**Une image contenant Graphique, dessin, dessin humoristique, clipart

Description générée automatiquement**

**6 Informatique – 2023 / 2024**

**GREENGAME**

**Berte Florian**

Mr Cordeel

Mme Bouteglifine

Programmation

Multimédia

SE-Logiciels

**Table des matières**

**Introduction …………………………………………………………………. Page 4**

**Présentation de GreenGame …………………………………………… Page 5**

**Besoins du Projet ………………………………………………………. Page 5 - 6**

**Logiciel - Système Réseau …………………..……………………… Page 7 -18**

**Programmation - Développement du Jeu de Quiz …………… Page 19 - 21**

**Multimédia - Modélisation 3D ……………………………………… Page 22 – 24**

**Conclusion …………………………………………………………………… Page 25**

**Sources ……………………………………………………………………….. Page 26**

**Introduction**

**GreenGame** est une entreprise innovante spécialisée dans la création de jeux vidéo éducatifs. Notre objectif est de sensibiliser le public aux technologies vertes à travers des jeux interactifs et divertissants. Notre prochain projet se concentre sur le développement d'un jeu de quiz sur les véhicules écologiques, avec une intégration spécifique d'un modèle 3D d'une voiture volante à hydrogène.

**Présentation de GreenGame**

**GreenGame** est composée d'une équipe multidisciplinaire de développeurs, d'ingénieurs réseau, de designers 3D, et d'experts en technologies écologiques. Notre mission est de promouvoir les innovations écologiques par le biais de jeux vidéo engageants et informatifs.

**Besoins du Projet**

Pour mener à bien ce projet, **GreenGame** a identifié plusieurs besoins critiques :

**1. Logiciel** **- Système Réseau**

**Infrastructure**

* **Serveurs** : Déploiement de serveurs pour héberger le jeu et les données des utilisateurs.
* **Réseau interne** : Mise en place d'un réseau pour les employés, incluant des postes de travail, un serveur de fichiers.

**Sécurité**

* Implémentation de solutions de tolérance aux pannes pour assurer la continuité du service.
* Utilisation de redondances matérielles et logicielles pour permettre à d'autres systèmes de prendre la relève en cas de panne.

**2.** **Programmation - Développement du Jeu de Quiz**

**Langage et Outils**

* **Langage de Programmation** : Utilisation de Python avec tkinter pour le développement graphique du jeu.
* **Frameworks et Bibliothèques** : Pygame pour les aspects de son, Open-Cv pour la 3d et Firebase pour la base de donnée.

**Fonctionnalités**

* **Quiz Interactif** : Développement d'un jeu de quiz avec des questions sur les véhicules écologiques.
  + Questions à choix multiple.
  + Système de points et de progression.
* **Interface Utilisateur** : Création d'une interface utilisateur intuitive et attractive.

**3.** **Multimédia - Modélisation 3D**

**Outil de Modélisation**

* **Logiciel** : Utilisation de Blender pour la modélisation 3D.

**Modèle de Voiture Volante à Hydrogène**

* **Conception Réaliste** : Création d'un modèle 3D détaillé d'une voiture volante à hydrogène.
* **Intégration dans le Jeu** : Inclusion du modèle dans le jeu pour une immersion visuelle et pédagogique.
* **Animation et Rendu** : Animation du modèle pour des démonstrations interactives.

**Logiciel - Système Réseau**

**1. Introduction**

**1.1 Présentation de l'entreprise**

**GreenGame** est une entreprise de développement de jeux vidéo, dédiée à créer des expériences immersives et innovantes pour les joueurs du monde entier sur les véhicule écologique. Afin d'améliorer notre infrastructure réseau et garantir une connectivité fiable et sécurisée, nous souhaitons mettre en place une architecture réseau conforme aux standards modernes.

**1.2 Objectifs**

* Assurer une communication rapide et sécurisée entre les différents départements.
* Mettre en place une infrastructure réseau robuste et évolutive.
* Faciliter la gestion des ressources informatiques et des données.
* Assurer une haute disponibilité et une redondance des services critiques.

**2. Architecture Réseau**

**2.1 Description des Composants**

* **VLANs:**
  + Développement (VLAN 10): Réseau pour les équipes de développement.
  + Marketing (VLAN 20): Réseau pour l'équipe marketing.
* **Routeurs (2911):**
  + Routeur 1 et Routeur 2 pour la redondance et la gestion du trafic entre VLANs.
* **Switches (2960-24TT):**
  + Switch Marketing Développement:
    - Ports 1-10 pour VLAN 10 (Développement)
    - Ports 11-20 pour VLAN 20 (Marketing)
  + Switch Administration: Connecte les équipements du département Administration.
  + Switch Server: Connecte les serveurs.
* **PC et Serveurs:**
  + Différents postes de travail et serveurs (DNS, DHCP, Web) connectés aux switches respectifs.

**2.3 Câblage**

* **Entre PC/Serveurs et Switches:**
  + Utilisation de câbles Ethernet Cat 6 pour assurer une connectivité rapide et fiable.
  + Chaque PC ou serveur est connecté à un port approprié sur son switch désigné.
* **Entre Switches et Routeurs:**
  + Utilisation de câbles Ethernet Cat 6 ou de câbles en fibre optique pour les connexions de haute performance et longue distance.
  + Les switches sont connectés aux routeurs via des liaisons montantes (uplink).

**2.4 Topologie Réseau**

La topologie du réseau de **GreenGame** est une **topologie en étoile étendue (Extended Star Topology)** combinée à une **topologie client-serveur**.

* **Topologie en étoile étendue:**
  + Les postes de travail et les serveurs sont connectés à des switches, qui eux-mêmes sont interconnectés et reliés aux routeurs.
  + Cette topologie centralise la gestion du réseau tout en offrant de la redondance via des routes multiples entre les switches et les routeurs.
* **Topologie client-serveur:**
  + Les différents PC des départements (Développement, Marketing, Administration) agissent comme des clients se connectant aux serveurs pour accéder aux ressources partagées telles que les bases de données, les services web, le DNS et le DHCP.
  + Cette structure est idéale pour notre environnement, où les utilisateurs nécessitent un accès centralisé et contrôlé aux services critiques.

**3. Spécifications Techniques**

**3.1 Adressage IP et Passerelles**

1. **Développement (VLAN 10)**:
   * Adressage IP: 192.168.10.2-3/24
   * Passerelle: 192.168.10.1
   * DNS: 192.168.40.4
2. **Marketing (VLAN 20)**:
   * Adressage IP: 192.168.20. 2-3/24
   * Passerelle: 192.168.20.1
   * DNS: 192.168.40.4
3. **Administration:**
   * Adressage IP: 192.168.30.2/24
   * Passerelle: 192.168.30.1
   * DNS: 192.168.40.4
4. **Serveur**:
   * Adressage IP: 192.168.40.3-5/24
   * Passerelle: 192.168.40.1
   * DNS: 192.168.40.4

**3.2 Services Réseau**

* **DNS**: Serveur DNS pour la résolution de noms de domaine internes.
* **DHCP**: Serveur DHCP pour l'attribution automatique des adresses IP.
* **Web**: Serveur web pour les applications internes et les outils de développement.

**4. Schéma explicatif**

**Une image contenant texte, diagramme, ligne, capture d’écran

Description générée automatiquement**

**5. Configuration et Gestion du Réseau**

**Routeur 1**

Router-1 Router>enable

Router-1 Router#configure terminal

Ces commandes permettent d'accéder à la configuration d'un routeur Cisco. enable passe du mode utilisateur au mode privilégié, et configure terminal permet d'entrer en mode de configuration global pour modifier les paramètres du routeur.

Router-1 Router(config)#interface GigabitEthernet 0/0.10

Router-1 Router(config-subif)#encapsulation dot1Q 10

Router-1 Router(config-subif)#ip address 192.168.10.1 255.255.255.0

Router-1 Router(config-subif)#standby 10 ip 192.168.10.1

Router-1 Router(config-subif)#standby 10 priority 110

Router-1 Router(config-subif)#standby 10 preempt

Router 1 Router(config-subif)#ip helper-address 192.168.40.5

Router-1 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/0.10 sur un routeur Cisco pour le VLAN 10 avec une adresse IP de 192.168.10.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.10.1) avec une priorité de 110 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-1 Router(config)#interface GigabitEthernet 0/0.20

Router-1 Router(config-subif)#encapsulation dot1Q 20

Router-1 Router(config-subif)#ip address 192.168.20.1 255.255.255.0

Router-1 Router(config-subif)#standby 20 ip 192.168.20.1

Router-1 Router(config-subif)#standby 20 priority 110

Router-1 Router(config-subif)#standby 20 preempt

Router 1 Router(config-subif)#ip helper-address 192.168.40.5

Router-1 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/0.20 sur un routeur Cisco pour le VLAN 20 avec une adresse IP de 192.168.20.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.20.1) avec une priorité de 110 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-1 Router(config)#interface GigabitEthernet 0/1

Router-1 Router(config-subif)#encapsulation dot1Q 30

Router-1 Router(config-subif)#ip address 192.168.30.1 255.255.255.0

Router-1 Router(config-subif)#standby 30 ip 192.168.30.1

Router-1 Router(config-subif)#standby 30 priority 110

Router-1 Router(config-subif)#standby 30 preempt

Router 1 Router(config-subif)#ip helper-address 192.168.40.5

Router-1 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/1 sur un routeur avec une adresse IP de 192.168.30.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.30.1) avec une priorité de 110 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-1 Router(config)#interface GigabitEthernet 0/2

Router-1 Router(config-subif)#encapsulation dot1Q 40

Router-1 Router(config-subif)#ip address 192.168.40.1 255.255.255.0

Router-1 Router(config-subif)#standby 40 ip 192.168.40.1

Router-1 Router(config-subif)#standby 40 priority 110

Router-1 Router(config-subif)#standby 40 preempt

Router 1 Router(config-subif)#ip helper-address 192.168.40.5

Router-1 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/2 sur un routeur avec une adresse IP de 192.168.40.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.40.1) avec une priorité de 110 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-1 Router(config-if)#end

Router-1 Router#show standby

Router-1 Router#write memory

Ces commandes finalisent la configuration et sauvegardent les paramètres d'un routeur Cisco. end quitte le mode de configuration pour revenir au mode privilégié. show standby affiche l'état du protocole HSRP. write memory sauvegarde la configuration en cours dans la mémoire non volatile, assurant ainsi qu'elle sera conservée après un redémarrage du routeur.

**Routeur 2**

Router-2 Router>enable

Router-2 Router#configure terminal

Ces commandes permettent d'accéder à la configuration d'un routeur Cisco. enable passe du mode utilisateur au mode privilégié, et configure terminal permet d'entrer en mode de configuration global pour modifier les paramètres du routeur.

Router-2 Router(config)#interface GigabitEthernet 0/0.10

Router-2 Router(config-subif)#encapsulation dot1Q 10

Router-2 Router(config-subif)#ip address 192.168.10.1 255.255.255.0

Router-2 Router(config-subif)#standby 10 ip 192.168.10.1

Router-2 Router(config-subif)#standby 10 priority 100

Router-2 Router(config-subif)#standby 10 preempt

Router-2 Router(config-subif)#ip helper-address 192.168.40.5

Router-2 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/0.10 sur un routeur Cisco pour le VLAN 10 avec une adresse IP de 192.168.10.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.10.1) avec une priorité de 100 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-2 Router(config)#interface GigabitEthernet 0/0.20

Router-2 Router(config-subif)#encapsulation dot1Q 20

Router-2 Router(config-subif)#ip address 192.168.20.1 255.255.255.0

Router-2 Router(config-subif)#standby 20 ip 192.168.20.1

Router-2 Router(config-subif)#standby 20 priority 100

Router-2 Router(config-subif)#standby 20 preempt

Router-2 Router(config-subif)#ip helper-address 192.168.40.5

Router-2 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/0.20 sur un routeur Cisco pour le VLAN 20 avec une adresse IP de 192.168.20.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.20.1) avec une priorité de 10 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-2 Router(config)#interface GigabitEthernet 0/1

Router-2 Router(config-subif)#encapsulation dot1Q 30

Router-2 Router(config-subif)#ip address 192.168.30.1 255.255.255.0

Router-2 Router(config-subif)#standby 30 ip 192.168.30.1

Router-2 Router(config-subif)#standby 30 priority 100

Router-2 Router(config-subif)#standby 30 preempt

Router-2 Router(config-subif)#ip helper-address 192.168.40.5

Router-2 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/1 sur un routeur avec une adresse IP de 192.168.30.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité (fiabilité) en définissant une adresse IP virtuelle (192.168.30.1) avec une priorité de 100 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-2 Router(config)#interface GigabitEthernet 0/2

Router-2 Router(config-subif)#encapsulation dot1Q 40

Router-2 Router(config-subif)#ip address 192.168.40.1 255.255.255.0

Router-2 Router(config-subif)#standby 40 ip 192.168.40.1

Router-2 Router(config-subif)#standby 40 priority 100

Router-2 Router(config-subif)#standby 40 preempt

Router-2 Router(config-subif)#ip helper-address 192.168.40.5

Router-2 Router(config-subif)#no shutdown

Ce code configure une sous-interface GigabitEthernet0/2 sur un routeur avec une adresse IP de 192.168.40.1/24. Il utilise l'encapsulation et configure HSRP pour la haute disponibilité en définissant (fiabilité) une adresse IP virtuelle (192.168.40.1) avec une priorité de 100 et la préemption activée, ce qui permet au routeur de devenir actif si sa priorité est la plus élevée. De plus, il configure un relais DHCP en spécifiant un serveur DHCP à l'adresse 192.168.40.5, et active finalement la sous-interface avec la commande no shutdown.

Router-2 Router(config-if)#end

Router-2 Router#show standby

Router-2 Router#write memory

Ces commandes finalisent la configuration et sauvegardent les paramètres d'un routeur Cisco. end quitte le mode de configuration pour revenir au mode privilégié. show standby affiche l'état du protocole HSRP. write memory sauvegarde la configuration en cours dans la mémoire non volatile, assurant ainsi qu'elle sera conservée après un redémarrage du routeur.

**Switch-Marketing-Développeur**

Switch-Marketing-Développeur Switch>enable

Switch-Marketing-Développeur Switch#configure terminal

Ces commandes permettent d'accéder à la configuration d'un routeur Cisco. enable passe du mode utilisateur au mode privilégié, et configure terminal permet d'entrer en mode de configuration global pour modifier les paramètres du routeur.

Switch-Marketing-Développeur Switch(config)#vlan 20

Switch-Marketing-Développeur Switch(config-vlan)#name Marketing

Switch-Marketing-Développeur Switch(config-vlan)#exit

Switch-Marketing-Développeur Switch(config)#vlan 10

Switch-Marketing-Développeur Switch(config-vlan)#name Developpement

Switch-Marketing-Développeur Switch(config-vlan)#exit

Le code configure deux VLANs sur un switch Cisco. Il commence par entrer en mode de configuration globale, crée le VLAN 20 et lui donne le nom "Marketing". Ensuite, il quitte le mode de configuration VLAN. Le processus est répété pour le VLAN 10, qui est nommé "Développement". Chaque VLAN est ainsi configuré pour segmenter le réseau en groupes distincts, facilitant la gestion et la sécurité.

Switch-Marketing-Développeur Switch(config)#interface range fastEthernet 0/1-10

Switch-Marketing-Développeur Switch(config-if-range)#switchport mode access

Switch-Marketing-Développeur Switch(config-if-range)#switchport access vlan 10

Switch-Marketing-Développeur Switch(config-if-range)#exit

Le code configure les interfaces FastEthernet 0/1 à 0/10 d'un switch Cisco pour qu'elles soient en mode accès et les assigne au VLAN 10. Cela permet de s'assurer que tous les ports de cette plage sont membres du VLAN 10, facilitant la segmentation du réseau pour le groupe "Développement".

Switch-Marketing-Développeur Switch(config)#interface range fastEthernet 0/11-20

Switch-Marketing-Développeur Switch(config-if-range)#switchport mode access

Switch-Marketing-Développeur Switch(config-if-range)#switchport access vlan 20

Switch-Marketing-Développeur Switch(config-if-range)#exit

Le code configure les interfaces FastEthernet 0/11 à 0/20 d'un switch Cisco pour qu'elles soient en mode accès et les assigne au VLAN 20. Cela permet de s'assurer que tous les ports de cette plage sont membres du VLAN 20, facilitant la segmentation du réseau pour le groupe "Marketing".

Switch-Marketing-Développeur Switch(config)#interface fastethernet 0/21

Switch-Marketing-Développeur Switch(config-if)#no switchport mode trunk

Switch-Marketing-Développeur Switch(config-if)#switchport mode access

Switch-Marketing-Développeur Switch(config-if)#exit

Switch-Marketing-Développeur Switch(config)#interface fastethernet 0/22

Switch-Marketing-Développeur Switch(config-if)#no switchport mode trunk

Switch-Marketing-Développeur Switch(config-if)#switchport mode access

Switch-Marketing-Développeur Switch(config-if)#exit

Le code configure l'interface FastEthernet 0/21 et FastEthernet 0/22 d'un switch Cisco pour passer du mode trunk au mode accès, assurant ainsi que ce port est utilisé pour se connecter à un seul VLAN. Cela simplifie la gestion du réseau en désactivant le mode trunk et en activant le mode accès pour cette interface spécifique.

Switch-Marketing-Développeur Switch(config)#interface GigabitEthernet 0/1

Switch-Marketing-Développeur Switch(config-if)#switchport mode trunk

Switch-Marketing-Développeur Switch(config-if)#switchport trunk allowed vlan 10,20

Switch-Marketing-Développeur Switch(config-if)#no shutdown

Switch-Marketing-Développeur Switch(config-if)#exit

Switch-Marketing-Développeur Switch(config)#interface GigabitEthernet 0/2

Switch-Marketing-Développeur Switch(config-if)#switchport mode trunk

Switch-Marketing-Développeur Switch(config-if)#switchport trunk allowed vlan 10,20

Switch-Marketing-Développeur Switch(config-if)#no shutdown

Switch-Marketing-Développeur Switch(config-if)#exit

Le code configure l'interface FastEthernet 0/21 et FastEthernet 0/22 d'un switch Cisco pour passer du mode trunk au mode accès, assurant ainsi que ce port est utilisé pour se connecter à un seul VLAN. Cela simplifie la gestion du réseau en désactivant le mode trunk et en activant le mode accès pour cette interface spécifique.

Switch-Marketing-Développeur Switch(config-if)#exit

Switch-Marketing-Développeur Switch(config)#exit

Switch-Marketing-Développeur Switch#write memory

Le code quitte le mode de configuration de l'interface et le mode de configuration globale, puis sauvegarde la configuration actuelle du switch dans la mémoire non volatile (NVRAM). Cela garantit que les modifications apportées, telles que les configurations de VLAN et d'interface, sont préservées même après un redémarrage du switch.

**Switch-Administration**

Switch-Administration Switch>enable

Switch-Administration Switch#configure terminal

Switch-Administration Switch(config)#interface gigabitEthernet 0/1

Switch-Administration Switch(config-if)#switchport mode trunk

Switch-Administration Switch(config-if)#interface gigabitEthernet 0/2

Switch-Administration Switch(config-if)#switchport mode trunk

Switch-Administration Switch(config-if)#exit

Switch-Administration Switch(config)#exit

Switch-Administration Switch#write memory

Ce code configure un switch Cisco en activant le mode trunk sur deux interfaces. La commande enable passe du mode utilisateur au mode privilégié, puis configure terminal permet d'entrer en mode de configuration global. Les interfaces gigabitEthernet 0/1 et gigabitEthernet 0/2 sont chacune configurées en mode trunk via la commande switchport mode trunk, ce qui permet à ces ports de transporter plusieurs VLAN. Après avoir terminé la configuration des interfaces, les commandes exit permettent de revenir au mode de commande précédent, et write memory sauvegarde la configuration dans la mémoire non volatile pour qu'elle soit conservée après un redémarrage.

**Switch-Server**

Switch- Server Switch>enable

Switch- Server Switch#configure terminal

Switch- Server Switch(config)#interface gigabitEthernet 0/1

Switch- Server Switch(config-if)#switchport mode trunk

Switch- Server Switch(config-if)#interface gigabitEthernet 0/2

Switch- Server Switch(config-if)#switchport mode trunk

Switch- Server Switch(config-if)#exit

Switch- Server Switch(config)#exit

Switch- Server Switch#write memory

Ce code configure un switch Cisco en activant le mode trunk sur deux interfaces. La commande enable passe du mode utilisateur au mode privilégié, puis configure terminal permet d'entrer en mode de configuration global. Les interfaces gigabitEthernet 0/1 et gigabitEthernet 0/2 sont chacune configurées en mode trunk via la commande switchport mode trunk, ce qui permet à ces ports de transporter plusieurs VLAN. Après avoir terminé la configuration des interfaces, les commandes exit permettent de revenir au mode de commande précédent, et write memory sauvegarde la configuration dans la mémoire non volatile pour qu'elle soit conservée après un redémarrage.

**6. Conclusion**

La mise en place de cette architecture réseau permettra à GreenGame de bénéficier d'une infrastructure solide, évolutive et sécurisée, soutenant ainsi la croissance et les ambitions de l'entreprise dans le développement de jeux vidéo dans les thème des véhicule écologique.

**Programmation - Développement du Jeu de Quiz**

**1. Introduction**

Pour notre premier jeu, GreenGame développe un jeu de quiz permettant de répondre à des questions sur les véhicules écologiques. Il pourra être joué seul ou en duo, offrant une expérience éducative et divertissante.

**2. Structure du Programme**

* 1. **Démarrage du Jeu :** Une animation d'un modèle 3D marque le début du jeu.
  2. **Principal :** Un menu permettant de choisir entre les modes de jeu Solo et Duo.
  3. **Jeu de Quiz :** Divers types de jeux de quiz seront disponibles :
* Trouver la bonne réponse parmi 4 choix.
* Vrai ou faux.
* Retrouver le son.
* Replacer sur la carte du monde.
* Replacer dans l'ordre chronologique.
* Replacer dans la bonne catégorie.
  1. **Scores des Joueurs :** Affichage des scores des joueurs pour suivre la progression et la performance.

**3. Bibliothèques Utilisées**

* **Tkinter** : Pour l'interface graphique.
* **OpenCV** : Pour l'animation 3D.
* **Pygame** : Pour la gestion des sons.
* **Firebase** : Pour la base de données.
* **JSON** : Pour le stockage des informations de l'application localement.

**4. Algorithmes de Jeu : Modes Solo et Duo**

* 1. **Modes Solo**

1. **Préparation du Jeu**
   * Le joueur choisit le nombre total de questions (5, 10, 20, etc.).
   * Sélection aléatoire des questions avec leur type de jeu correspondant.
   * Stockage des questions sélectionnées.
2. **Déroulement du Jeu**
   * Pour chaque question :
     + Affichage de la question et démarrage du chronomètre.
     + Attente de la réponse du joueur ou de la fin du chronomètre.
     + Vérification de la réponse :
       - Si correcte, ajout de points au score.
       - Si incorrecte, enregistrement de la question dans une liste des questions incorrectes.
     + Répétition pour le reste des questions.
3. **Fin du Jeu**
   * Affichage du score final.
   * Affichage des questions incorrectes, limité à 3 ou moins.
   1. **Modes Duo**
4. **Création et Rejoignement du Groupe**

* Le joueur hôte crée un groupe de jeu.
* Sélection aléatoire de 20 questions avec leur type de jeu correspondant, stockées dans la base de données.
* Attente de la connexion du joueur invité au groupe.

1. **Début de la Partie**
   * Le joueur invité rejoint le groupe et télécharge les questions avec leur ordre.
   * Le joueur invité indique que la partie a commencé à l'hôte.
2. **Déroulement du Jeu**
   * Pour chaque question :
     + Affichage de la question et démarrage du chronomètre.
     + Attente de la réponse du joueur ou de la fin du chronomètre.
     + Vérification de la réponse :
       - Si correcte, ajout de points au score du joueur et mise à jour dans la base de données.
     + Répétition pour le reste des questions.
3. **Fin de la Partie**
   * Attente de la fin de la partie pour les deux joueurs.
   * Affichage du score final pour chaque joueur.

**Multimédia - Modélisation 3D**

**1. Introduction**

Pour ce projet, nous avons choisi de créer un modèle 3D d'une voiture volante fonctionnant à l'hydrogène. Ce concept allie deux idées innovantes : la mobilité aérienne urbaine et l'énergie propre. Cette combinaison vise à résoudre les défis actuels de la congestion urbaine et de la pollution tout en proposant une vision futuriste et écologique des transports.

1. **Modélisation initiale :**

* **Base de Modélisation:** Commencez avec un cube comme base.
* **Forme Globale:** Extrudez pour créer la forme générale de la voiture.
* **Ajout de Modificateurs:** Appliquez les modificateurs de subdivision et de symétrie.
* **Détails Additionnels:** Ajoutez des détails avec des loop cuts et des formes supplémentaires.
* **Ajustements Fins:** Redimensionnez pour affiner les proportions.
* **Matériaux et Lumières:** Appliquez des matériaux et ajoutez des lumières pour finaliser la modélisation.

1. **Animation :**

* **Animation de vol:** Création de la séquence d'animation pour le décollage, le vol et l'atterrissage de la voiture volante avec la camera qui tourne au tour de la base.
* **Animation de conduite:** Animation de la voiture en mode terrestre avec la camera qui tourne au tour de la base.

1. **Rendu final :**

* **Rendu image par image:** Le rendu est réalisé image par image, avec un total de 425 images par animation.
* **Durée du rendu:** Chaque animation nécessite environ 12 heures de rendu pour obtenir des images de haute qualité avec le moteur de rendu Cycles dans Blender.

1. **Aperçu des rendus**

Une image contenant véhicule, roue, Véhicule terrestre, Véhicule de jouet

Description générée automatiquement

Une image contenant véhicule, roue, Véhicule terrestre, voiture

Description générée automatiquement

1. **Conclusion**

Ce projet de voiture volante à hydrogène représente une vision du futur où les transports sont non seulement rapides et efficaces mais aussi respectueux de l'environnement. En combinant les technologies de la mobilité aérienne et de l'hydrogène, nous pouvons imaginer un avenir plus propre et plus vert.

**Conclusion**

GreenGame s'engage à créer des jeux vidéo éducatifs de haute qualité, mettant en avant les technologies écologiques. Ce projet de quiz sur les véhicules écologiques, avec son système réseau robuste, ses mesures de tolérance aux pannes, et son modèle 3D détaillé, représente une étape importante vers notre objectif de sensibilisation et d'éducation à travers le jeu.

**Sources**

Aide sur l'écriture du document: Chat-GPT, https://chatgpt.com/

Inspiration pour le modèle 3D: https://www.domestika.org/fr/projects/433386-low-poly-car