

COACHING PROGRAM Week 7

Testing dan Debugging

Bagian 1: Kenapa *Testing* dan *Debugging* itu Penting?

Bayangkan kamu sedang asyik bermain game, tiba-tiba karaktermu terjebak di dalam dinding, musuh tidak bisa dikalahkan, atau skor tidak bertambah. Itulah yang disebut *bug*, kesalahan dalam kode program yang menyebabkan game tidak berjalan sebagaimana mestinya.

Bug adalah musuh bebuyutan para *game developer*! *Bug* bisa merusak pengalaman bermain, membuat *player* frustrasi, dan bahkan merugikan pengembang game. Oleh karena itu, *testing* dan *debugging* adalah tahapan krusial dalam *game development*.

- **Testing:** Proses menemukan *bug* di dalam game. *Testing* dilakukan dengan cara menjalankan game dan mencoba berbagai aksi untuk melihat apakah ada yang tidak berjalan sesuai harapan.
- **Debugging:** Proses mencari tahu penyebab *bug* dan memperbaikinya. *Debugging* dilakukan dengan cara menganalisis kode program dan menggunakan *tools debugging* di Unity.

Bagian 2: Teknik-teknik *Testing*

Ada berbagai teknik *testing* yang bisa kamu terapkan, di antaranya:

- **Playtesting:** Cara paling sederhana, yaitu dengan memainkan game-mu sendiri atau meminta orang lain untuk memainkannya. Perhatikan apakah ada *bug* atau hal-hal yang perlu diperbaiki di game.
- **Unit Testing:** Menguji bagian-bagian kecil dari kode program (unit) secara terpisah. Ini membantu menemukan *bug* lebih awal sebelum mereka menyebabkan masalah yang lebih besar.
- **Integration Testing:** Menguji interaksi antar berbagai unit kode program. Ini memastikan bahwa semua bagian game bekerja sama dengan baik.
- **Regression Testing:** Menguji ulang game setelah melakukan perubahan kode program. Ini memastikan bahwa perubahan yang kamu lakukan tidak memunculkan *bug* baru atau merusak fitur yang sudah ada.

COACHING PROGRAM Week 7

Testing dan Debugging

Bagian 3: *Debugging* di Unity

Unity menyediakan *tools debugging* yang *powerful* untuk membantumu menemukan dan memperbaiki *bug*:

- **Console Window:** Menampilkan pesan *error*, *warning*, dan informasi lain dari game. Perhatikan pesan-pesan di console untuk mencari tahu *bug* yang terjadi.
- **Debugger:** Memungkinkanmu untuk menghentikan eksekusi program pada titik tertentu, memeriksa nilai variabel, dan menjalankan kode program langkah demi langkah. Ini sangat membantu untuk menganalisis alur program dan menemukan penyebab *bug*.
- **Profiler:** Menganalisis performa game, seperti penggunaan CPU, memori, dan rendering. Ini membantu menemukan *bottleneck* performa yang bisa menyebabkan game menjadi *lag*.

Bagian 4: *Tips Debugging*

- **Baca Pesan *Error*:** Pesan *error* di console memberikan informasi penting tentang jenis *bug* dan di mana *bug* tersebut terjadi.
- **Gunakan `Debug.Log()`:** Tambahkan `Debug.Log()` di kode programmu untuk menampilkan nilai variabel atau pesan lain ke console. Ini membantumu memahami alur program dan menemukan kesalahan logika.
- ***Breakpoints*:** Gunakan *breakpoint* di *debugger* untuk menghentikan eksekusi program pada titik tertentu dan memeriksa nilai variabel.
- **Jalankan Langkah demi Langkah:** Gunakan *step over*, *step into*, dan *step out* di *debugger* untuk menjalankan kode program langkah demi langkah dan melihat perubahan nilai variabel.
- **Isolasi Masalah:** Jika kamu menemukan *bug*, cobalah untuk mengisolasi masalah dengan menonaktifkan bagian-bagian kode program atau objek di *scene*. Ini membantumu mencari tahu bagian mana yang menyebabkan *bug*.

COACHING PROGRAM Week 7

Testing dan Debugging

Tugas:

- Cari dan perbaiki *bug* di *project game endless runner* yang telah kamu buat di minggu ke-6.
- Tambahkan fitur baru ke dalam game (misalnya, *power-up*, musuh, dll.) dan lakukan *testing* untuk memastikan fitur tersebut bekerja dengan baik.

Dengan menguasai teknik *testing* dan *debugging*, kamu bisa menghasilkan game yang berkualitas dan menyenangkan untuk dimainkan. Selamat menjadi *game developer* profesional! 🎮