

هو علشان نحول الكائنات من *model transformation* بعد ال *inverse camera matrix* السبب اللي بيخلينا نضرب في ال *model space* ل *camera (or view) space*.

هي *model-view transformation* الخطوات اللي بنمشي عليها في ال

1. **Model Transformation:** دي أول خطوة بنعمل فيها تحويل للكائنات من النظام الخاص بيهم (Model Space) للنظام العام بتاع المشهد أو العالم (World Space).

2. **View Transformation (Camera Transformation):** اللي *World Space* ل *View Space* لما نيجي نحول من ال *camera matrix* هو منظور الكاميرا، بنستخدم

*matrix* بتاع كاميرا ال *inverse* اللي بيحصل هنا إننا بنشوف الكائنات من منظور الكاميرا، وعلشان نعمل ده، لازم نجيب ال *inverse* ونضرب فيه. لما نضرب في ال

إحنا بنحول المشهد من النظام العالمي لمنظور الكاميرا، *inverse camera matrix* بالتالي، لما بنضرب في ال

الفكرة الأساسية إن الكاميرا مش بنعتبرها "كائن" بيتحرك داخل المشهد؛ الكاميرا هي المرجع اللي بنشوف منه المشهد كله. علشان كذا، اللي بيهما هو نقل المشهد بالنسبة للكاميرا بدل ما ننقل الكاميرا نفسها.

ليه بننقل المشهد بدل ما ننقل الكاميرا؟

لو فكرنا فيها، إحنا عايزين دايماً نحتفظ بالكاميرا في مكان ثابت و"تخيل" إن الكاميرا هي اللي ثابتة، والمشهد كله هو اللي بيتحرك حوالها علشان يظهر في مجال الرؤية. بدل ما ننقل الكاميرا لمكان جديد، بننقل كل الكائنات في المشهد بحيث يكونوا بالنسبة للكاميرا *view space*.

إيه *inverse camera matrix* ليه بنستخدم ال

لو حبينا "نتحرك بالكاميرا" علشان تبص من زاوية مختلفة، نقدر نتخيل إن الكاميرا بتتحرك فعلاً. ولكن، علشان نفهم حركة الكاميرا، بنعكس العملية

- بدل ما نحرك الكاميرا للأمام، بنحرك المشهد كله للخلف.
- بدل ما ندور الكاميرا، بنلف المشهد كله في الاتجاه العكسي.

*inverse* استخدام *view space* دي عملية أشبه بإنك تحرك المشهد حوالين الكاميرا، وكأن الكاميرا ثابتة في نقطة الأصل بتاع *camera matrix* للمشهد، وده بيخلي الكاميرا فعلياً "ثابتة" بينما إحنا بنغير منظورنا للمشهد