

Writing & Presentation Assignment Week 6

Online Game Introduction, Photon Unity Networking

pada sesi ini kami membahas tentang sebuah framework online game bernama Photon Unity Networking yang akan jauh mempermudah pembuatan game online. Hal yang dipelajari meliputi :

- Offline vs Online Game

Perbedaan offline game dan online game yaitu

Offline Game	Online Game
The code is much simpler to make	Almost always harder to make
No need extra `online` requirement	Need extra requirement (database server, matchmaking service etc)
No need for external network connection	Need at least some internet connection or even a constant connection to the internet
Development focus on gameplay / user facing side.	Need game backend engineer / server engineer to handle the online service and API

- Pengenalan Photon Unity Networking (PUN)

Photon Engine adalah sebuah Game Engine yang spesialisasinya adalah dalam membangun game Multiplayer. Engine ini menggabungkan berbagai perkembangan dalam komponen networking dan pengembangan software yang dapat membuat game berjalan dengan performa yang memadai untuk menangani lingkungan multiplayer.

Dengan jaringan yang aktif, melebihi 22 juta pengguna online per harinya, Photon Engine menunjukan dirinya sebagai platform yang handal dalam pengembangan sebuah game. Photon Engine mempunyai 4 jenis produk yang dapat digunakan untuk mengembangkan game Multiplayer kita, antara lain...

1. Photon Unity Networking (PUN)

Photon merupakan real-time multiplayer game development framework. Ini memungkinkan kita untuk membuat game tanpa memikirkan sisi server/backend pada sebuah game.

2. Realtime

Realtime adalah sebuah produk yang fungsi utamanya lebih membenarkan isu- isu game yang latensi rendah. Photon Realtime bekerja secara cross-platform yang artinya jaringan game kita dapat bermain dengan berbagai platform, yang PC dapat bermain dengan MacOS dan yang konsol-konsol juga dapat bermain bersama.

3. BOLT

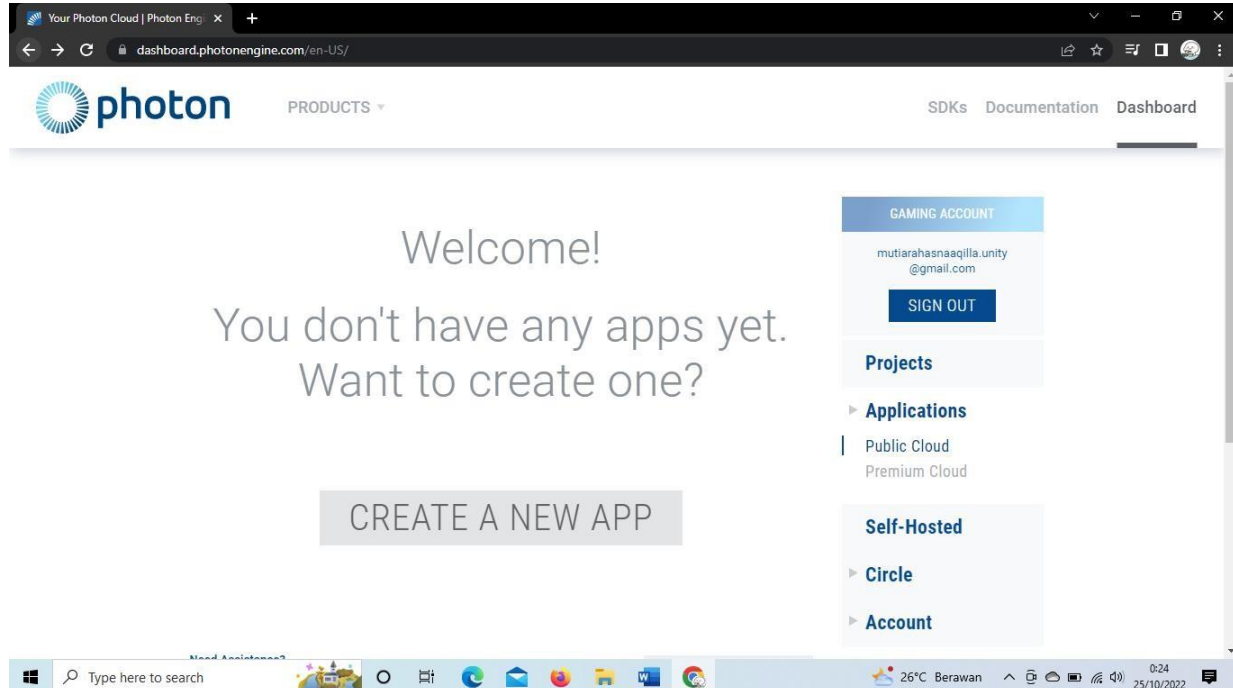
Produk BOLT dari Photon lebih mengedepankan genre seperti First Person Shooter, Third Person Shooter, Battle Royale dan Action Games. Jika dilihat dari berbagai genre tersebut, produk yang satu ini spesialisasinya lebih ke permainan yang membutuhkan kekuatan komputasi yang besar dan banyak untuk menopang berbagai event dinamis yang selalu terjadi dan berganti dalam lingkup permainan dan juga mengoptimasi performa lag serta aksi respons dari sisi klien.

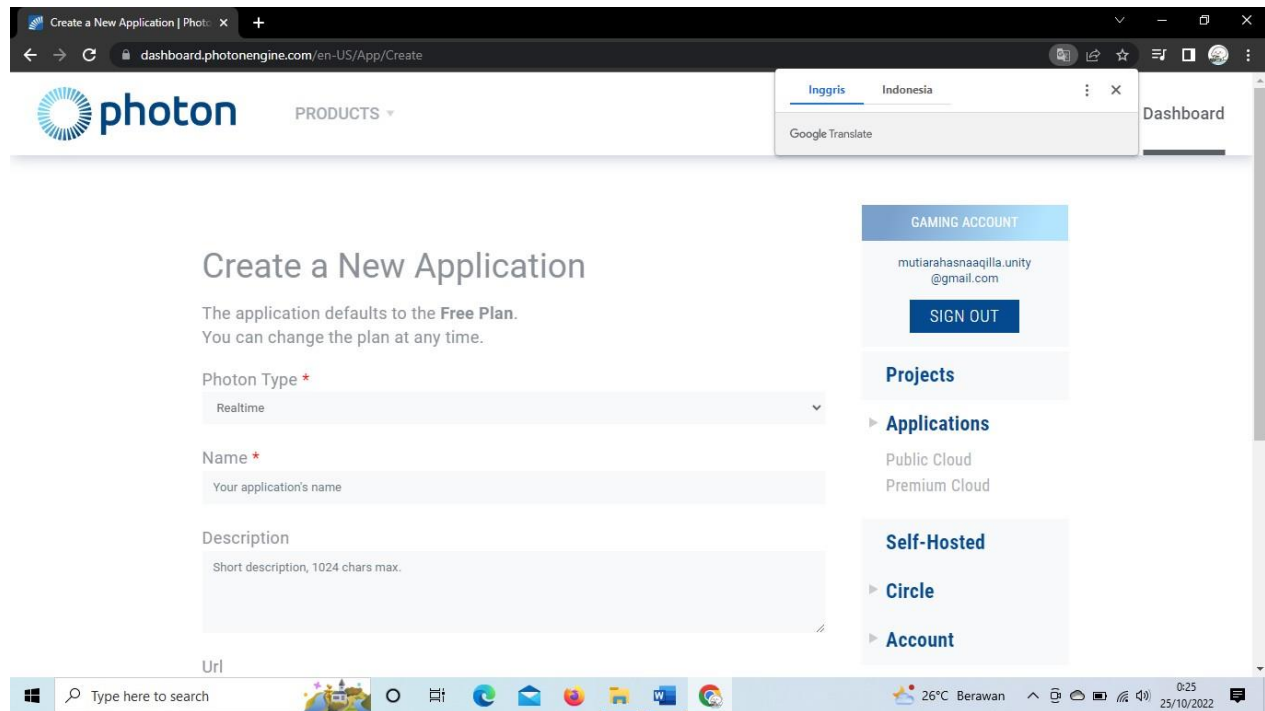
4. Quantum

Terakhir adalah produk Quantum, dan engine yang satu ini (mungkin) adalah salah satu produk terbaik yang dapat ditawarkan oleh Photon. Genre yang dipeluk seperti MOBA, Sports, RTS, Brawlers, FPS. Spesialisasinya adalah performa komputasi server yang sangat tinggi, sehingga hal-hal seperti pengerjaan fisika, matematika, Artificial Intelligence dan hal-hal lain yang biasanya perlu perhitungan ataupun sortir, Quantum akan mengerjakan semuanya dengan cepat, dan selain itu tidak terdapat Netcode yang artinya struktur kode dalam mode Single Player maupun Local/Online Multiplayer sama, tidak ada perbedaan antara mode game.

- Pembuatan akun photon

Untuk membuat PhotonPUN harus login dulu ke <https://id.photonengine.com/en-US/Account/SignUp> Di website tersebut hanya punya 1 dashboard, **create new app**





- Instalasi PUN ke unity project

Di unity sendiri harus menginstall PUN 2 dari <https://assetstore.unity.com/> jangan lupa untuk add to my asset



Introduction to Remote Procedure Call (PUNRPC)

Salah satu fitur yang membedakan PUN dari paket Photon lainnya adalah dukungan untuk **Remote Procedure Calls (RPC)**. RPC merupakan sebuah method panggilan pada klien di ruangan yang sama. Untuk mengaktifkan RPC untuk beberapa method kita perlu menerapkan **atribut [PunRPC]**.

```
[PunRPC]
void ChatMessage(string a, string b)
{
    Debug.Log(string.Format("ChatMessage {0} {1}", a, b));
}
```

Untuk memanggil fungsi yang ditandai sebagai RPC, kita memerlukan **PhotonView**. Contoh :

```
PhotonView photonView = PhotonView.Get(this);
photonView.RPC("ChatMessage", PhotonTargets.All, "jup", "and jup!");
```

PhotonView seperti "target" untuk RPC: Semua klien menjalankan metode hanya pada GameObject jaringan dengan PhotonView tertentu. Jika Anda mengenai objek tertentu dan memanggil RPC "ApplyDamage", maka klien penerima akan menerapkan kerusakan pada objek yang sama!

Kita dapat menambahkan beberapa parameter (asalkan PUN dapat membuat serialisasi). Ketika Anda melakukannya, metode dan panggilan harus memiliki parameter yang sama. Jika klien penerima tidak dapat menemukan metode yang cocok, itu akan mencatat kesalahan.

Ada satu pengecualian untuk aturan ini: Parameter terakhir dari metode RPC dapat berupa tipe PhotonMessageInfo, yang akan menyediakan konteks untuk setiap panggilan. Anda tidak mengatur PhotonMessageInfo dalam panggilan.

```
[PunRPC]
void ChatMessage(string a, string b, PhotonMessageInfo info)
{
    // the photonView.RPC() call is the same as without the info parameter.
    // the info.sender is the player who called the RPC.
    Debug.Log(string.Format("Info: {0} {1} {2}", info.sender, info.photonView, info.timestamp));
}
```

Dengan pengirim berada di PhotonMessageInfo dan "penargetan" melalui PhotonView, Anda dapat menerapkan pemotretan seseorang tanpa parameter tambahan. Anda tahu siapa yang menembak dan apa yang dipukul.

Untuk mencegah kehilangan RPC, klien dapat berhenti mengeksekusi pesan masuk (inilah yang dilakukan LoadLevel untuk kita). Saat kita mendapatkan RPC untuk memuat beberapa adegan, segera setel "isMessageQueueRunning = false" hingga konten diinisialisasi. Menonaktifkan antrian pesan akan menunda pesan masuk dan keluar sampai antrian dibuka. Jelas, sangat penting untuk membuka kunci antrian saat Anda siap untuk melanjutkan. Contoh :

```
private IEnumerator MoveToGameScene()
{
    // Temporary disable processing of further network messages
    PhotonNetwork.isMessageQueueRunning = false;
    LoadNewScene(newSceneName); // custom method to load the new scene by name
    while(newSceneDidNotFinishLoading)
    {
        yield return null;
    }
    PhotonNetwork.isMessageQueueRunning = true;
}
```

[Online Project] Rock Paper Scissor Multiplayer

Pada sesi ini kami membuat Project game online Rock Paper Scissor. Materi yang dipelajari dalam pembuatan Mini Golf ini yaitu :

1. Rock Paper Scissor Revisited
2. Syncing & Showing Players Hand
3. Syncing Score
4. End Screen (Win/Lose)

Disclaimer karena saya blom sampai membuat / Mengerjakan dari Project tersebut saya akan menjelaskan melalui tulisan saja.

1. kami berhasil membuat player masuk ke dalam lobby dan bermain game dengan player lain melalui server photon. kami membuat player melakukan sinkronisasi agar pergerakan dari player satu sama lain sama, kami juga membuat dan menampilkan ping, pada hari ini kami membuat sebuah script dan prefab, kami membuat prefab CardNetPlayer dan memuat script CardNetPlayer, di dalam script CardNetPlayer kami membuat logic game sederhana yang bertugas mengambil value dari card seperti rock, paper, dan scissors, di dalam script juga kami membuat sebuah event listener yang bertujuan untuk menjalankan method memilih attack value, kami juga menduplicate dan menambahkan code baru di script game manager.

2. kami berhasil membuat project rock paper scissors membuat sync antara player dengan player lain, tidak itu juga kami berhasil membuat menu baru seperti back to lobby dan beberapa fitur lain. kami berhasil menampilkan nama player di project rock paper scissors secara online, kami juga membuat panel screen win dan lose yang menampilkan nama player yang menang pada game. kami juga membuat button pada lobby tertutup dan tidak bisa diakses oleh player lain Ketika game dimulai, sama juga dengan button replay yang ada pada game, kami berhasil membuat logic bahwa game hanya dapat di replay oleh master game, jadi player lain tidak bisa membuat game mereplay dan Kembali ke menu ataupun lobby.
3. kami berhasil menambahkan sebuah fungsi pada game rock paper scissors yaitu membuat avatar character dan juga nickname pada game, kami berhasil membuat method menambahkan avatar character pada project rock paper scissors dengan menyimpan di playerprefs dan di upload ke photon, kami juga berhasil mengubah-ubah image avatar dengan menggunakan button yang dibuat, dengan begitu kami berhasil merubah avatar dengan klik pada button. kami juga menguji dengan mencoba build and run dan mencocokkan avatar pada game dari pemilihan avatar hingga game berlangsung.