

# Programmer Certification

## Objectifs

- Lire et comprendre les bilans
- Lire et comprendre les comptes de résultat comparatifs
- Lire et comprendre les rapports budgétaires et réels

## Prérequis

- Maîtrise des bases de mathématique financière
- Compréhension et expression écrite et orale en anglais

## Programme

### 1. Débogage, résolution de problèmes et interprétation de l'API

- 1.1. Étant donné un exemple de message de débogage, créez le code qui a créé le message de débogage. message de journal.
  - 1.2. Étant donné un extrait de code et le(s) message(s) d'erreur associé(s), déterminer quel(s) objet(s) est (sont) nul(s).
  - 1.3. Compte tenu d'une tâche de programmation spécifique nécessitant l'utilisation d'une classe particulière de l'API, déterminer la méthode et/ou les propriétés, les arguments, ou l'objet approprié. l'API, déterminer la méthode appropriée et/ou les propriétés, arguments ou autres syntaxes à utiliser.

### 2. Création de code

- 2.1. Indiquer quand et comment initialiser et utiliser les variables, y compris, mais sans s'y limiter l'utilisation appropriée de tous les modificateurs de variables et des collections de données telles que les tableaux, les listes et les dictionnaires. les tableaux, les listes et les dictionnaires.
  - 2.2. A partir d'une liste de mots-clés et d'éléments de syntaxe, construire une déclaration de fonction viable.
  - 2.3. A partir d'un extrait de code et d'une description du résultat souhaité, identifier la fonction appropriée pour contrôler ou déclencher un état, y compris, mais sans s'y limiter, les éléments suivants fonction appropriée pour contrôler ou déclencher un état, y compris mais non limité à le contrôleur de l'animateur.
  - 2.4. Étant donné un scénario où un type spécifique d'entrée est requis et où les blocs de construction nécessaires sont fournis, construire l'écoute d'entrée nécessaire de construction nécessaires sont fournis, construire l'écouteur d'entrée nécessaire, y compris notamment, mais pas exclusivement, le clavier et les entrées tactiles.
  - 2.5. Démontrer quand et/ou comment utiliser les différents opérateurs logiques et de contrôle de flux utilisés en C# et Unity. logiques et de contrôle de flux utilisés en C# et Unity.

- 2.6. Identifier, à partir d'un scénario, les actions appropriées à entreprendre lorsqu'un élément de l'interface utilisateur signale un changement.

### 3. Évaluation du code

- 3.1. Compte tenu d'un scénario concernant la nécessité de gérer une fonction d'événement, déterminez l'action appropriée à prendre, y compris le clavier et les touches tactiles.  
l'action appropriée à prendre, y compris, mais sans s'y limiter, le clavier et la saisie tactile. et de la saisie tactile.
  - 3.2. Etant donné un extrait de code qui produit une erreur à cause d'une variable dont le type de données est déclaré de façon incorrecte, identifiez l'erreur. est déclarée de façon incorrecte, identifiez l'erreur.
  - 3.3. Dans un extrait de code qui produit une erreur à cause d'une fonction ou d'une variable mal déclarée ou mal utilisée (public/private mismatch), identifier l'erreur.déclarée ou utilisée de manière incorrecte (incompatibilité public/privé), identifiez l'erreur. y compris, mais sans s'y limiter, l'utilisation d'événements d'animation.
  - 3.4. Etant donné un clip de code contenant une définition de classe, distinguer si la classe est une classe ECS ou une autre classe.est une classe ECS ou un autre type de classe.
  - 3.5. Etant donné un ensemble de clips de code, reconnaître le clip qui utilise des conventions de nommage qui respectent les normes de dénomination de l'Unité.
  - 3.6. Etant donné un extrait de code (ou un ensemble d'extraits de code), reconnaître les commentaires qui décrivent avec précision ce que fait le code.

### 4. Naviguer dans l'interface

- 4.1. Décrire le but, les caractéristiques et les fonctions des différentes fenêtres de l'IDE Unity.
  - 4.2. Démontrer comment changer l'IDE de script par défaut.
  - 4.3. A partir d'un scénario comprenant les éléments suivants, créer une machine d'état fonctionnelle a une partie limitée d'un scénario de jeu
    - b. un ensemble de clips d'animation
    - c. une liste de paramètres de propriété
  - 4.4. Créer et programmer une machine à états fonctionnels dans le contrôleur Unity Animator  
Controller, y compris, mais sans s'y limiter, l'utilisation de la syntaxe des fonctions Animator.