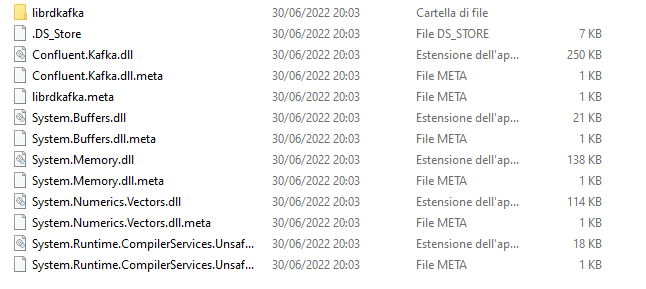
**Istruzioni per avviare il progetto.**

A cura di Renato Esposito

* **Requisiti**
  + *Unity 2020.3.36f1* (https://unity3d.com/get-unity/download/archive)
  + *Visual Studio o qualsiasi altro editor* (https://visualstudio.microsoft.com/it/downloads/)
  + *Libreria Confluent Kafka* (leggi sotto)
  + (Opzionale) *GitHub Desktop* https://desktop.github.com/

**Procedura**

* ***Confluent Kafka***: per poter installare la libreria è necessario creare un nuovo progetto (separato da Unity) e scaricare dal Gestore di pacchetto NuGet “Confluent.Kafka” (link del progetto: <https://github.com/confluentinc/confluent-kafka-dotnet/>). Nella cartella del progetto (solitamente la cartella Packages, ma potrebbe cambiare in base alla versione dell’IDE) sono ora visibili alcuni file:  
    
    
    
    
    
    
    
    
  copiare questi file e ritornare nella repository del progetto, andare in  
  “Assets”, creare la cartella “Plugins” ed incollare i file.

**N.B**: riprendendo il progetto così com’è stato lasciato, non è necessario creare ulteriori cartelle, ma risulta fondamentale la creazione di un nuovo progetto (separato da Unity) e il recupero dei file scaricati dal Gestore NuGet.

* ***Funzionamento***: avviare l’ambiente Unity e la sessione (tasto “play”), a questo punto è necessario attendere che il sistema riceva da un “producer” (cioè da chi si occupa del Digital Twin) le coordinate.   
  L’avatar ha comunque la possibilità di potersi muovere attraverso i pulsanti W A S D, inoltre leggendo la console è possibile visualizzare le combinazioni di tasti che permettono di:  
  memorizzare i dati per eventuali simulazioni (spiegazione nel prossimo punto), aprire ciascuna finestra, cambiare la camera, saltare e ritornare sul piano (dopo il salto).
* ***Simulazione***: nel caso in cui non fossero ancora presenti le camere è possibile creare dei test semplicemente camminando per l’ambiente (attraverso i tasti W A S D) e al termine dell’operazione premere il tasto “8”, si creerà all’interno della cartella “Assets” un file “data.json” che deve essere inviato (tramite mail o USB) a chi si occupa del digital twin/anomaly detection.   
  **N.B**: quando non sarà più necessario effettuare simulazioni si raccomanda di commentare/eliminare la funzione DataMemorization e la coroutine UpdateEachSecond.

Per qualsiasi info: [renato.esposito1999@outlook.com](mailto:renato.esposito1999@outlook.com) o renato.esposito001@studenti.uniparthenope.it