



**SAPIENZA**  
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria

Corso di laurea triennale in Ingegneria Informatica

Tesi di laurea

Blender Ecosystem Generator (BEG)

Relatore:

**Prof. Andrea Vitaletti**

Correlatore:

**Dott. Emanuel Demetrescu**

Laureando:

**Gaetani Edoardo**

**1471751**

Anno accademico 2013/2014



# INDICE

<b>Introduzione</b> .....	2
<b>Capitolo 1 - Blender e Python</b> .....	3
1.1 - Cos'è Blender .....	3
1.2 - Interfaccia grafica .....	5
1.3 - Blender e Python .....	8
<b>Capitolo 2 - Obiettivi del progetto</b> .....	9
2.1 - Motivazioni .....	9
2.2 - Problematiche riscontrate .....	11
<b>Capitolo 3 - Installare Python</b> .....	13
3.1 - Configurazione ambiente .....	13
3.1.1 - Linux .....	14
3.1.2 - Mac OS X .....	16
3.1.3 - Windows .....	17
<b>Capitolo 4 - Struttura Add-On</b> .....	18
4.1 - Interfaccia .....	18
4.2 - Ecosystem Library .....	20
4.2.1 - Lista Ecosistemi .....	21
4.2.2 - Object to Ecosystem .....	23
4.3 - Landscape Editor .....	25
4.3.1 - Lista di Particle System .....	26
4.3.2 - Object to Ecosystem Editor .....	29
4.4 - Installazione .....	30
<b>Capitolo 5 - Conclusioni</b> .....	32
<b>Bibliografia</b> .....	35
<b>Ringraziamenti</b> .....	36

## Introduzione

Il lavoro descritto in questa tesi è la realizzazione di uno script in Python, che utilizzato in Blender, renda meno complicato l'uso del software all'utente il cui obiettivo è la gestione di ecosistemi.

Il lavoro di tesi viene strutturato come segue:

- Nel primo capitolo viene presentato il software Blender, con tutte le sue funzionalità, e il linguaggio di programmazione Python.
- Nel secondo capitolo vengono illustrati gli obiettivi che ci poniamo di raggiungere in seguito alla realizzazione di questo progetto e i problemi emersi durante lo sviluppo.
- Nel terzo capitolo viene spiegato come l'utente debba configurare il sistema per accedere ai moduli standard di Python.
- Il quarto capitolo è dedicato alla descrizione delle tecniche implementative usate e del funzionamento dello script.
- Infine vi è una conclusione che riassume i risultati ottenuti e propone delle possibili estensioni di quanto fino ad ora realizzato.

# Capitolo 1

## Blender e Python

In questo capitolo verranno presentati gli strumenti utilizzati e le loro caratteristiche. Nella prima parte verrà presentato il software Blender e la sua interfaccia, mentre nella seconda parte verrà illustrato il ruolo di Python all'interno del software.

### 1.1 Cos'è Blender?

Blender è un software open-source di modellazione, animazione e rendering di immagini tridimensionali. In principio, però, era stato ideato per essere utilizzato come applicazione interna dallo studio di animazione olandese NeoGeo e solo nel 1998, grazie all'autore principale Ton Rosendal, nacque l'idea di distribuire il programma per continuare lo sviluppo. Dopo una raccolta fondi, Rosendal raggiunse il suo obiettivo e nell'Ottobre del 2002 fu rilasciato il codice sorgente di Blender. Oggi Blender è un progetto open-source molto attivo ed è guidato dalla Blender Foundation.

Blender è disponibile per tutti i più importanti sistemi operativi come Microsoft Windows, Mac OS X, Gnu/Linux, ma anche per sistemi derivati Unix meno conosciuti come FreeBSD.

Nonostante sia un software libero, ha un robusto insieme di funzionalità paragonabili a quelle di altri noti software a pagamento come Cinema 4D, 3D Studio Max, LigthWave 3D e Maya.

Tra le sue potenzialità possiamo ricordare:

- Supporto per una grande varietà di primitive geometriche, incluse le mesh poligonali, le curve di Bézier, superfici NURBS, le metaball e i font vettoriali.
- Integrazione attraverso plugin con un ampio numero di motori di rendering esterni.
- Strumenti di simulazione realistica del moto dei corpi solidi e fluidi.
- Gestione dell'editing video/audio non lineare.
- Caratteristiche interattive attraverso il Blender Game Engine, che permettono la programmazione di applicazioni real-time come la creazione di videogiochi.
- Motore di rendering interno versatile ed integrazione nativa col motore esterno YafRay (un ray-tracer open source).
- Motore di rendering unbiased Cycles disponibile da Blender 2.61, che sfrutta la GPU per il rendering.

- Scripting in python per automatizzare e/o controllare numerosi aspetti del programma.

## 1.2 Interfaccia Grafica

In questo paragrafo verranno presentati gli elementi principali dell'interfaccia di default di Blender, per comprendere meglio quali sono gli obiettivi che ci poniamo di raggiungere in questa tesi.

All'interno della schermata di default di Blender sono presenti quattro regioni principali suddivise in due parti: un header contenente menu e selettori propri della regione e un corpo contenente gli strumenti. In genere l'header si trova in basso nella regione, ma può essere spostato in alto premendo su di esso tasto destro -> Flip to Top o viceversa in basso premendo tasto destro -> Flip to Bottom. Il primo bottone a sinistra dell'header è un selettore, detto editor type, che consente di cambiare il tipo di regione o modificare la disposizione delle regioni nella scena.

In alto a destra, vi è la regione Outliner che mostra l'albero della scena, cioè mostra un elenco degli elementi presenti in quest'ultima e le loro proprietà, dipendenze e relazioni.

Sotto la regione Outliner, vi è la regione Properties, una delle regioni più importanti del programma. Essa racchiude al suo interno varie schede associate a vari contesti come ad esempio gli strumenti per le impostazioni di rendering, particle system, material, texture e tanti altri.

In basso vi è la regione Timeline, ossia la barra temporale, che racchiude alcuni strumenti per controllare le animazioni come il frame corrente, il range (frame iniziale e finale), la riproduzione e altro ancora.

La regione al centro, invece, è la 3D View. Questa regione è chiaramente una riproduzione tridimensionale di un ambiente che non esiste nella realtà, ma dove è possibile individuare un centro, detto origine, e quindi assegnare a qualsiasi oggetto presente nella scena un punto di coordinate X, Y, Z, che indicano dove si trova quell'oggetto rispetto all'origine. I tre assi colorati che si vedono uscire da ogni oggetto Blender sono proprio i tre assi cartesiani: rosso, verde e blu rappresentato rispettivamente X, Y, Z. Nell'header della 3d View vi è un altro selettore molto importante che permette di scegliere la modalità di modifica: le principali sono due (Object Mode e Edit Mode). La Object Mode è utilizzata quando si vuole manipolare (spostare, scalare) un oggetto, mentre la Edit Mode quando si vogliono modificare i dati di un oggetto (vertici, baricentro). Le altre modalità di modifica sono Sculpt Mode, Vertex Paint, Texture Paint e Weight Paint.

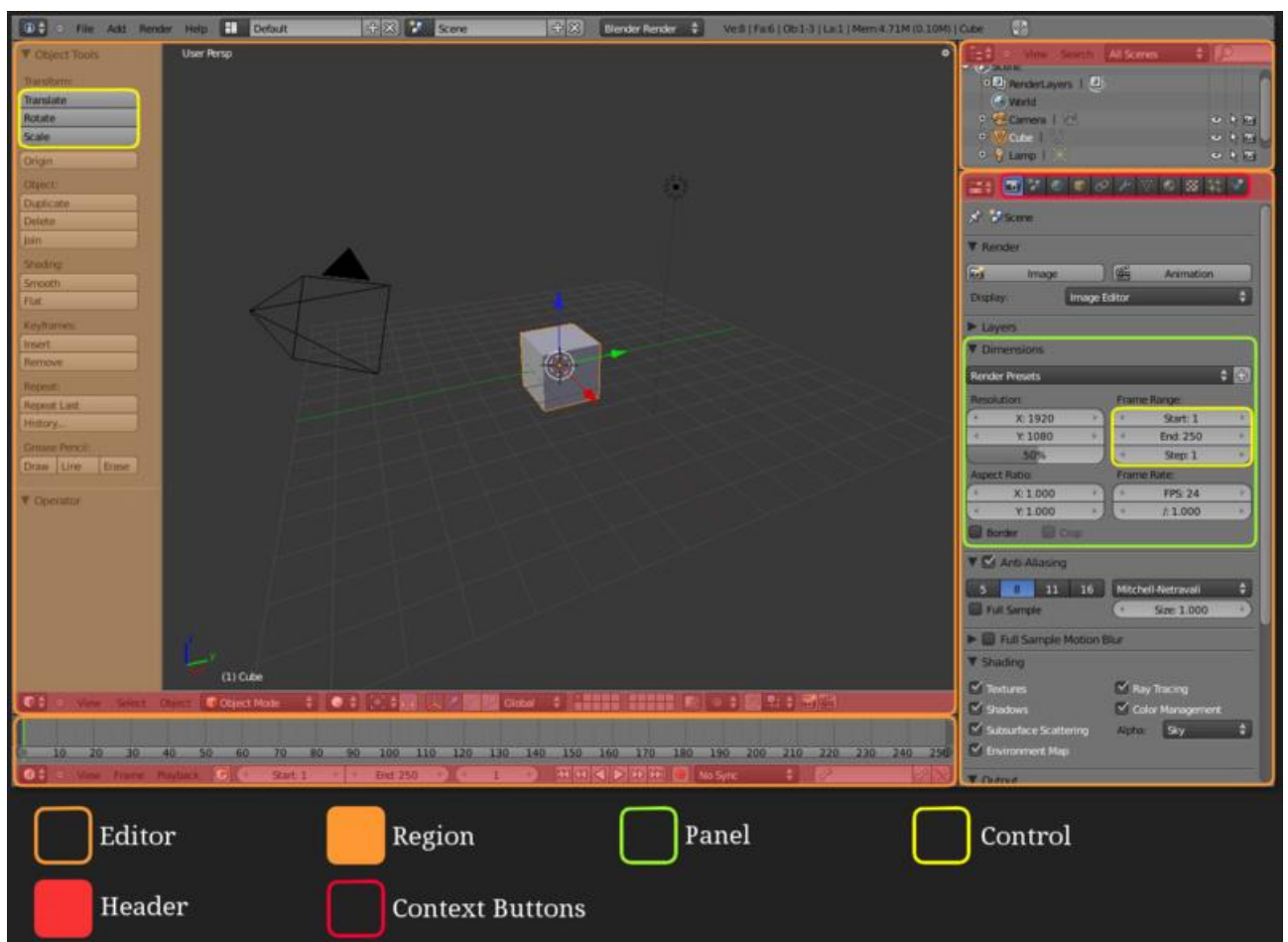
In ogni regione, inoltre, sono presenti numerosi pannelli e bottoni di controllo che danno la possibilità di effettuare specifiche operazioni in quella regione.

La struttura principale dell'interfaccia, quindi, è quella appena descritta; tuttavia è possibile modificarla semplicemente trascinando le varie scene o cambiando il layout di scena attraverso un selettore che si trova nell'header della regione Info.



Un'altra caratteristica di Blender è la possibilità di lavorare con una o più scene, le quali a loro volta possono essere divise in sezioni e sottosezioni che possono essere formate da qualunque immagine di Blender.

In seguito vi è una rappresentazione della schermata di default di Blender.



## 1.3 Blender e Python

Python è un linguaggio di programmazione interpretato e orientato agli oggetti, utilizzabile per molti tipi di sviluppo software. Offre una forte integrazione con altri linguaggi, una libreria standard molto estesa e gira su tutti i principali sistemi operativi (Microsoft Windows, Mac OS X, Gnu/Linux) ma anche su piattaforma Amiga, palmari Palm e cellulari Nokia.

Ciò che fa di Blender un software molto rinomato è proprio la possibilità di aggiungere funzionalità attraverso script Python, poiché dispone al suo interno di un interprete Python pienamente equipaggiato.

Da quando Blender è diventato open-source, la parte che ha usufruito del maggior sviluppo è stata proprio l'API di Python a cui sono stati aggiunti molti nuovi moduli, ma anche l'interfaccia del programma è stata migliorata per aiutare l'utente sotto questo punto di vista.

In particolare vi sono tre regioni molto utili per chi ha bisogno di programmare in Python e sono la Text Editor, in cui si può scrivere il modulo Python da eseguire nel programma, la Python Console, in cui si possono scrivere specifici comandi Python e vedere il risultato direttamente sul software, e la regione Info, in cui viene rappresentato il comando Python rispettivo ad ogni azione che viene effettuata sul programma.

## Capitolo 2

### Obiettivi del progetto

In questo capitolo vengono illustrate le motivazioni che hanno portato alla nascita di questo progetto e gli obiettivi che ci poniamo di raggiungere una volta terminato lo script ma anche le problematiche che abbiamo incontrato durante lo sviluppo.

#### 2.1 Motivazioni

Ciò che ha portato alla nascita di questa tesi è sicuramente un confronto tra il risultato che un utente può ottenere utilizzando Blender e quello che è possibile raggiungere utilizzando altri software dello stesso tipo, in particolare per quanto riguarda la gestione di ecosistemi.

Analizziamo ora a grandi linee due software simili a Blender:

- Vue Esprit
- Visual Nature

Uno dei principali fattori di rilievo, che rende Blender preferibile ai software sopra menzionati, è rappresentato dal costo, spesso non accessibile alla totalità degli utenti.

Confrontiamo ora le differenze dei tre software:

- Blender è in grado di confrontarsi con dati cartografici ed i suoi formati file più diffusi (come lo shp) grazie a strumenti già disponibili (anche se l'integrazione per ora non è automatizzata nell'Add-On da noi sviluppato BEG). Anche Visual Nature supporta ciò, mentre questa funzionalità manca in Vue Esprit.
- Con Blender è possibile creare e modificare modelli 3D ed effettuare render fotorealistici con il motore unbiased interno Cycles. Visual Nature non supporta tali funzionalità, mentre Vue Esprit permette all'utente esclusivamente il render.
- Il materiale prodotto con Blender può essere esportato verso altri software di real-time come Unity o Unreal Engine, così come è possibile farlo anche in Visual Nature. Al contrario, Vue Esprit non lo consente.
- Ultimo, ma non meno importante, fattore è la possibilità di modificare e quindi migliorare Blender attraverso Add-On Python, prerogativa esclusiva di questo software.

L'Add-On quindi nasce per colmare un vuoto tra gli editor di ecosistemi nel mondo open source ed ha, pur con poche features, caratteristiche che lo distinguono dai software a pagamento e che lo rendono immediatamente utilizzabile.

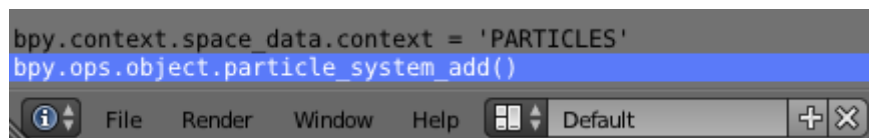
## 2.2 Problematiche riscontrate

La difficoltà maggiore riscontrata durante tutto il lavoro è stata la mancanza di documentazione che ha reso anche la realizzazione di un semplice bottone, o la visualizzazione di una lista di elementi, un ostacolo non facile da superare.

Infatti, essendo Blender un software open source, non ha un'azienda alle spalle come Google, Oracle o altre ancora, che si occupa della documentazione, ma ogni singolo utente che contribuisce al progresso del software può partecipare allo sviluppo della documentazione.

Durante tutta la fase di sviluppo dello script, ogni volta che bisognava implementare una nuova funzionalità, quindi scrivere il codice ed eseguirlo in Blender, i passi da seguire erano 3:

- Come primo passo si cercava la riga Python da inserire nel codice direttamente su Blender. Come accennato nello scorso capitolo, infatti, ogni volta che viene usata una funzionalità messa a disposizione dal software, nella regione Info appare la linea di codice Python che corrisponde all'azione eseguita. Ad esempio se si vuole creare un nuovo Particle System nella regione Info appare:



- Ovviamente ciò era utile solo nel caso in cui servivano funzionalità già esistenti nel software. In caso contrario, come secondo passo, si cercava nella documentazione di Blender, disponibile su [http://www.blender.org/documentation/blender\\_python\\_api\\_2\\_70\\_5/](http://www.blender.org/documentation/blender_python_api_2_70_5/), alcune funzioni utili alla causa. La ricerca, però, spesso non portava ai risultati sperati proprio perché la documentazione è abbastanza povera.
- Come ultimo passo, si cercavano sul web degli Add-On già esistenti che implementavano funzioni simili a quello che serviva a noi. In questo modo abbiamo aggirato anche il problema della documentazione povera. Tuttavia la ricerca era spesso lunga e portava via, a volte, anche alcuni giorni.

## Capitolo 3

### Installare Python

In questo capitolo viene illustrato come configurare l'ambiente per accedere ai moduli standard di Python.

#### 3.1 Configurazione ambiente

Per poter usufruire appieno delle funzionalità che offre Blender, occorre un'installazione completa e funzionante di Python, disponibile su [www.python.org/downloads/](http://www.python.org/downloads/).

Una volta scaricato e installato Python, bisogna definire una variabile d'ambiente PYTHONPATH, in quanto Blender deve essere a conoscenza di dove è stato installato Python. Prima di questa operazione, però, bisogna verificare su [www.blender.org](http://www.blender.org) qual è l'esatta versione di Python incorporata in Blender, per evitare problemi di compatibilità.

Nei prossimi paragrafi, verranno discussi i diversi modi di installazione e configurazione di Python in base al sistema operativo installato sulla macchina.

### 3.1.1 Linux

Blender 2.7x ha bisogno della versione di Python 3.4. Per averne un'installazione completa su Linux bisogna innanzitutto scaricare i file sorgente (Python-3.4.0.tar.bz2) dal link fornito in precedenza, ed estrarne il contenuto. Successivamente, da terminale, bisogna posizionarsi nella cartella dove è stato estratto il file zip e digitare:

```
./configure --prefix="/opt/py34"
```

Si noti che `"/opt/py34"` è solo un esempio.

Visto che di default, Python 3.x non compila zlib, ma molti addons di Blender ne hanno bisogno, è inoltre necessario installare zlib e zlib-dev libraries nel sistema e de commentare la seguente riga:

```
"zlib zlibmodule.c -I$(prefix)/include -L$(exec_prefix)/lib -lz"
```

che si trova in Modules/Setup nella directory dove vi sono i file sorgente.

Infine bisogna digitare da terminale:

```
make; sudo make install
```



In definitiva i passi da compiere per avere un'installazione completa e funzionante di Python su linux sono:

## Linux/Generic

```
# get source
wget http://www.python.org/ftp/python/3.4.0/Python-3.4.0.tgz
tar -xvf Python-3.4.0.tgz

# make install
cd Python-3.4.0
./configure --prefix=/opt/py34
make
sudo make install
```

## Linux/Ubuntu

```
# get required packages for build
sudo apt-get install build-essential libncursesw5-dev libreadline5-dev libssl-dev lib-
bgsdbm-dev libc6-dev libsqlite3-dev tk-dev

# get source
wget http://www.python.org/ftp/python/3.4.0/Python-3.4.0.tgz
tar -xvf Python-3.4.0.tgz

# make install
cd Python-3.4.0
./configure
make
sudo make altinstall

# make 3.4 the default python system wide (number 1 at the end stays there)
sudo update-alternatives --install /usr/bin/python python /your/path/to/python3.4

# ensure various versions of python play nice with each other
sudo update-alternatives --config python
```

### 3.1.2 Mac OS X

Nel sistema operativo Mac OS X è già presente l'installazione di Python 2.7. Ma come detto in precedenza Blender 2.7x ha bisogno della versione di Python 3.4 o superiore.

Bisogna quindi eseguire alcune operazioni per aggiornare la versione di Python pre installata. Come primo passo bisogna ovviamente scaricare i file sorgente dell'ultima versione di Python dal link fornito all'inizio di questo capitolo e installarli.

In seguito all'installazione, ciò di cui si ha ulteriormente bisogno è:

- Una cartella MacPython 3.4 nella nostra cartella Applications. Qui si trovano l'IDLE (l'ambiente di sviluppo, che è una parte standard di distribuzioni Python), il PythonLauncher che gestisce l'esecuzione degli script Python, e lo strumento "Build Applet" che ti permette di impacchettare gli script Python come applicazioni a se stanti sul tuo sistema.
- Un framework `/Library/Frameworks/Python.framework` che comprende le librerie e gli eseguibili Python. L'installer aggiunge questa posizione al Path della shell.

Bisogna infine creare un file `~/.MacOSX/environment.plist` per configurare la variabile d'ambiente `PYTHONPATH` (per i dettagli leggere il Documento Tecnico QA1067 di Apple).

Si noti inoltre che se si sceglie di installare una versione più recente di Python da [python.org](https://python.org), si hanno due diverse ma funzionali installazioni di Python sul computer, quindi sarà importante che gli usi siano coerenti con ciò che si vuole fare.

### 3.1.3 Windows

Windows, a differenza di Mac OS X, non ha una pre installazione di Python. Tuttavia il processo di configurazione è molto simile. Bisogna infatti scaricare e installare l'ultima versione di Python e successivamente configurare la variabile d'ambiente `PYTHONPATH`.

Ciò che bisogna fare è, innanzitutto, trovare la cartella dove vi è l'eseguibile Python. In seguito bisogna andare nelle impostazioni di sistema avanzate (in Pannello di controllo\Sistema e sicurezza\Sistema) e aggiungere alla variabile d'ambiente Path, il percorso della cartella trovata in precedenza.

Un esempio della variabile Path può essere:

```
C:\WINDOWS\system32;C:\WINDOWS;C:\Python33
```

Dove `C:\WINDOWS\system32;C:\WINDOWS;` sono impostate di default da Windows.

# Capitolo 4

## Struttura Add-On

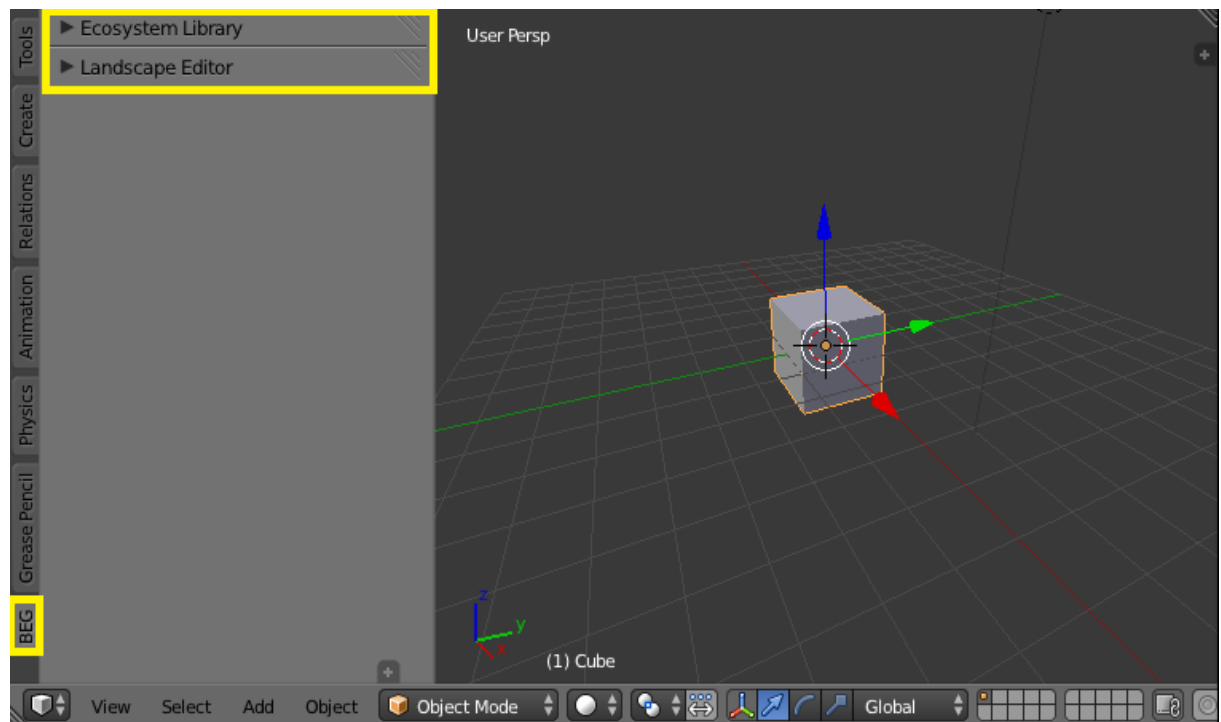
In questo capitolo vengono descritte le funzionalità aggiunte su Blender dallo script. In particolare il primo paragrafo viene dedicato a descrivere i cambiamenti sull'interfaccia che vengono apportati quando viene installato l'Add-On, nei paragrafi successivi si illustrano le principali funzionalità aggiunte e l'ultimo è dedicato a spiegare come installare l'Add-On.

### 4.1 Interfaccia

Una volta installato l'Add-On, esso apporterà delle modifiche nella regione della view 3D. In particolare crea nel menu Tools un nuovo sottomenu chiamato BEG, e in esso due pannelli:

- Ecosystem Library
- Landscape Editor

La trattazione delle funzionalità dei due pannelli avverrà nel seguente paragrafo.



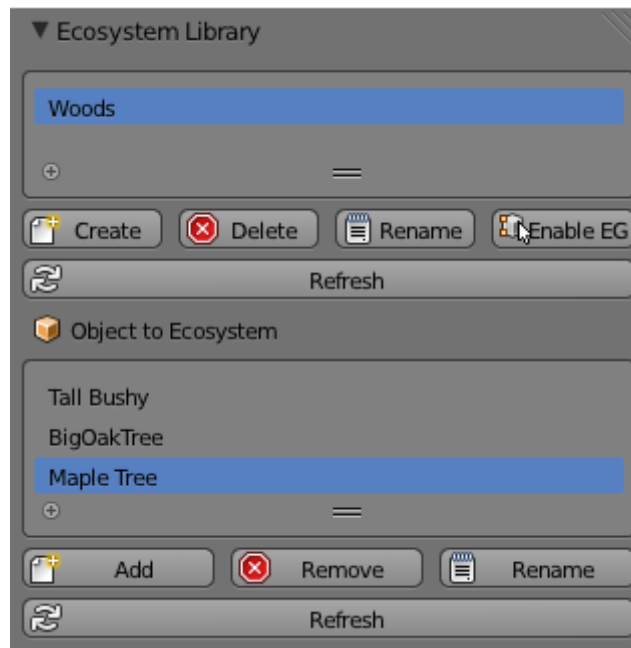
Questi due pannelli vengono generati dalle seguenti classi Python:

```
class EcosystemLibrary(bpy.types.Panel):  
    bl_category = "BEG"  
    bl_label = "Ecosystem Library"  
    bl_space_type = "VIEW_3D"  
    bl_region_type = "TOOLS"  
    bl_options = {'DEFAULT_CLOSED'}
```

```
class LandscapeEditor(bpy.types.Panel):  
    bl_category = "BEG"  
    bl_label = "Landscape Editor"  
    bl_space_type = "VIEW_3D"  
    bl_region_type = "TOOLS"  
    bl_options = {'DEFAULT_CLOSED'}
```

## 4.2 Ecosystem Library

Il pannello Ecosystem Library si presenta così all'utente:



In questo caso, e durante tutta l'esposizione della tesi, vengono usati come esempio un unico ecosistema, il bosco, e tre diverse specie, un cespuglio, una quercia e un melo.

Ecosystem Library ha la funzionalità di creare una libreria, che altro non è che un ecosistema, utilizzabile nel secondo pannello.

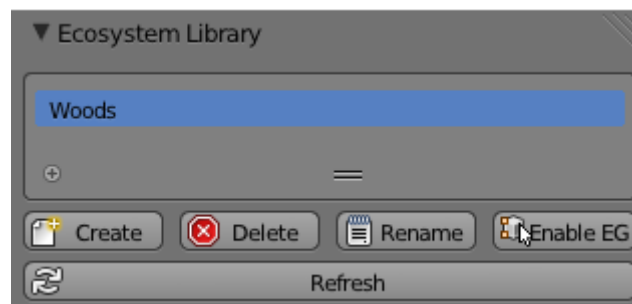
Possiamo suddividere il pannello in due parti:

- Nella prima parte vi è una lista di Ecosistemi con i relativi pulsanti di creazione, modifica ed eliminazione.

- Nella seconda parte vi è la lista di oggetti appartenenti a quell'Ecosistema e, anche in questo caso, i relativi pulsanti di creazione, modifica ed eliminazione.

### 4.2.1 Lista Ecosistemi

Esaminiamo ora nel dettaglio tutti i pulsanti del primo sotto-pannello e mostriamo per ognuno la definizione della classe attraverso la quale sono stati creati.



- Il primo bottone della prima fila è il pulsante Create che crea un nuovo ecosistema e gli assegna di default il nome 'Ecosystem'.

```
class New_Group(bpy.types.Operator):  
    bl_idname = "new_group.operator"  
    bl_description = "Create New Group"
```

- Il secondo bottone è il pulsante Delete che elimina l'ecosistema selezionato. In Blender l'oggetto selezionato viene evidenziato in blu, come mostrato nell'immagine precedente.

```
class Remove_Group(bpy.types.Operator):  
    bl_idname = "remove_group.operator"  
    bl_description = "Remove Group"
```

- Il terzo bottone è il pulsante Rename che non fa altro che modificare il nome dell'ecosistema selezionato.

```
class Rename_Group(bpy.types.Operator):  
    bl_idname = "rename_group.operator"  
    bl_description = "Rename Group"
```

- Il quarto bottone, che è uno dei più importanti di questo sotto-pannello, è il pulsante Enable EG, che crea una nuova scena chiamata 'ECO-SYS', nella quale verranno trasferiti gli oggetti che vengono aggiunti ad un ecosistema, per rimuoverli dalla scena principale.

```
class New_Scene(bpy.types.Operator):  
    bl_idname = "set_scene.operator"  
    bl_description = "Create ECOSYS Scene"
```

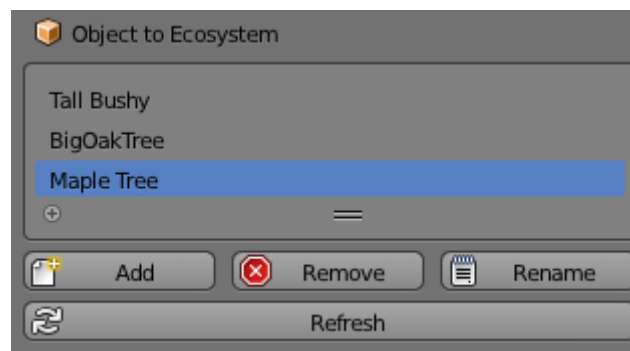
- L'unico bottone sulla seconda fila, invece, è il pulsante Refresh, che aggiorna la lista degli ecosistemi creati.

```
class Refresh_Group(bpy.types.Operator):  
    bl_idname = "refresh_groups.operator"  
    bl_description = "Refresh Groups"
```



## 4.2.2 Object to Ecosystem

In questo paragrafo esamineremo dettagliatamente le funzioni che svolgono tutti i pulsanti del sotto-pannello Object To Ecosystem, con le relative definizioni delle classi utilizzate.



- Il primo bottone della prima fila è il pulsante Add, che aggiunge l'oggetto selezionato nella scena (nella view 3D), all'ecosistema selezionato nella lista del primo sotto-pannello. Appena l'oggetto è stato aggiunto all'ecosistema, viene trasferito nella scena 'ECOSYS'.

```
class Add_Object(bpy.types.Operator):  
    bl_idname = "add_object.operator"  
    bl_description = "Add Selected Object to Selected Group"
```

- Il secondo bottone della prima fila è il pulsante Remove, che rimuove l'oggetto selezionato dall'ecosistema di appartenenza. Non appena è stato rimosso, l'oggetto viene ripreso dalla scena 'ECOSYS' e riportato nella scena principale.

```
class Remove_Object(bpy.types.Operator):  
    bl_idname = "remove_object_group.operator"  
    bl_description = "Remove Selected Object from Selected Group"
```

- Il terzo bottone della prima fila è il pulsante Rename, che modifica il nome dell'oggetto selezionato.

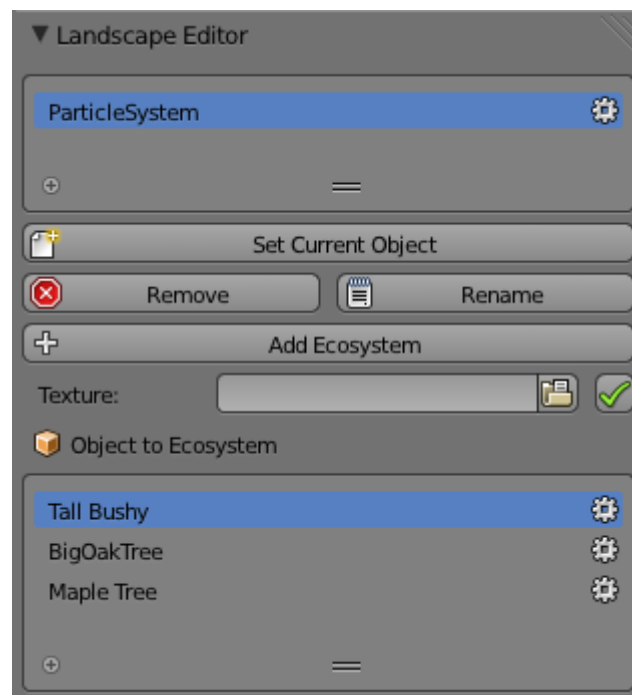
```
class Rename_Object(bpy.types.Operator):  
    bl_idname = "rename_object.operator"  
    bl_description = "Rename Object"
```

- L'unico bottone sulla seconda fila, invece, è il pulsante Refresh, che aggiorna la lista degli oggetti appartenenti ad un ecosistema.

```
class Refresh_Objects_Group(bpy.types.Operator):  
    bl_idname = "refresh_objects_group.operator"  
    bl_description = "Refresh List of Objects"
```

## 4.3 Landscape Editor

Il pannello Landscape Editor si presenta così all'utente:



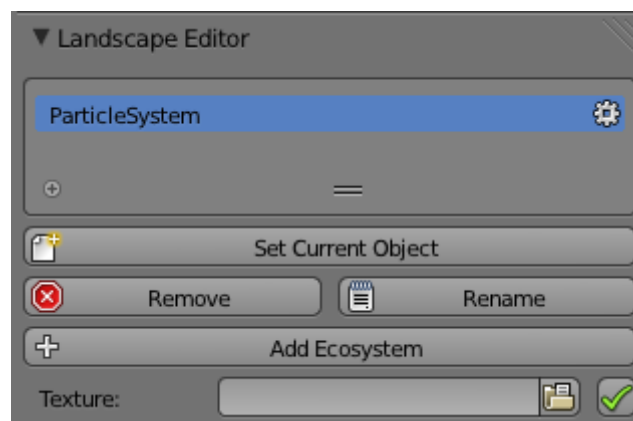
Anche in questo caso, come nel precedente, per facilitare la trattazione suddividiamo il pannello in due sotto-pannelli:

- Nel primo sotto-pannello è presente la lista dei sistemi particellari di un oggetto, con i relativi pulsanti di creazione, eliminazione e modifica del nome. Oltre a questi vi sono anche due altri pulsanti che descriveremo in seguito.

- Il secondo sotto-pannello, invece, è la copia di ‘Object to Ecosystem’ descritto nel precedente paragrafo, a cui è stato aggiunto in-line un altro pulsante.

### 4.3.1 Lista di Particle System

Esaminiamo ora nel dettaglio tutti i pulsanti del primo sotto-pannello e mostriamo per ognuno la definizione della classe attraverso la quale sono stati creati.



- Il pulsante Set Current Object crea un nuovo Particle System, assegnandogli di default il nome ‘Empty Ecosystem’, e lo assegna all’oggetto selezionato in scena. Il Particle System creato sarà di tipo ‘Hair’, ossia verranno generate solo particelle statiche che si utilizzano per rappresentare capelli, erba ecc.

```
class New_PS(bpy.types.Operator):  
    bl_idname = "new_ps.operator"  
    bl_description = "Create New Particle System"
```

- Il pulsante Remove elimina il Particle System selezionato nella lista sopra di esso.

```
class Remove_PS(bpy.types.Operator):  
    bl_idname = "remove_ps.operator"  
    bl_description = "Remove Particle System"
```

- Il pulsante Rename rinomina il Particle System selezionato.

```
class Rename_PS(bpy.types.Operator):  
    bl_idname = "rename_ps.operator"  
    bl_description = "Rename Particle System"
```

- Il pulsante Add Ecosystem è molto importante, perché è quello che fa da collante tra il primo e il secondo pannello. Esso, infatti, prende l'ecosistema che si è creato precedentemente nel pannello superiore e lo associa al Particle System selezionato. Una volta che viene associato l'ecosistema, anche il Particle System prenderà il nome di quest'ultimo.

```
class Add_Group_To_PS(bpy.types.Operator):  
    bl_idname = "add_group_ps.operator"  
    bl_description = "Add Selected Group to Selected Particle System"
```

- In basso al sotto-pannello vi è una String property, ossia una proprietà associata alla scena, che altro non è che una stringa visibile nel Text Box. Il pulsante affianco al Text Box è un pulsante che apre il File Browser. L'utente ha ora la possibilità di scegliere un'immagine nel proprio computer, il cui path verrà salvato nel Text Box.

```
bpy.types.Scene.path = bpy.props.StringProperty(description="simple file path", maxlen=1024, subtype='FILE_PATH')
```

- L'ultimo pulsante in basso a destra è un segno di spunta verde. La sua funzione è duplice: infatti ha il compito di prendere il path dal Text Box e creare un'Image texture con l'immagine che andrà a cercare in quel path e associare tale texture al Particle System.

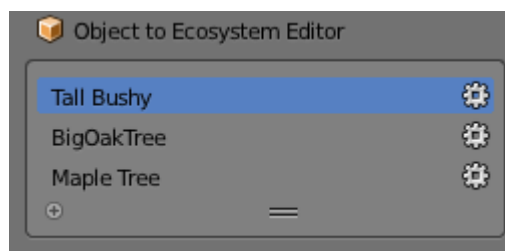
```
class Load_Map(bpy.types.Operator):  
    bl_idname = "load_map.operator"  
    bl_description = "Select Distribution Map"
```

- Infine vi è un pulsante in-line nella lista dei Particle System che premuto fa apparire un popup, il quale permette all'utente di inserire un numero che sarà il numero totale di particelle nell'ecosistema.

```
class SetNumberParticles(bpy.types.Operator):  
    bl_idname = "set_count_object.operator"  
    bl_description = "Set Object's number"
```

### 4.3.2 Object to Ecosystem Editor

Object to Ecosystem Editor è simile al pannello che vi è in Ecosystem Library, ma in esso è possibile editare una proprietà degli oggetti nell'ecosistema, che ora vedremo in dettaglio.



- L'unico pulsante presente nel pannello è l'ingranaggio sulla destra di ogni elemento. Quando si preme quel pulsante appare un menu popup in cui l'utente può inserire un numero. Questo numero è la percentuale con cui l'oggetto è presente nell'ecosistema rispetto agli altri oggetti.

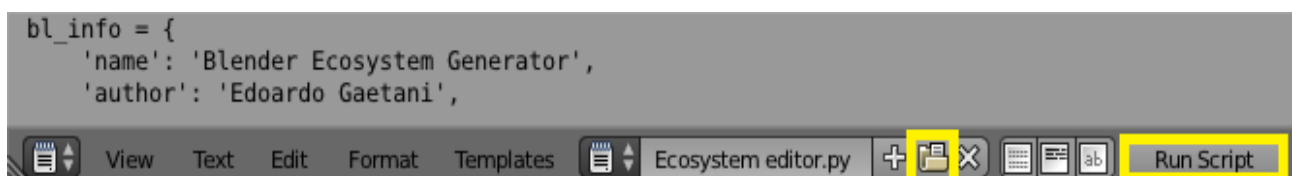
```
class Set_Count(bpy.types.Operator):  
    bl_idname = "set_count.operator"  
    bl_description = "Set Count Object"
```

Si noti che questo pannello non è sempre visibile, infatti scompare quando la lista dei Particle System è vuota. Questo accorgimento è dovuto al fatto che non ha senso settare la percentuale di un oggetto se non esistono sistemi particellari che lo contengono.

## 4.4 Installare l'Add-On

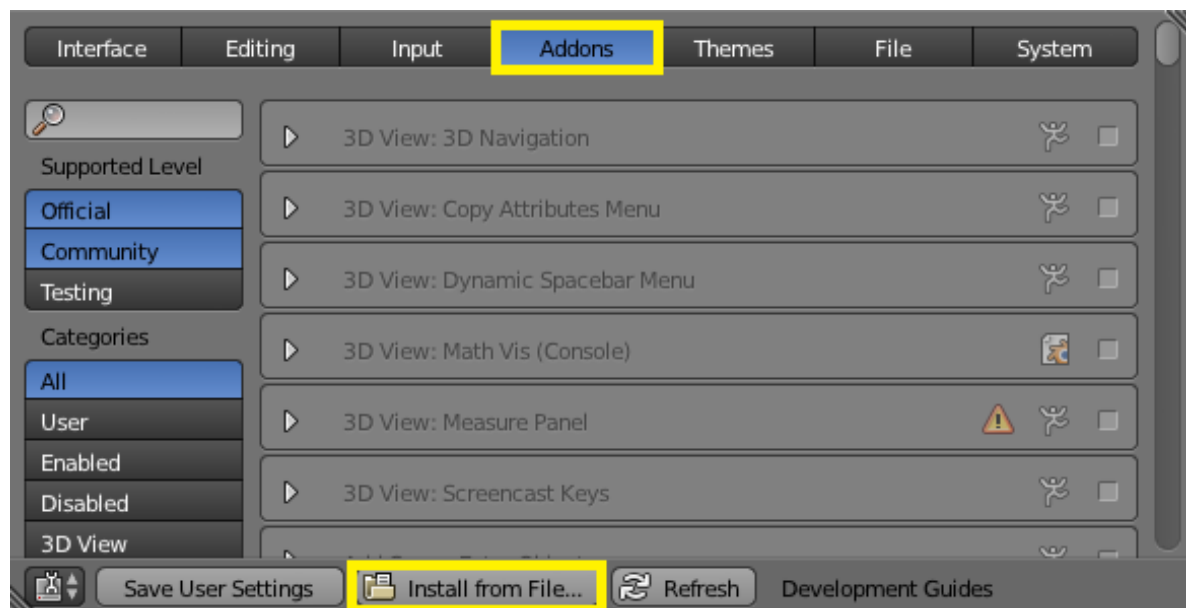
È possibile utilizzare l'Add-On in due differenti modalità:

- La prima è in modalità provvisoria, ossia che vale solo per la sessione corrente di Blender. Per fare ciò basta aprire Blender, andare nella regione Text Editor, cliccare sul pulsante raffigurato da una cartella che aprirà il File Browser, scegliere il file dell'Add-On che avrà estensione .py, e cliccare Run Script. Ora, fino a che non verrà chiuso Blender, sarà possibile usare l'Add-On.

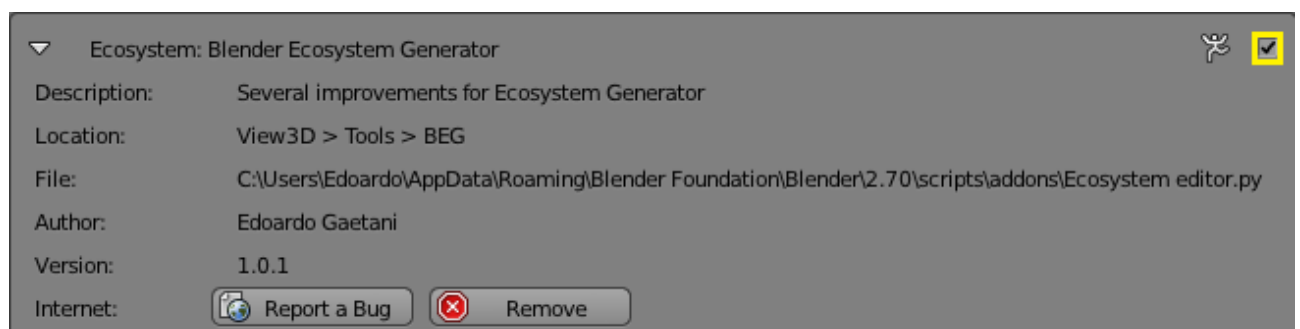




- La seconda è in modalità permanente e richiede quindi l'installazione vera e propria dell'Add-On. Per fare ciò bisogna andare nella regione User Preferences, nel suo sottomenu Addons e cliccare 'Install from file...'.



Ora bisognerà selezionare l'Add-On installato e mettere il segno di spunta su di esso per renderlo attivo in Blender ogni volta che il software viene avviato.



## Capitolo 5

### Conclusioni

Abbiamo iniziato questo lavoro di tesi con un'introduzione al software Blender, presentando le caratteristiche che hanno fatto di esso un software molto importante nella 3D Graphics, e illustrando la sua interfaccia grafica (cfr. capitolo 1).

Quindi abbiamo presentato le motivazioni che hanno portato alla nascita di questo progetto, e tutte le difficoltà riscontrate durante il percorso (cfr. capitolo 2).

In seguito abbiamo illustrato come configurare l'ambiente per avere un'installazione completa di Python (cfr. capitolo 3).

Nel quarto capitolo, infine, abbiamo presentato il prodotto e illustrato tutte le sue funzionalità.

Nel complesso, il progetto realizzato è un ottimo punto di partenza e risulta sufficientemente accurato e molto semplice da usare per un utente che utilizza Blender per la gestione di ecosistemi. È stato anche raggiunto un ottimo livello di affidabilità dell'Add-On attraverso una gestione molto accurata di possibili casi particolari che un utente potrebbe ritrovarsi.

Nonostante ciò, comunque, l'Add-On ha sicuramente dei margini di miglioramento, soprattutto per quanto riguarda l'editor degli ecosistemi.

Infatti nella creazione e nella gestione di un ecosistema, l'utente ha tutti i mezzi che gli servono per creare un terreno e popolarlo con oggetti che possono rappresentare diverse specie di alberi o altro ancora.

Nel pannello di editor, invece, l'utente ha, ad esempio, la possibilità di caricare una mappa di distribuzione, che altro non è che un'immagine in scala di grigi, da associare ad un sistema particellare, ma non ha la possibilità di modificare tale immagine direttamente dall'Add-On. Quindi questa è sicuramente una miglioria che in futuro potrebbe essere fatta.

In generale, comunque, all'Add-On potrebbero essere aggiunte maggiori features e dovrebbe avere una maggiore integrazione con strumenti open già disponibili per Blender.

In ogni caso, i risultati raggiunti sono sicuramente soddisfacenti e gli obiettivi iniziali sono stati in linea di massima rispettati.

Il codice dell'Add-On e la versione pdf della tesi sono disponibili al link <http://goo.gl/TbXW13>.



# Bibliografia

- [ 1 ]      Sito ufficiale di Blender. URL: <http://www.blender.org/features/2-71/>
- [ 2 ]      Documentazione ufficiale Blender/Python. URL: [http://www.blender.org/documentation/blender\\_python\\_api\\_2\\_70\\_5/](http://www.blender.org/documentation/blender_python_api_2_70_5/)
- [ 3 ]      Sito del manuale di Blender. URL: <http://wiki.blender.org/index.php>
- [ 4 ]      Sito degli Add-On per Blender. URL: <http://blenderaddonlist.blogspot.it/>
- [ 5 ]      Add-On 'Selection Sets'. URL: <http://www.blendernation.com/2014/07/14/blender-add-on-selection-sets-version-0-2/>
- [ 6 ]      Add-On 'Object Group Editor'. URL: <http://blenderaddonlist.blogspot.it/2013/11/addon-group-editor.html>

# Ringraziamenti

*Al termine della realizzazione di questo progetto mi sento di ringraziare il professor Andrea Vitaletti e il dottor Emanuel Demetrescu per avermi dato la possibilità di approfondire le mie conoscenze nell'ambito della Computer Graphics.*

*Ringrazio tutta la mia famiglia per avermi dato la possibilità di affrontare questa esperienza universitaria con il loro appoggio morale.*

*Ringrazio i miei compagni di corso con cui ho condiviso le difficoltà di questa carriera universitaria ma anche numerosi momenti divertenti.*

*Ringrazio i miei amici tricasini, compagni di tantissime serate speciali.*

*Ringrazio i miei amici "scasciati" che mi hanno messo i bastoni fra le ruote durante tutto lo sviluppo della tesi, facendomi giocare a calcetto oltre l'inverosimile.*

*Ringrazio infine la mia dolce metà con cui ho condiviso tre anni stupendi a Roma e che ho stressato un po' troppo chiedendo consigli di ogni tipo durante l'intera stesura della tesi.*