

WADE Mame Diarra
NETO Anete
EL AMRANI Lina

SAÉ 304 : Découvrir et mettre en place un réseau IoT

User Story 4 : Suivi de la distance par rapport au diamant et annonce de gagnant

Objectif :

Lorsque les joueurs se déplacent dans le jeu, le but est qu'ils sachent en temps réel s'ils se rapprochent ou s'éloignent du diamant. Le jeu doit informer les deux joueurs de la progression de l'autre joueur vers le diamant, et lorsqu'un joueur est suffisamment proche du diamant pour y accéder, un message de victoire est affiché.

Explication détaillée du processus :

1. Placer le diamant au centre entre les deux joueurs :

- Au début du jeu, une position centrale est calculée en fonction des positions des deux joueurs (user1 et user2). Le diamant est ensuite placé à cet endroit.
- Cela garantit que les joueurs sont en compétition pour trouver le diamant placé de manière équitable.

2. Calculer la distance au diamant :

- Chaque joueur a sa position calculée à chaque itération de la boucle du jeu avec `mc.entity.getTilePos(player_id)`. La distance entre la position du joueur et le diamant est ensuite calculée en utilisant la formule de distance euclidienne :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

Fig 1 : Formule de distance euclidienne entre deux points dans un espace 3D

```
Player user1: Previous Distance = 5.830951894845301, Current Distance = 5.830951894845301
Player user2 position: Vec3(28,0,-19)
Player user2: Previous Distance = 5.830951894845301, Current Distance = 5.830951894845301
Player user1 position: Vec3(18,0,-13)
Player user1: Previous Distance = 5.830951894845301, Current Distance = 5.830951894845301
Player user2 position: Vec3(28,0,-19)
```

Fig 2 : Calcule de distance des deux joueurs sur commande

```
# Fonction pour calculer la distance entre un joueur et le diamant
def distance_to_diamond(player_id):
    player_pos = mc.entity.getTilePos(player_id)
    return math.sqrt((player_pos.x - diamond_x) ** 2 + (player_pos.y - diamond_y) ** 2 + (player_pos.z - diamond_z) ** 2)
```

Fig 3 : code python pour calculer la distance entre chaque joueur et diamant

3. Vérifier si le joueur se rapproche ou s'éloigne du diamant :

- Lorsqu'un joueur se déplace, la distance est comparée à celle mesurée lors du déplacement précédent. Si la distance actuelle est plus faible, cela signifie que le joueur se rapproche du diamant, et un message est envoyé aux joueurs via `mc.postToChat()`.
- Si la distance est plus grande, le joueur s'éloigne, et un message est également envoyé pour l'informer.
- Chaque joueur a son propre message de notification qui lui indique s'il se rapproche ou s'éloigne du diamant, renforçant ainsi l'interaction en temps réel entre les joueurs.

```
# Fonction pour vérifier la direction du joueur
def check_direction(player_id, prev_distance):
    curr_distance = distance_to_diamond(player_id)
    print(f"Player {player_names[player_id]}: Previous Distance = {prev_distance}, Current Distance = {curr_distance}")
    if curr_distance < prev_distance:
        mc.postToChat(f"{player_names[player_id]} is getting closer to the diamond!")
    elif curr_distance > prev_distance:
        mc.postToChat(f"{player_names[player_id]} is getting farther from the diamond!")
    return curr_distance
```

Fig 4 : code python pour vérifier la direction du joueur et envoyer un message



Fig 5 : messages de distance sur minecraft

4. Déclencher la victoire :

- Si un joueur se trouve exactement sur le diamant (en vérifiant ses coordonnées sur l'axe X, Y, Z), ce joueur est déclaré vainqueur.
- Le jeu s'arrête immédiatement, et un message de victoire est envoyé à tous les joueurs. Le diamant est alors supprimé du jeu, et un "mégapouvoir" est activé pour le gagnant via la fonction `mega_pouvoir()`.
- Le bloc du diamant est détruit après la fin de la partie.

```

# Annonce du gagnant
if winner_id:
    winner_name = player_names[winner_id]
    mc.postToChat(f"{winner_name} has won by standing on the diamond!")
    # Supprimer le diamant
    mc.setBlock(diamond_x, diamond_y, diamond_z, block.AIR)
    mega_pouvoir(mc, winner_id, player_names) # Activer le mega pouvoir
else:
    mc.postToChat("No one won the game.")

```

Fig 6 : code python annonce du gagnant

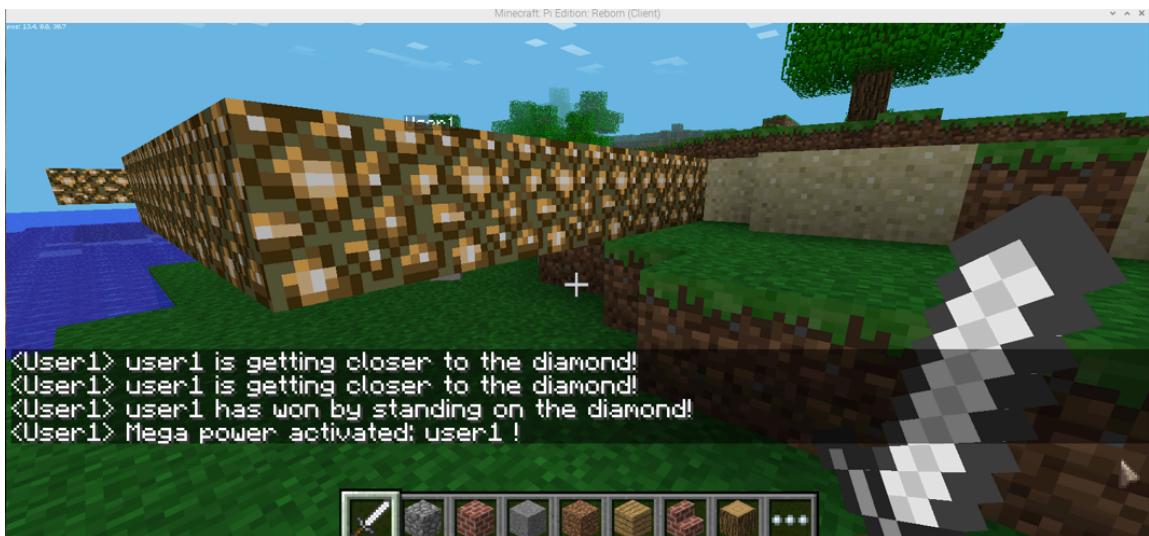


Fig 7 : annonce du gagnant et activation du mega pouvoir

5. Gestion des messages MQTT :

En parallèle de cette logique de distance et de victoire, des messages sont envoyés via MQTT en fonction des actions des joueurs sur leurs gants (par exemple, appuyer sur les boutons pour activer des super-pouvoirs ou des malus).