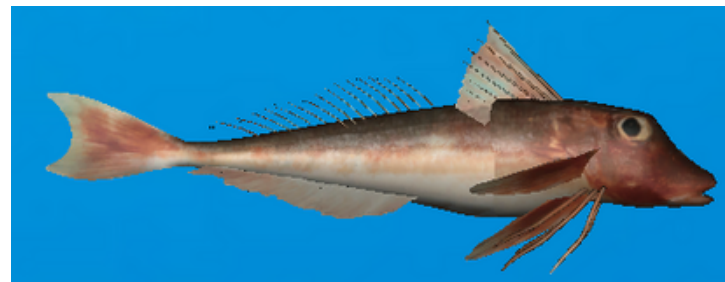
Entre chaque changement de scène il y a une **animation** qui prend tout l'écran et qui fait office de transition.

# Liste des personnages

## 1. Les poissons



Bar commun



Grondin Rouge



Dorade Royale



Sar Commun



Syngnathe Aiguille

# Liste de personnages

## 2. Spawn

Chaque poisson a son propre point de spawn défini à l'avance, donc pour chaque lancement de partie ils apparaîtront tous sur l'endroit leur étant prédéfini.

## 3. Réactions :

Les poissons ont une zone définie dans laquelle ils ne peuvent pas sortir, que ce soit à droite ou à gauche. Ils se déplacent de gauche à droite ou de droite à gauche, le changement de direction survient dès le moment où le poisson touche la zone limite du cadre imposé.

Dès qu'un poisson est touché avec le curseur, celui activera directement la prochaine scène, lui étant lié. Un délai peut être envisagé avant que l'activation ne se fasse, c'est pourquoi il faudra rester un certain nombre de temps sur le poisson avant de l'activer.

## Type de Quiz

Lorsque le joueur a sélectionné un poisson, il doit répondre à 3 questions concernant celui-ci. Les questions sont sous forme de QCM à 2 choix. L'utilisateur a 10 secondes pour y répondre, pour cela il doit se déplacer à la gauche ou la droite de l'écran. Pendant ce laps de temps la caméra de la Kinect est activé et l'utilisateur se voit se déplacer. À la fin du temps imparti, un paragraphe explicatif est montré au joueur.

### Exemples de questions :

- La bouche de la syngnathe aiguille est adaptée pour manger du plancton.
  - Choix 1 : **VRAI** - Choix 2 : FAUX
  - On dit qu'elle a une bouche suçoir, elle lui permet d'aspirer les planctons dont elle se nourrit.
- La syngnathe aiguille est de la même famille que :
  - Choix 1 : **Les hippocampes** - Choix 2 : Les anguilles
  - La syngnathe et les hippocampes font partie de la famille des "Syngnathidae".
- C'est le mâle qui porte les œufs.
  - Choix 1 : **VRAI** - Choix 2 : FAUX
  - le mâle garde les œufs dans sa poche ventrale pendant 5 semaines jusqu'à l'éclosion, et il peut incuber les œufs de différentes femelles.

# Configurations

Retrouvez les différents fichiers liés au projet dans le dossier “Aquarium V2”. Pour lancer le jeu, il suffira d’ouvrir le fichier Aquarium V2.exe dans le dossier “Build”.

## 1.Scripts

Les scripts sont rangés selon leur utilité, que ce soit en fonction des scènes ou de leur utilisation.

- **Dossier ChangeScene :**

- GameBehaviour : Permet de gérer le changement de scènes.
- SceneChanger : Permet de gérer le délai pour la scène de transition.

- **Dossier FishScene :**

- BodySourceViewHand : Permet d’activer et de contrôler la main avec la kinect.
- Fish : Permet de gérer le déplacement des poissons.
- Hand : Permet de gérer le comportement de la main avec les différents objets.

- **Dossier Kinect :**

N’est pas forcément utile actuellement dans le projet, mais permet de rajouter d’autres utilisations de la Kinect.

- **Dossier QuizScene :**

- **Dossier QuizGame :**

- **Dossier poissons :**

- BarQTS : Permet de charger sur Unity la base de données en fonction du fichier “BarQuestions.xml”.
      - BarQuestions.xml : Permet de modifier les questions, les réponses, etc.

- QtsData : Permet de charger le fichier “BarQts” par des encarts sous forme Unity (dans les composants).

- QuestionManager : Permet de gérer tout le gameplay du quiz avec les questions, les timers, la réaction des boutons, etc.

- XmlLoader : Permet de charger et de lire le fichier “BarQuestions.xml”.

- BodySourceManager : Permet de gérer les mouvements et l’activation de la kinect.

- BodySourceView : Permet d’activer et de contrôler la tête avec la kinect.

- ColorSourceManager : Permet de gérer le fonctionnement de la caméra de la Kinect.

- ColorSourceView : Permet d’afficher le rendu de la caméra de la Kinect.

- Head : Permet de gérer le comportement de la tête avec les différents objets.

- Timer : Permet de gérer le timer lié à la première question.

## 2. Dossiers

Pour procéder à des changements, il faudra utiliser le logiciel Unity mais pour accéder au dossier complet, il faudra aller dans le dossier “Assets”. Le jeu se trouve exclusivement dans le dossier “kinect”, les autres n’auront pas besoin d’être modifié sauf pour “TextMeshPro” afin d’ajouter des polices d’écriture.

- **Dossier Images** : Permet de retrouver les images présentent dans le jeu.
- **Dossier Materials** : Permet d’avoir les différentes textures utilisées dans le projet.
- **Dossier Prefabs** : Permet de retrouver les différents objets récurrent dès scènes, comme la main ou les poissons.
- **Dossier Scenes** : Permet de naviguer entre les différentes scène présentent dans le jeu.
- **Dossier Scripts** : Permet de modifier certains comportements lié au gameplay du jeu, des poissons ou de la kinect.
- **Dossier Videos** : Permet de retrouver les différentes vidéos du jeu.