



# **RELAZIONE**

## **“PRODUCTION PIPELINE”**



*Marta Martino*

## La scena 3D

La scena presenta la riproduzione della pipeline richiesta, più una UI 2D per l'interrogazione delle risorse e dei moduli.

Non avendo a disposizione assets, sono state utilizzate forme semplici per la rappresentazione 3D degli elementi, distinguibili per forme e colori.

La UI presenta diverse opzioni:

- modifica della scala temporale con uno slider;
- mettere in pausa la simulazione;
- selezione di moduli e risorse da appositi elementi dropdown.

Per visualizzare le statistiche, è anche possibile selezionare direttamente gli elementi con il mouse. La UI si aggiornerà di conseguenza.

In scena sono presenti due camere: una fissa con visuale dall'alto sulla pipeline, e l'altra mobile che consente di muoversi nello spazio 3D.

## Organizzazione del codice e funzionamento

Le principali classi sono organizzate come segue:

- Source
  - BasicSource
    - Base
    - Body
    - Detail
  - AssembledSource
- Module
  - Assembler
  - Buffer
  - Conveyor
  - FlowSplitter
  - QualityAssurance
  - SourceProvider
  - SourceReceiver
- PipelineManager: manager della pipeline, mantiene aggiornate le strutture dati che forniscono dati alla UI.
- UIManager: manager della UI, si registra agli eventi del PipelineManager per fornire informazioni sugli elementi in pipeline.
- MouseInteraction: per la selezione degli elementi nella scena e conseguente visualizzazione delle statistiche.
- BillboardEffect: i testi 3D nella scena sono orientati verso la camera, in modo da favorirne la leggibilità.
- Utilities: funzioni di utilità.
- CameraManagement: gestisce lo switch tra le due camere (una fissa e una mobile).

- **MobileCamera**: gestisce il movimento della camera mobile (WASD + click & drag del mouse).

I moduli della pipeline sono programmati per funzionare in collegamento ad altri moduli, che devono essere assegnati via editor.

I moduli si passano le risorse tra di loro e comunicano tramite appositi eventi. Quando la risorsa viene “passata” da un modulo all’altro, il suo parent cambia.

Il ciclo di vita delle risorse inizia dai SourceProvider e termina nei SourceReceiver. I SourceProvider presentano un’opzione modificabile da editor che consente di avviare la produzione non appena la simulazione inizia, per evitare di dover attendere il tempo tra una produzione e l’altra. Di default, questo parametro è settato a true.

Le tre risorse principali (Base, Body, Detail) sono implementate come prefab (in Assets/Resources/Prefabs/) e vengono istanziate dinamicamente. Ogni risorsa è rappresentata da una forma diversa.

Anche i moduli sono prefab (Assets/Prefabs) ed è stata scelta una combinazione forma/colore differente per ogni tipologia.

La visualizzazione delle statistiche può essere eseguita in due modi: o tramite gli appositi dropdown, o selezionando direttamente gli elementi col mouse.

Sia le risorse che i moduli hanno associato un oggetto TextMesh per identificarli più facilmente nell’ambiente 3D. Inoltre, tutti gli oggetti sono selezionabili col mouse in modo da visualizzarne le statistiche.

La camera mobile risponde ai tasti WASD + click&drag del mouse: in questo modo si è liberi di interagire anche con la UI.

## Assunzioni

- Tutti i moduli (Module) hanno due array, rispettivamente per i moduli in entrata e uscita. Se per un certo modulo non è previsto l’uso di uno dei due (es. SourceProvider non ha moduli in entrata), e l’utente glielo assegna comunque, il modulo non necessario verrà ignorato. Allo stesso modo, i moduli “in eccesso” non verranno considerati (es. 3 moduli in output per il FlowSplitter).
- Tutti i moduli hanno bisogno di un nome univoco che venga associato da editor.
- Nei moduli QualityAssurance, i tipi dei moduli output non sono specificati. Idealmente si potrebbero avere in output due moduli Conveyor che portano ad altrettanti moduli Receiver, ma dal modulo QualityAssurance non si avrebbero abbastanza informazioni per sapere rispettivamente quale Conveyor porta al Destroyer e quale porta allo Storer. Si assume quindi che almeno uno dei due moduli in output sia un Receiver.
- I prefabs delle BasicSource sono impostati in modo da avere la mesh 3D settata come child. Per il posizionamento delle risorse sui conveyor, si assume che le mesh abbiano il pivot settato al loro centro.
- Le AssembledSource sono composte da esattamente 2 Sources.
- Nell’implementazione corrente, si assume che durante la simulazione non vengano aggiunti nuovi moduli.

- Le statistiche presentano le informazioni correnti relative ai moduli e alle risorse. In particolare, le risorse selezionabili sono quelle attualmente presenti nella pipeline. Quando due risorse vengono assemblate in una risorsa più grande, scompaiono dalla lista delle risorse selezionabili e compare la nuova risorsa assemblata. Se la velocità di lavoro della pipeline è elevata, la lista delle risorse viene costantemente aggiornata e risulta di difficile consultazione. Per questa ragione, la lista viene resa consultabile solo quando la simulazione viene messa in pausa dall'apposita checkbox nella UI. Questo consentirà comunque di avere una situazione chiara delle risorse e della loro collocazione attuale.
- SourceReceiver può essere di due tipi: destroyer o storer. Questi moduli ricevono la risorsa e la piazzano sulla propria mesh. Per evitare sovraccarichi di memoria dovuti a un'eccessiva quantità di sources generate, sia lo storer che il destroyer distruggono il gameobject della risorsa. In particolare, per evitare che graficamente le sources si sovrappongano, il SourceReceiver mostra soltanto l'ultima risorsa ricevuta, ed elimina la precedente.
- PipelineManager e CameraManager, per ragioni di comodità, sono implementati come Singleton. Nel caso in cui la scena dovesse ospitare più pipeline o più gestori di camere, sarebbe necessario renderli comuni istanze di oggetti.

## Plugin

Si sfrutta il plugin DOTween per il movimento delle risorse sui conveyors.