Project: *Endless Runner*

Kita akan membuat game *endless runner* di mana *player* berlari tanpa henti, melompati rintangan, dan mengumpulkan koin. Berikut langkah-langkahnya:

1. Setup *Project*

* **Buat *Project* Baru:** Buka Unity Hub, buat *project* baru, pilih 2D, dan beri nama *project*-mu (misalnya, "EndlessRunner").
* **Import Aset:** Kumpulkan aset yang dibutuhkan, seperti:
  + Karakter *player* (dengan animasi lari dan lompat)
  + Rintangan (berbagai bentuk)
  + Koin
  + *Background* (bisa lebih dari satu untuk variasi)
  + *Sound effect* (lompatan, mengumpulkan koin, *game over*)
  + Musik *background*

2. Desain *Level*

* **Buat *Background* Berulang:** Susun *background* agar terlihat berulang (looping) ketika *player* bergerak, sehingga memberikan kesan "*endless*".
* **Tambahkan Platform:** Buat tanah atau platform tempat karakter berlari. Kamu bisa menggunakan *tilemap* atau menyusun *sprite* secara manual.
* **Variasi *Level*:** Agar tidak membosankan, buat variasi di *level*-mu, seperti mengubah ketinggian platform, menambahkan jurang, atau mengubah jenis rintangan.

3. Buat *Player* yang Keren

* **Tambahkan Karakter:** *Drag and drop* karakter *player*-mu ke dalam *scene*.
* **Komponen Fisika:** Tambahkan komponen Rigidbody2D untuk memberikan sifat fisika (gravitasi, *collision*) dan Box Collider 2D untuk mendeteksi tabrakan.
* **Animasi:** Pastikan karaktermu memiliki animasi berlari dan melompat (bisa dibuat di *Animation Window*).
* **Kontrol *Player*:** Buat *script* C# untuk mengontrol gerakan dan lompatan *player* (gunakan Input.GetAxis() untuk gerakan horizontal dan Input.GetKeyDown() untuk lompatan).

4. Tantang *Player* dengan Rintangan

* **Tambahkan Rintangan:** Letakkan berbagai jenis rintangan di *level*-mu.
* ***Collider*:** Tambahkan komponen Box Collider 2D pada rintangan untuk mendeteksi tabrakan dengan *player*.
* **Gerakan Rintangan:** Buat *script* untuk menggerakkan rintangan ke arah *player* dengan kecepatan tertentu. Kamu bisa menambah variasi kecepatan atau pola gerakan rintangan agar game lebih menantang.

5. Kumpulkan Koin Sebanyak-banyaknya!

* **Sebarkan Koin:** Letakkan koin di sepanjang *level*, baik di jalur yang mudah maupun yang sulit dijangkau.
* ***Collider*:** Tambahkan komponen Circle Collider 2D pada koin.
* ***Script* Koin:** Buat *script* untuk:
  + Mendeteksi *collision* antara *player* dan koin.
  + Menambah skor *player* saat mengumpulkan koin.
  + Menghilangkan koin dari *scene* setelah diambil.

6. Tampilkan Skor di UI

* **Buat *Canvas*:** *Canvas* adalah tempat untuk menampilkan elemen-elemen UI (User Interface), seperti teks, gambar, dan tombol.
* **Tambahkan Teks:** Buat objek Text di *canvas* untuk menampilkan skor.
* ***Script* UI:** Buat *script* untuk mengupdate nilai skor di *text* secara *real-time*.

7. *Game Over* Saat Kalah

* **Deteksi Tabrakan:** Di *script* *player*, tambahkan logika untuk mendeteksi *collision* dengan rintangan.
* ***Game Over* Screen:** Buat *scene* baru untuk *game over screen*. Tambahkan teks "Game Over" dan tombol untuk memulai ulang atau kembali ke menu utama.
* ***Trigger* *Game Over*:** Saat *player* bertabrakan dengan rintangan, *trigger* perpindahan ke *scene* *game over*.

8. *Polish* Game-mu

* ***Sound Effect* dan Musik:** Tambahkan *sound effect* untuk lompatan, mengumpulkan koin, dan *game over*. Tambahkan musik *background* yang sesuai dengan tema game.
* **Kesulitan Game:** Atur kesulitan game dengan mengubah:
  + Kecepatan gerakan *player* dan rintangan.
  + Jumlah rintangan dan koin.
  + Jarak antar rintangan.
* ***Build* Game:** Setelah game selesai, *build* game-mu ke platform yang diinginkan, seperti Windows, Android, atau WebGL.

Tips:

* **Mulai dari yang Sederhana:** Fokus pada fitur dasar terlebih dahulu, lalu kembangkan fitur lain secara bertahap.
* **Aset Sederhana:** Gunakan aset yang sederhana agar kamu bisa fokus pada *gameplay* dan pemrograman.
* **Berani Bereksperimen:** Jangan takut untuk mencoba hal-hal baru dan mengembangkan ide-ide kreatifmu!
* ***Debugging* itu Penting:** *Debugging* adalah bagian tak terpisahkan dari *game development*. Gunakan *console* dan *debugger* di Unity untuk mencari dan memperbaiki *error* dalam kodemu.

Selamat menjelajahi dunia *game development* dan wujudkan game impianmu! 🎮🎉