

# PASSARE DA GRASSHOPPER A VECTORWORKS

---



VECTORWORKS®  
ARCHITECT



## INTRODUZIONE

Il **visual scripting** ti permette di “costruire” un processo parametrico in modo più semplice e versatile rispetto alla programmazione tradizionale. I suoi schemi logici in forma di diagrammi di flusso permettono di visualizzare lo script. I recenti progressi nelle tecnologie costruttive permettono la definizione di forme uniche e complesse tramite strumenti per il visual scripting: queste funzioni consentono di costruire modelli digitali e stanno divenendo sempre più diffuse in campo architettonico. Ci sono diversi strumenti di visual scripting disponibili per gli architetti, come Grasshopper®, un modulo per Rhinoceros® 3D. Molti architetti usano Grasshopper per costruire geometrie parametriche, per “cuocerle” in Rhino, e poi importarle in un altro programma (“cuocere” è il processo di conversione dell’anteprima di Grasshopper nella geometria statica di Rhino). Importare una geometria statica che non può essere modificata parametricamente è però antitetico allo scopo del visual scripting. Riteniamo infatti che sia fondamentale poter modificare la geometria parametricamente dopo che è stata incorporata in un processo di design. Il modulo **Marionette** è una funzionalità di visual scripting completamente integrata in Vectorworks®, per cui non c’è un’interfaccia separata o una sconnessione tra la geometria “cotta” e lo script che l’ha creata.



**FIGURA 1**  
Un network Marionette è in grado di creare delle gerarchie lungo un percorso

## COSA CAMBIA TRA GRASSHOPPER E VECTORWORKS?

Grasshopper è un modulo per Rhino, non un prodotto indipendente, per cui richiede Rhino per essere eseguito ed è dunque limitato dalle capacità di Rhino. D’altra parte, Vectorworks Architect è un software di modellazione 3D e BIM con uno strumento di visual scripting pienamente integrato: Marionette. Questo significa che Marionette crea geometrie, oltre a effettuare analisi di documenti, creare oggetti parametrici e comandi di menu. Vectorworks possiede estese capacità 2D e così anche Marionette. Vectorworks è un software cross-platform (Mac e Windows). Se si fa uso di Vectorworks Designer, allora è anche possibile utilizzare Marionette con funzioni specifici per Landmark (il modulo di Vectorworks per l’architettura del territorio) e Spotlight (il modulo Vectorworks per la progettazione di scenografie ed eventi). Marionette è una funzione integrata in Vectorworks, non un modulo con un’interfaccia separata, per cui non ci sono finestre separate o nuovi comandi da imparare.

---

Marionette si basa completamente su Python e lo script di ogni nodo può essere facilmente esposto e modificato facendo doppio clic sul nodo. Questo rende Marionette altamente personalizzabile anche con una conoscenza basilare di Python®. Marionette non è solo uno strumento di visual scripting ma anche un modo per imparare a usare Python.

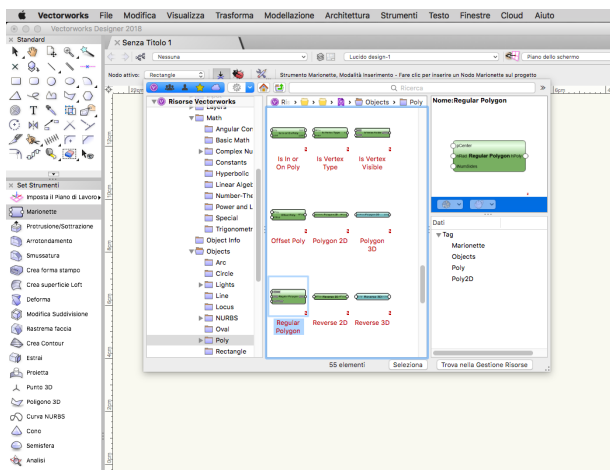
## PERCHÉ PASSARE DA GRASSHOPPER A MARIONETTE

A differenza di Rhino, Vectorworks Architect dispone di un ambiente di modellazione flessibile per l'esplorazione del design, ma anche una gamma completa di strumenti per la modellazione BIM e l'interscambio dei dati. Vectorworks possiede estese capacità di modellazione 3D, oltre a solide capacità di creare tavole 2D e presentazioni BIM e Marionette può interagire con tutte queste capacità. Marionette lavora all'interno di Vectorworks in maniera simile a come Grasshopper lavora come estensione di Rhino, per cui ogni caratteristica è limitata dalla capacità del software principale. Avere però così tante capacità con cui operare rende Marionette molto più potente, e siccome Marionette è integrato all'interno di Vectorworks non c'è alcun bisogno di operare su file multipli. Ci sono anche benefici diretti dall'utilizzo di Marionette come strumento di scripting. Questi includono:

- Interfaccia nativa
- Creazione di oggetti parametrici
- Creazione di comandi di menu
- Capacità di analisi
- Scripting basato su Python

### INTERFACCIA NATIVA

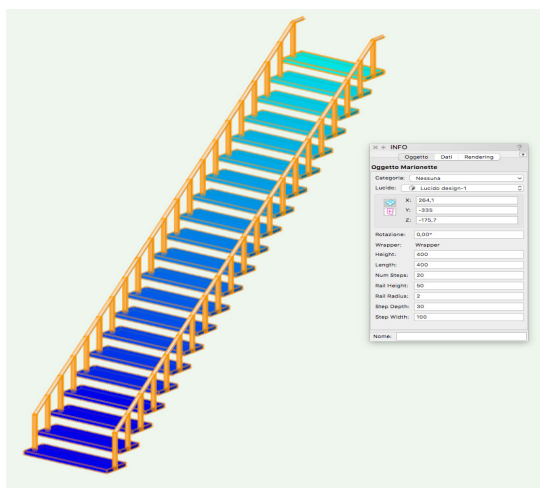
Marionette è uno strumento di Vectorworks e i nodi di Marionette sono oggetti parametrici che esistono in Vectorworks: non c'è necessità di un'interfaccia separata o di un formato di file diverso. I nodi vengono inseriti sul piano dello schermo in scala reale, per cui se si sta lavorando su un oggetto di grandi dimensioni o su un lucido scalato in una vista 3D, i nodi esisteranno parallelamente allo schermo e manterranno una dimensione corretta. I nodi possono essere selezionati proprio come ogni altro oggetto in Vectorworks. Possono venire inseriti in Categorie e Lucidi a piacere, il che significa che è possibile assegnare loro attributi e controlli di visibilità differenti. Siccome sono oggetti all'interno di Vectorworks, è possibile utilizzarli per creare oggetti parametrici, che possono essere modificati e distribuiti con la massima facilità.



**FIGURA 2**  
Strumento Marionette con accesso al Selettore Risorse

## CREAZIONE DI OGGETTI PARAMETRICI

Dato che i nodi di Marionette sono elementi nativi di Vectorworks, il network che creano può venire convertito in un oggetto parametrico. Quando un oggetto Marionette viene creato da un network, i parametri di input, incluso il nodo “cursore”, possono venire esposti nella Tavolozza Informazioni, e i nodi possono venire modificati proprio come ogni altro oggetto parametrico. Se l’oggetto Marionette utilizza il controllo della geometria, facendo doppio clic si può attivare lo strumento Modifica, che permette di modificare la geometria di controllo e cambiare l’oggetto. Se invece non c’è geometria di controllo, facendo doppio clic si accede al network, in modo da poterlo modificare. Gli oggetti Marionette possono essere trasferiti di progetto in progetto come una qualsiasi risorsa Vectorworks e possono venire copiati o duplicati in serie all’interno del disegno.



**FIGURA 3**  
Un oggetto parametrico creato con Marionette

## CREAZIONE DI COMANDI DI MENU

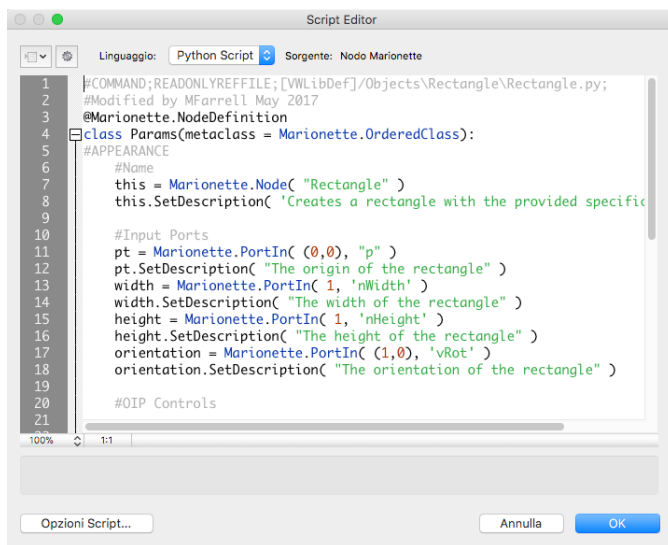
Tramite Marionette è possibile creare oggetti parametrici ed anche comandi di menu. Tali comandi vengono registrati nella propria Cartella Utente e diventano disponibili per ogni progetto che viene aperto successivamente. I comandi possono essere modificati e trasferiti in modo analogo ai nodi Marionette e agli oggetti parametrici. La funzionalità dei comandi è ottima per automatizzare processi tediosi, per organizzare gli oggetti o per estrarne dati.

## CAPACITÀ DI ANALISI

Dato che Vectorworks è un software BIM, Marionette può essere usato per estrarre dati dagli oggetti inseriti sul progetto. File ricchi di dati come gli Shapefile possono venire importati in Vectorworks e poi modificati tramite Marionette. Siccome Marionette è un modulo nativo di Vectorworks ed è basato su Python, risulta facile creare network che estraggono dati da altre fonti come file sul proprio computer o da internet.

## SCRIPTING SU PYTHON

I nodi Marionette sono scritti nel linguaggio di programmazione di Python: facendo doppio clic su un nodo si espone lo script Python presente al suo interno. Se si ha familiarità con Python, questa modalità di lavoro rende semplice il processo di creazione di nodi personalizzati. Si può digitare un nuovo script direttamente in un nodo, per poi registrarlo come un nuovo oggetto parametrico e aggiungerlo alle proprie risorse di Vectorworks, che possono essere facilmente condivise con altri. Le librerie Python esterne, come SciPy, NumPy, o PIL, possono venire aggiunte alle librerie Python di Vectorworks, permettendo di avere a disposizione innumerevoli possibilità di disegno parametrico.




```
1 #COMMAND;READONLYREFFILE;[VWLibDef]/Objects\Rectangle\Rectangle.py;
2 #Modified by MFarrell May 2017
3 @Marionette.NodeDefinition
4 class Params(metaclass = Marionette.OrderedClass):
5     #APPEARANCE
6     #Name
7     this = Marionette.Node( "Rectangle" )
8     this.SetDescription( 'Creates a rectangle with the provided specific
9
10    #Input Ports
11    pt = Marionette.PortIn( (0,0), "p" )
12    pt.SetDescription( "The origin of the rectangle" )
13    width = Marionette.PortIn( 1, 'nWidth' )
14    width.SetDescription( "The width of the rectangle" )
15    height = Marionette.PortIn( 1, 'nHeight' )
16    height.SetDescription( "The height of the rectangle" )
17    orientation = Marionette.PortIn( (1,0), 'vRot' )
18    orientation.SetDescription( "The orientation of the rectangle" )
19
20    #OIP Controls
21
```

FIGURA 4

Lo script modificabile Python all'interno di un nodo

# COME PASSO DA GRASSHOPPER A MARIONETTE?

## INTERFACCIA UTENTE

Marionette è un modulo integrato in Vectorworks, non un plug-in con un'interfaccia separata, per cui segue le stesse regole di altri strumenti Vectorworks. Se non si ha familiarità con Vectorworks, vedere il documento [Passare da Rhino a Vectorworks](#). Lo strumento Marionette  si trova nella Tavolozza Strumenti "Modellazione"; dopo averlo attivato, tramite la Barra di Modo si possono scegliere un nodo dal Selettore Risorse. All'interno del Selettore Risorse si può fare clic su cartelle e sottocartelle organizzate in classi simili ai componenti di Grasshopper o si può cercare per nomi o parole chiave, tramite la barra di ricerca. Facendo clic su un nodo viene visualizzata la sua anteprima nel pannello Anteprima, dove si possono leggere meglio i nomi delle porte input ed output. Facendo doppio clic si seleziona il nodo e lo si può posizionare nell'area di disegno. Una volta che il nodo è nell'area di disegno, lo si può collegare ad altri nodi facendo clic su un porta di output e trascinando un filo fino alla porta di input di un altro nodo, proprio come in Grasshopper.

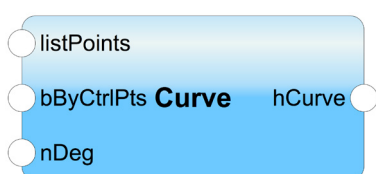
## NODI MARIONETTE

I nodi Marionette sono oggetti parametrici di Vectorworks, per cui possono venire trattati allo stesso modo di altri oggetti: possono essere assegnati ad una Categoria specifica e si possono modificare gli attributi. I nodi vengono inseriti automaticamente in Categorie, ognuna delle quali ha un particolare colore sfumato, in base alla tipologia di funzione del nodo. I nodi mostrano sempre il loro nome completo e quello delle loro uscite. I nomi di ogni porta iniziano con un prefisso che indica il tipo di dati che la porta accetta e, se necessario, una loro descrizione.

PREFISSO	DESCRIZIONE
<b>b</b>	Booleana (Vero o Falso)
<b>h</b>	Maniglia (un oggetto Vectorworks come un rettangolo o un simbolo)
<b>i</b>	Intero (un numero intero solitamente usato per contare o indicizzare)
<b>m</b>	Matrice (una serie di numeri usati in certe espressioni matematiche)
<b>n</b>	Numero/Dimensione (un numero solitamente usato per una distanza o una misura)
<b>p</b>	Punto (una coordinata 2D o 3D nello spazio)
<b>rgb</b>	Valore RGB a 16 bit (un set di valori colore basati su rosso, verde e blu che varia tra 0 e 65536)
<b>s</b>	Stringa (informazione definita come testo)
<b>v</b>	Vettore (un dato 2D o 3D che definisce una grandezza e una direzione)
<b>item</b>	Un singolo valore di un qualsiasi tipo
<b>list</b>	Una lista di valori di un qualsiasi tipo

---

Nel nodo raffigurato in **FIGURA 5**, le tre tipologie di input sono una lista di punti per definire una curva, un valore vero o falso per controllare se la curva viene definita tramite punti di controllo o punti interpolati e un numero, che definisce il grado della curva. La porta di output è una maniglia per la curva creata. Per saperne di più sul nodo è sufficiente fare clic sul bottone “Descrizione” visibile nella Tavolozza Informazioni: si apre un dialogo che fornisce informazioni aggiuntive su ognuna delle porte e sulle funzioni del nodo.



**FIGURA 5**  
Un nodo Curva con prefissi che descrivono le sue porte in ingresso e in uscita

Per creare rapidamente un nodo personalizzato partendo da uno già esistente è sufficiente fare doppio clic su un nodo per aprire lo Script Editor e visualizzare il codice Python che definisce l’aspetto e la funzione del nodo. La prima linea di codice in ogni nodo standard della libreria è un comando che definisce da dove viene il codice, per cui i nodi vengono aggiornati automaticamente quando si installano dei Service Pack di aggiornamento. Per creare il proprio nodo, cancellare la prima riga di codice e poi modificarlo come si preferisce. Questi nodi personalizzati possono venire salvati come oggetti parametrici di Vectorworks e posizionati in un file negli Elementi base della Cartella Utente.

## COSTRUIRE UN NETWORK

I network di Marionette vengono costruiti impostando parametri tramite dei nodi di input ed effettuando azioni su quei parametri tramite altri nodi, proprio come i network in Grasshopper. Per aggiungere un valore a un nodo di input, selezionare il nodo e inserire il valore nel campo dati nella Tavolozza Informazioni. Ci sono anche alcuni nodi input che insegnano come modificare il proprio script per creare menu a comparsa, bottoni e cursori. I nodi sono connessi tra loro tramite fili ma, a differenza di Grasshopper, più di un filo può essere connesso a una porta di input: il nodo tratterà le diverse fonti di dati come una singola lista.

## WRAPPER

Quando un network, o parte di esso vengono costruiti, possono essere “confezionati.” Un Wrapper è un network che è stato consolidato per operare come un singolo nodo. È possibile assegnare un nome ai nodi di input ed a quelli scollegati e quando questi nodi fanno parte di un Wrapper, i valori dei nodi di input diventano accessibili tramite la Tavolozza Informazioni; le porte scollegate dei nodi con nome compariranno all’esterno del Wrapper. Questa metodologia funziona in modo simile ai cluster di Grasshopper ma i Wrapper hanno funzionalità aggiuntive, dato che possono venire convertiti in comandi di menu o in oggetti parametrici.



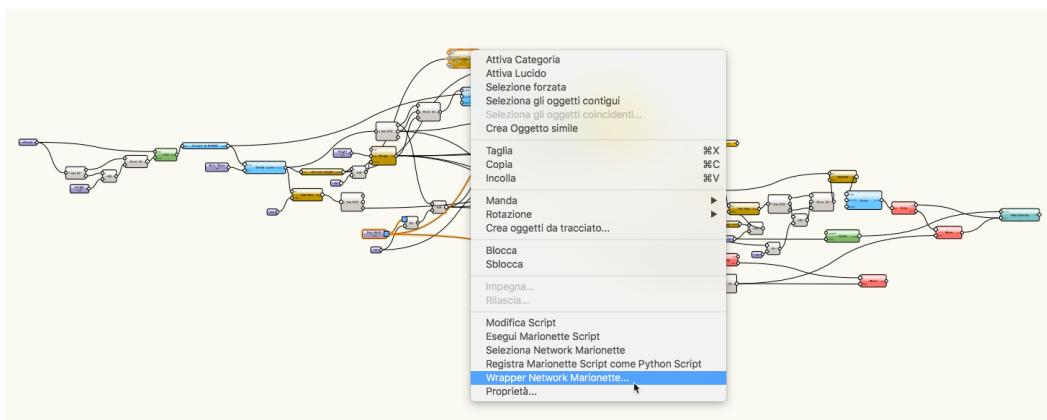


FIGURA 6

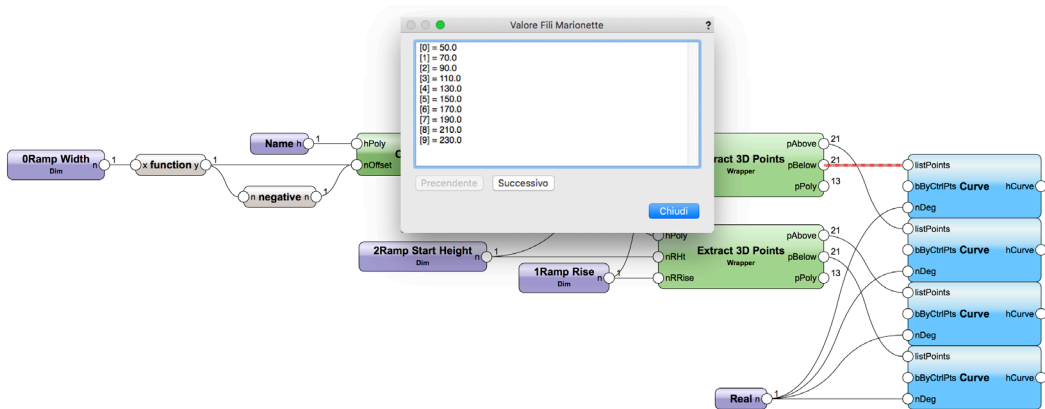
Fare clic destro su un qualsiasi nodo in un network e selezionare la voce "Wrapper Network Marionette"

## OPERARE SU UN NETWORK

A differenza di Grasshopper, quando un nodo viene posizionato nell'area di disegno non crea immediatamente una geometria di anteprima. Nodi e network non creano geometrie fino a quando non vengono "eseguiti", facendo clic destro e scegliendo "Esegui Marionette Script" o facendo clic su "Esegui" nella Tavolozza Informazioni. Ciò è simile alla "cottura" in Grasshopper, in cui la geometria esiste separata dal network. Quando un network viene eseguito in Marionette viene creato un gruppo di geometrie e questo gruppo è connesso al network: quando il network viene poi eseguito di nuovo, la geometria viene sostituita da quella nuova. Se si vuole creare più iterazioni della geometria, è possibile disconnetterla dal network, intervenendo sul gruppo della geometria o modificando il nome del gruppo.

## MODALITÀ DEBUG

Per comprendere quali informazioni viaggiano attraverso i nodi è possibile selezionare la modalità "Debug" nella Barra di Modo invece della modalità "Inserimento nodo". L'icona per la modalità debug è uno scarafaggio 🐞. Quando viene selezionata la modalità Debug è possibile fare clic su un filo per vedere le informazioni che lo attraversano. Ogni volta che si fa clic su un filo in modalità Debug il network viene eseguito, per cui se si ha un network complesso che richiede tempo per venire eseguito o un network che produce valori casuali che verrebbero modificati a ogni esecuzione successiva è possibile fare clic sulle Impostazioni (l'icona con la chiave inglese e la matita) nella Barra di Modo ed attivare l'opzione "Cache esecuzione in modalità Debug": in questo modo tutti i valori dell'ultimo avvio del network vengono messi in una cache e possono essere analizzati senza dover eseguire ogni volta il network. Se non ci sono fili che escono da una porta di output, è opportuno attaccare un nodo "Dummy End" alla porta, in modo da creare un filo e poterne analizzare i valori.



**FIGURA 7**  
 Facendo clic su un filo mentre si lavora in modalità Debug vengono visualizzati i valori che attraversano il filo.

## CONCLUSIONI

Il visual scripting si basa principalmente sullo snellimento del flusso di lavoro tramite la costruzione di processi parametrici. Se la geometria creata da uno script deve venire esportata su un altro software, l'aspetto parametrico della geometria viene perso. Anche le connessioni dirette non risolvono definitivamente il problema, perché rimangono limitate alle capacità di interazione tra due software. Per sfruttare appieno ed insieme i vantaggi del BIM e del visual scripting, la capacità di scripting deve essere completamente integrata nel software BIM. Grazie alle vaste capacità di modellazione 3D di Vectorworks, il modulo Marionette permette di costruire geometrie parametriche complesse, e grazie alle funzioni BIM e di messa in tavola, Marionette può anche essere utilizzato per analizzare disegni e progetti. In più, la completa integrazione di Marionette all'interno di Vectorworks permette la creazione di oggetti parametrici e comandi di menu. Non da ultimo, la possibilità di intervenire sugli script Python presenti ogni nodo garantisce le infinite capacità di personalizzazione messe a disposizione da Marionette.

# SCOPRI DI PIÙ

su come Vectorworks può aiutarti a sviluppare  
e incrementare il tuo flusso di lavoro BIM.

Manda una mail a [vectorworks@videocom.it](mailto:vectorworks@videocom.it).

**VECTORWORKS, INC.**

7150 Riverwood Drive, Columbia, MD 21046-1295 USA

[www.vectorworks.net](http://www.vectorworks.net)

©2017-2018 Vectorworks, Inc. & VideoCOM.

Tutti i diritti sono riservati.



DISTRIBUTORE ESCLUSIVO PER L'ITALIA

**videocom**

0383.366712 | [www.videocom.it](http://www.videocom.it)