

PROJECTE FI DE CURS

IMPRESSIO 3D

Introducció al llenguatge Python aplicat a Blender o com automatitzar processos. Projecte de 30 hores.

Institució:	CIFO La Violeta
Professor:	El molt honorable Joan Masdemont i Fontàs
Jo:	Jordi Gorbe i Neira
Data:	26/07/2019

Index:

1. Introducció al treball	3
2. Introducció a Python i Blender	4
3. Software i Maquinaria	6
4. Fases del projecte	9
5. Projecte	16
6. Conclusions	26
7. Annexos	26
8. Bibliografia	26

1. Introducció al treball

L'objectiu del treball és automatitzar processos repetitius perquè ho faci la màquina i no tu. Així només has de concentrar-te en el disseny i els processos que no podràs automatitzar amb el software.

La idea surt ja des de el primer dia de classe on el professor del curs, Joan Masdemont i Fontàs, ens va parlar dels projectes duts a terme en el curs anterior. La casuística dels projectes estava dividida en diversos grups, els més importants eren el disseny de les peces mitjançant interfície gràfica i l'automatització del disseny mitjançant programació. Jo vinc del món del disseny (sóc arquitecte) i del món de la programació (actualment estic estudiant programació multiplataforma) així doncs, tenia molt clar que estava més interessat en l'automatització del procés del disseny que en el disseny per se.

El disseny de la peça es farà mitjançant Blender 2.79 que és un programa basat en Python. Blender disposa d'una API que té un subllenguatge molt simple però extens que donarà les ordres a Blender. A més d'aquest subllenguatge necessitarem saber les ordres bàsiques de Python, que en el meu cas les desconeixia, com ara loops, strings, int, o boolean.

La peça en si no és important, per tant, ens limitarem a fer un objecte molt bàsics amb una variació cada cop. Agafarem un projecte ja fet en el curs passat i el modificarem i millorarem. Un clauer personalitzat amb els noms dels clients.

2. Introducció a Python i Blender

2.1 Python

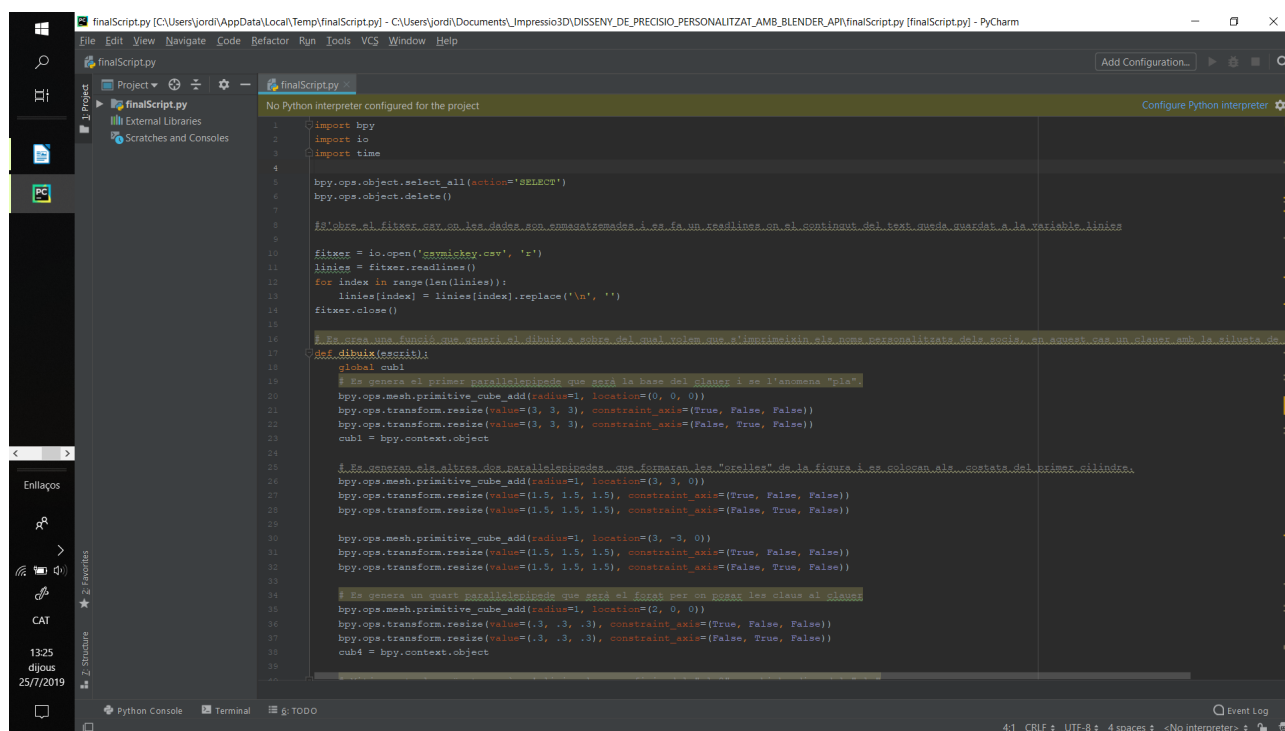
Python és un llenguatge de programació d'alt nivell i propòsit general molt utilitzat. Va ser creat per Guido van Rossum l'any 1991. La seva filosofia de disseny busca llegibilitat en el codi i la seva sintaxi permet als programadors expressar conceptes en menys línies de codi del que seria possible en llenguatges com C. També proveeix estructures per permetre programes més entenedors tant a petita com a gran escala.

Python suporta diversos paradigmes de programació, incloent-hi programació orientada a objectes, imperativa i també funcional o procedimental. Presenta un sistema dinàmic i una gestió de la memòria automàtica i té una gran i exhaustiva biblioteca estàndard.

Com altres llenguatges de programació dinàmics, Python és usat sovint com a un llenguatge script, però també es fa servir en una àmplia gamma de contextos no-script. Utilitzant eines desenvolupades per tercers com Py2exe, cx_Freeze o Pyinstaller, el codi Python pot ser reduït a programes executables independents. Existeixen intèrprets de Python per molts sistemes operatius diferents.

Per la gent que com jo ve del món C, el sagnat és la característica més visible del Python. Al contrari que en altres llenguatges, és obligatori i s'utilitza per delimitar blocs. Un augment en el sagnat, si va precedit per una declaració correcta, significa l'inici d'un nou bloc; la disminució significa el final del bloc actual.

L'editor de Python 3.7 utilitzat és PyCharm Community Edition 2019.1.3.



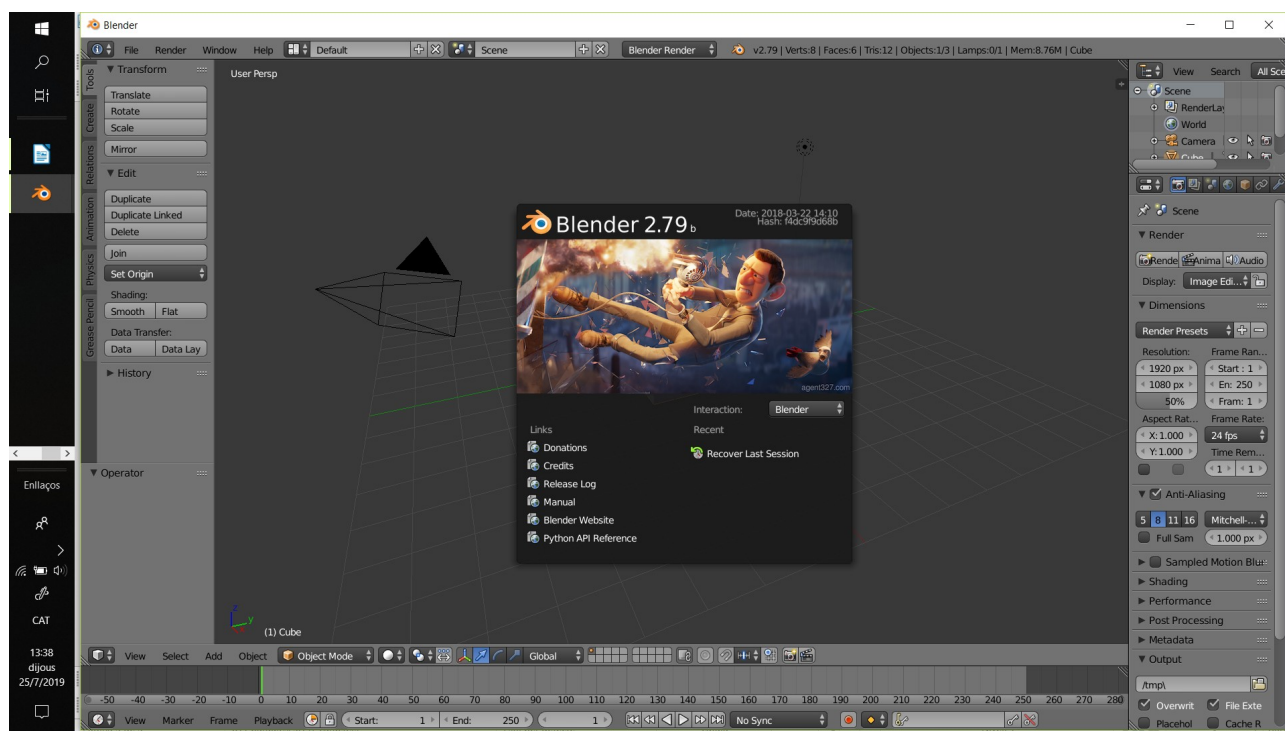
2.2 Blender

Blender és un programari lliure multiplataforma, dedicat a l'edició tridimensional. Això inclou eines per al modelatge d'objectes, l'edició de materials, l'animació, el render, compositing amb un sistema de nodes i un motor de jocs, a més de funcionalitats complementàries com l'edició no lineal de vídeo.

Disposa d'una interfície gràfica d'usuari molt peculiar, que ha rebut sovint crítiques per ser poc intuïtiva, ja que no es basa en el sistema clàssic de finestres. Tot i així, té grans avantatges, com la configuració personalitzada de la distribució dels menús i de les vistes de càmera.

Té, entre d'altres, les característiques següents:

- Multiplataforma, lliure, gratuït i amb una mida d'origen molt petita, entre 10 i 20 MB en funció del sistema operatiu.
- Capacitat per a una gran varietat de primitives geomètriques, inclou corbes, malles poligonals, buits, NURBS, metaballs. Característiques interactives per a jocs com detecció de col·lisions, recreacions dinàmiques i lògica.
- Juntament amb les eines d'animació inclou cinemàtica inversa, deformacions per armadura o quadrícula, assignació de càrregues a vèrtexs i partícules estàtiques i dinàmiques. Edició elemental d'àudio i sincronització de vídeo.
- Possibilitat de renderitzat intern versàtil i integració externa amb el potent traçador de raigs o "raytracer" lliure YafRay.
- **Llenguatge Python per a automatitzar o controlar diverses feines.**

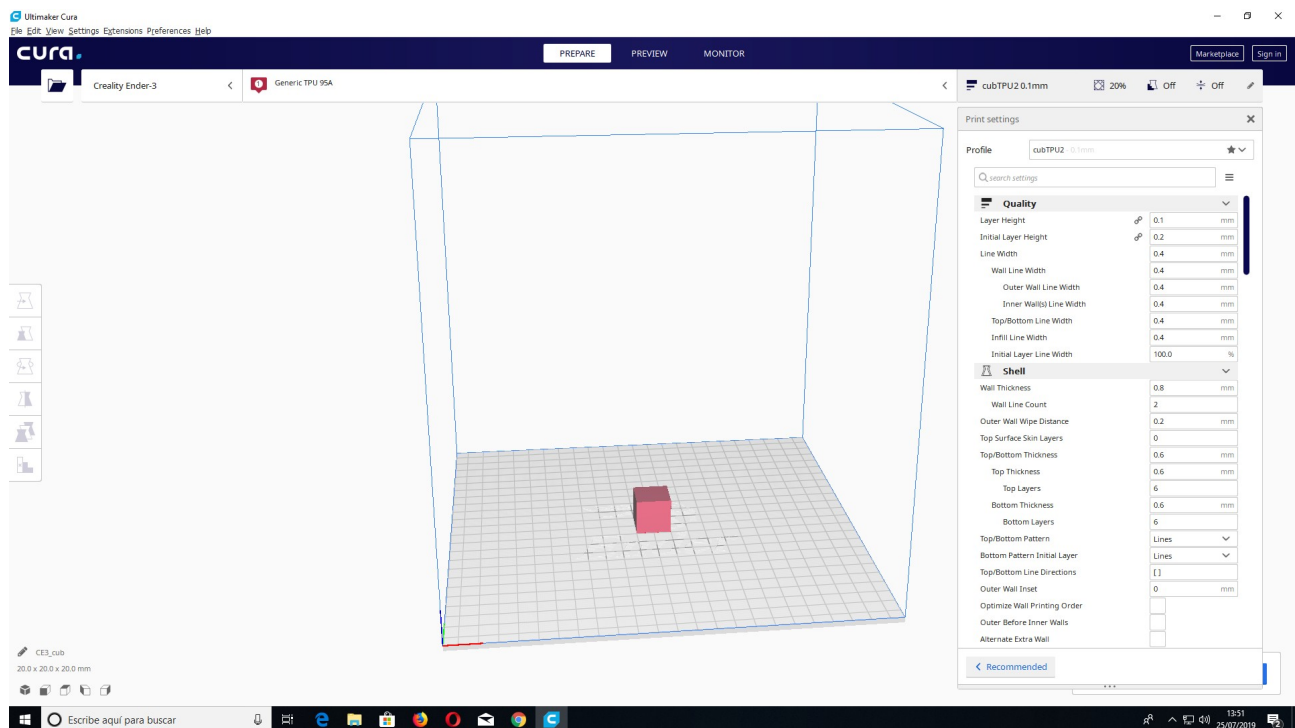


3. Software i Maquinari

3.1 Software

Ultimaker Cura 4.1.0. És un software de tractament de models per el preproces de impressió 3D. Et genera un Gcode.

Ultimaker Cura funciona tallant el fitxer del model de l'usuari en capes i generant un Gcode específic de la impressora. Un cop acabat, el Gcode es pot enviar a la impressora per a la fabricació de l'objecte físic. El programari de codi obert és compatible amb la majoria de les impressores 3D d'escriptori que poden treballar amb fitxers en els formats 3D més habituals, com ara STL, OBJ, X3D, 3MF, així com formats de fitxers d'imatges com BMP, GIF, JPG i PNG.



3.2 Maquinària

Ovbiament un Pc amb Windows 10 i Impressora 3D model **Crealty 3D Ender-3**. El se cost és de menys de 200€. Aquest model d'impressora 3D Open Source, permet imprimir de forma assequible nombrosos tipus de filament: PLA, ABS, PETG, flexible, entre altres en un volum d'impressió per sobre de l'estàndar.

Característiques:

- Impressora 3D Open source.
- Materials: PLA, ABS, PETG, Flexible 95A, composites, etc.
- Silenciosa.
- Volum d'impressió per sobre de l'estàndar.
- Econòmica.
- Fàcil anivellació.
- Extrusor optimitzat.
- Subministrament d'energía d'alta qualitat i segura.
- Disposa de llit calent.

Especificacions Tècniques:

- Tipus d'extrusió: FDM (Sistema Bowden).
- Volum de construcció: 220 x 220 x 250 mm.
- Diàmetre filament: 1,75 mm.
- Diàmetre embocadura: 0.4 mm.
- Gruix de capa: 0.1 – 0.35 mm.
- Precisió: +/-0.1 mm.
- Temperatura màxima extrusor: 255 °C.
- Temperatura màxima de llit: 110 °C.
- Velocitat màxima de trajecte: 180 mm/s.
- Formats model 3D: STL, OBJ, G-Code
- Connectivitat: Targeta SD y cable USB.
- Sistema operatiu compatibles: Windows/ Mac/ Linux.
- Xassís: Perfils V-Slot d'alumini.
- Dimensions de la impressora 3D: 440 x 410 x 465 mm.

- Pes: 8.6 Kg.
- Pes net: 10 Kg.
- Embalatja Dimensiones: 600 x 350 x 160 mm.
- Entrada: AC 100-265V 50-60Hz.
- Sortida DC 24V 15A 360W.
- Endoll de la UE.



Magnetic Bed



Mean Well Power 24V

4. Fases del Projecte

Fase 1_Introducció a Python

Com no tenia ni repajolera idea de Python el primer que vaig fer, amb el beneplàcid del professor Joan Masdemont i Fontàs, va ser buscar en youtube algun curs ràpid d'introducció a Python per poder entendre el codi mare que utilitzarem pel nostre disseny.

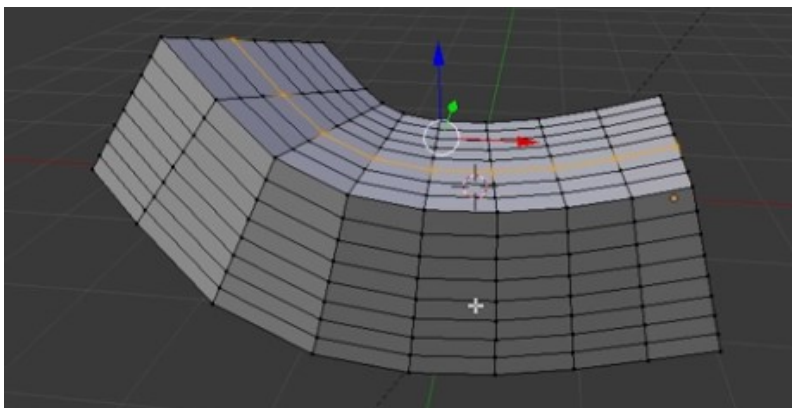
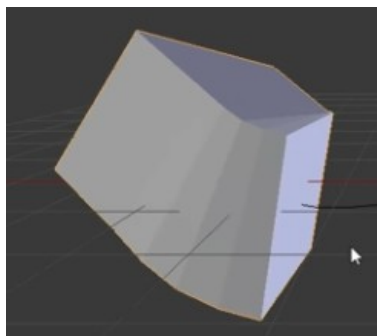
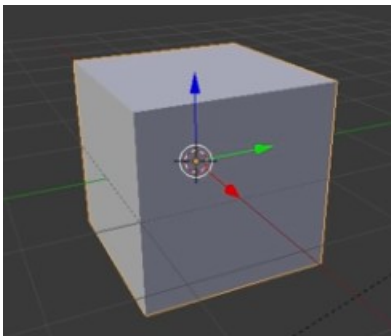
El primer resultat de la cerca al youtube va ser "**Python Tutorial for Beginners [Full Course] 2019**". És un curs de 6 hores impartit per en Mosh Hamedani.

Link: <https://www.youtube.com/watch?v=uQrJ0TkZlc>.

Sintaxi del llenguatge: my Python summary notes. [Veure annexe 1]

Fase 2_Introducció a Blender

Blender és un programa pensat pel disseny multimèdia. La diferència amb els programes clàssics de disseny assistit per ordinador seria conceptual. En un programa Cad l'objecte final és una suma d'objectes 3D que has anat dissenyant. En canvi, en Blender podríem obtenir el mateix resultat agafant un cub i deformant-lo fins a arribar al mateix resultat.



Fase 3_Introducció a l'API de Blender

L'API de Blender ens ajudarà a automatitzar processos. Primer haurem d'aprendre els conceptes bàsics sobre com utilitzar Python per controlar Blender i millorar la forma de fer el seu treball. Blender fa molt fàcil automatitzar les tasques. Saber utilitzar un script pot portar la vostra obra d'art al següent nivell, que us permetrà centrar-vos menys en les tasques repetitives i més en les coses que importen.

Els conceptes bàsics a aprendre sobre l'API són:

- Com les accions a Blender es tradueixen a ordres.
- Conceptes bàsics sobre com utilitzar la consola interactiva de Python.
- Creació d'escriptures macro-like mitjançant la finestra d'informació.
- Diferència entre un addont i un script.

Els conceptes bàsics d'automatització són:

What can you do with it?

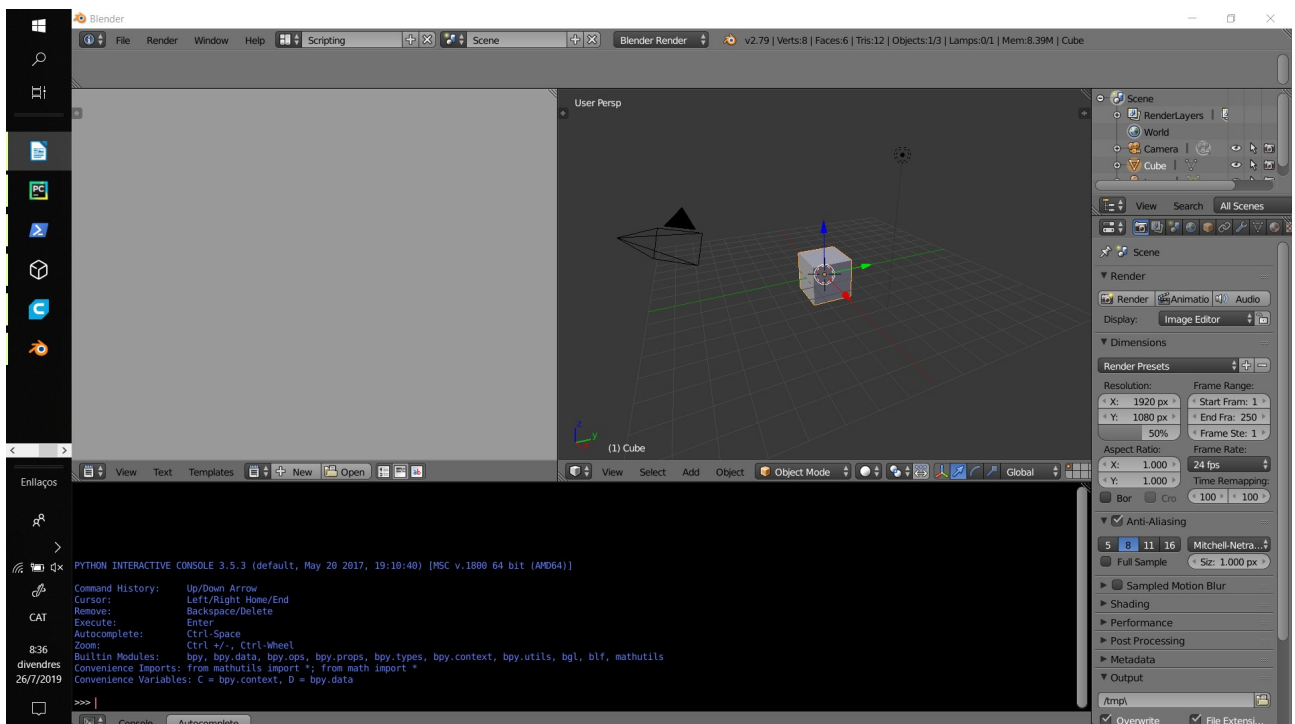
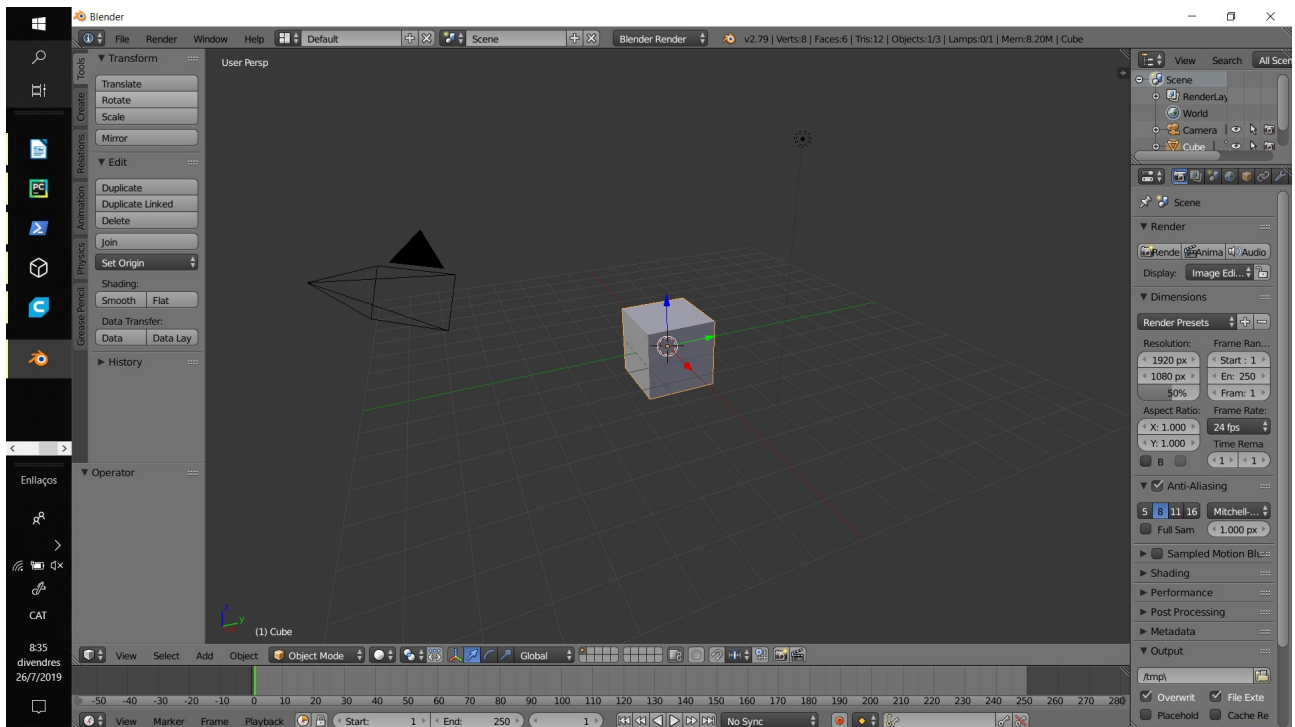
Python scripting in blender has the ability to transform many aspects of how you use the program, including how it looks and behaves.

Automate Tasks	Custom UI & Panels	Create new Operators	Command-line scripting
<ul style="list-style-type: none">- Repeat the same action quickly for many objects- Turn a sequence of steps into a single command- Fast & easy and little coding required	<ul style="list-style-type: none">- Put operators into one handy menu- Modify existing menus with tools/extra settings you like- Hack blender's interface!	<ul style="list-style-type: none">- Create completely new functions- Packaged with UI components- Addon distributed- Power of python, you can do virtually anything!	<ul style="list-style-type: none">- Run blender commands without an interface- Automate or optimize your rendering process- Self-executing templates

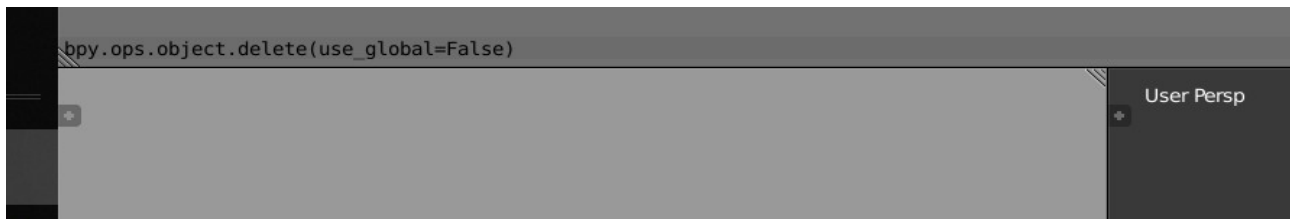
Per aprofundir més en aquest camp mirarem el següent tutorial: **Intro to Python Scripting in Blender - Workshop to automate tasks for artists.**

Link: <https://www.youtube.com/watch?v=KNa5kJd2Epo>

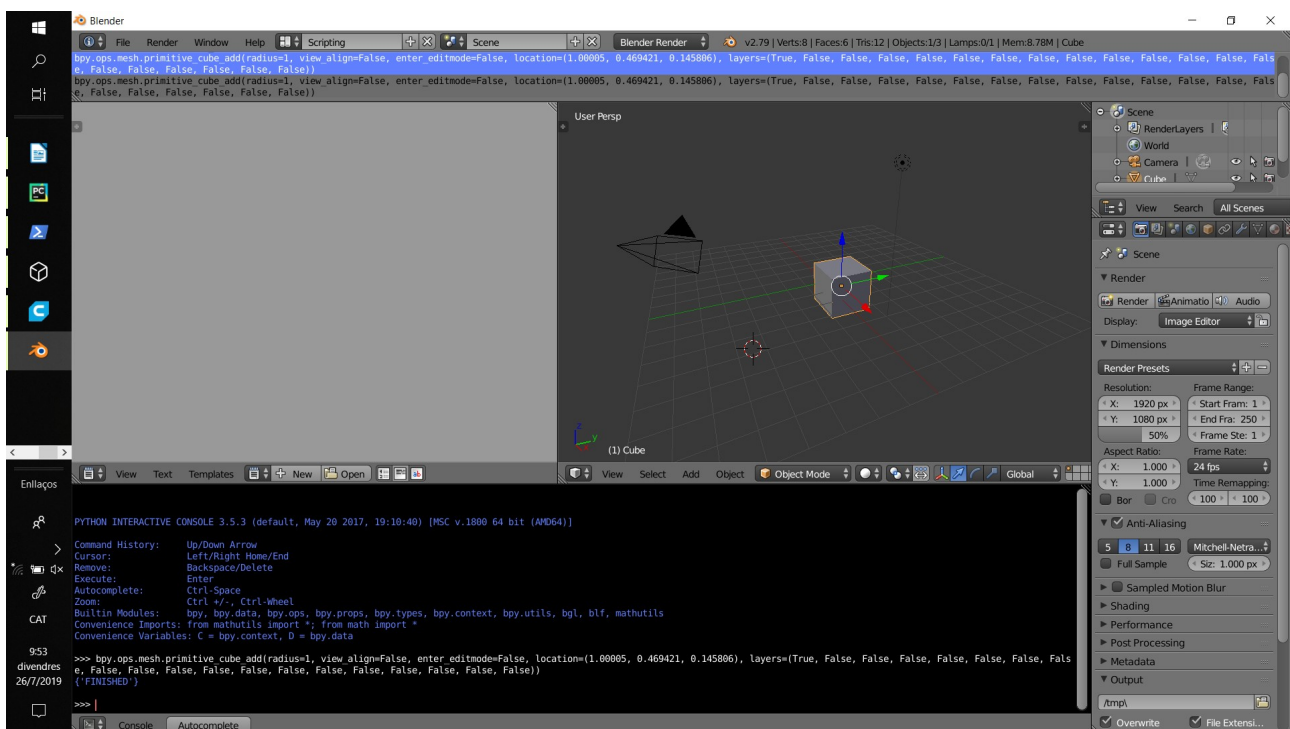
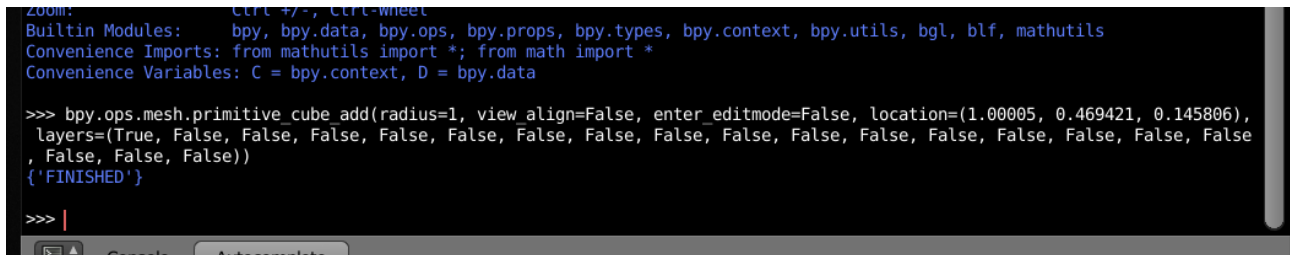
El primer concepte bàsic que aprendrem és que cada acció a Blender es pot traduir en una ordre en Python. Per veure això més clar el primer que farem és personalitzar el display de Blender a Scripting. Així, a temps real, comprovarem que cada ordre que fem en l'interface gràfica té la seva torna a l'script de Blender.



Per exemple, eliminarem el cub de l'escena:



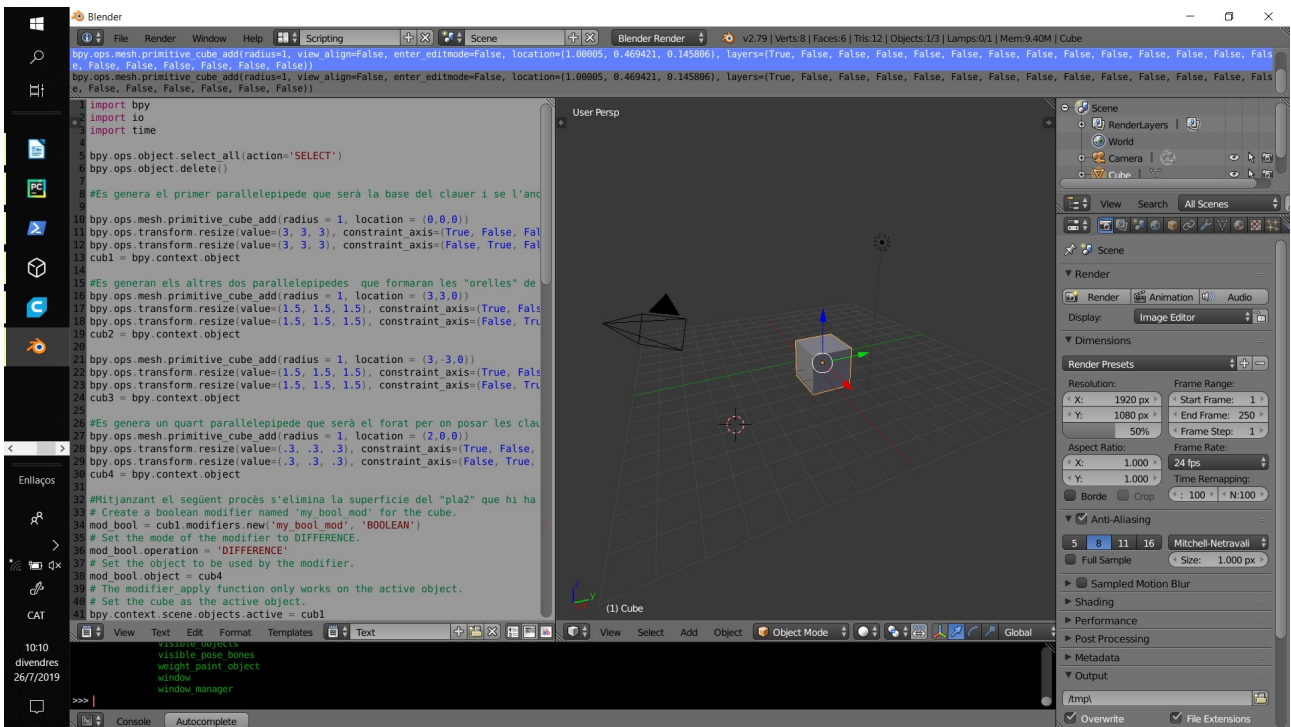
Blender també disposa d'una finestra que anomenarem Interactive Python Console. Seria com la terminal de windows però ens dóna control de Blender en temps real. Per exemple podem crear un cub des d'aquesta terminal.

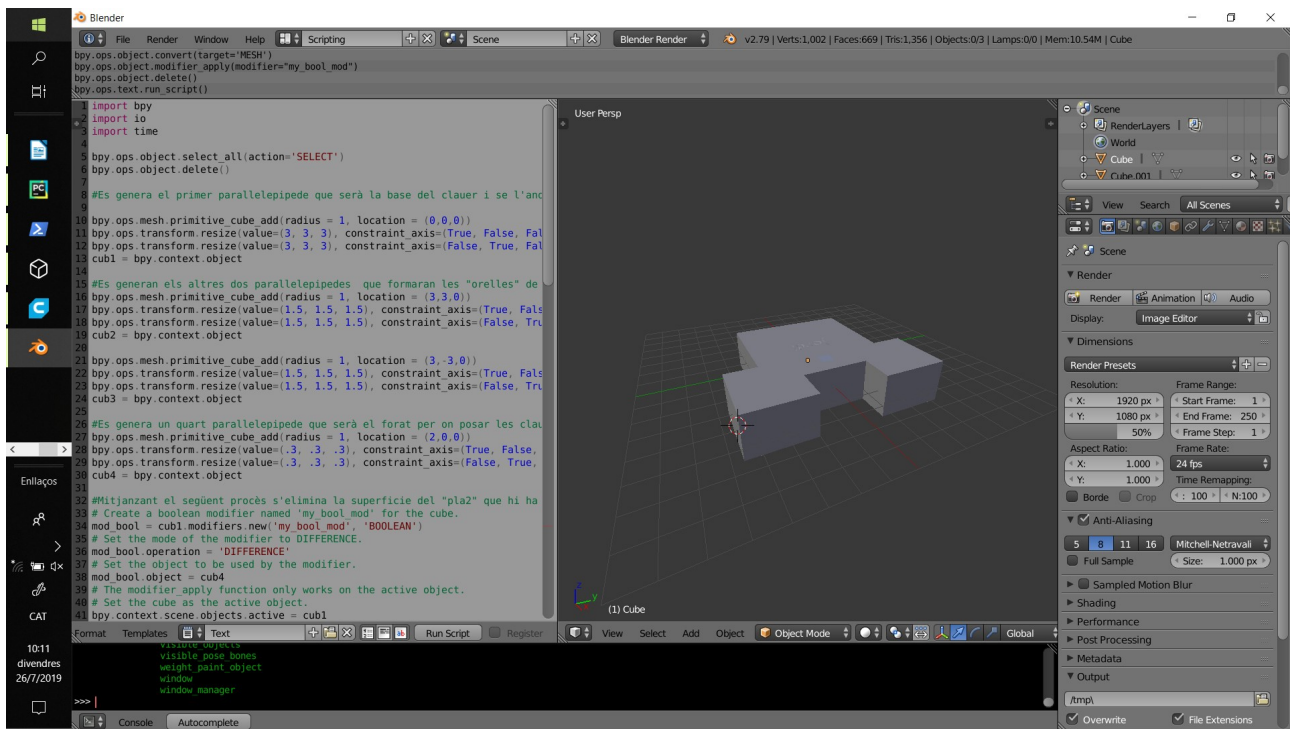


Aquesta terminal interactiva és molt útil perquè et dona la possibilitat de veure totes les ordres disponibles sota un comandament. Per exemple, escriurem `bpy.context.` + ctrl + spacebar.

```
selectable_objects
selected_bases
selected_bones
selected_editable_bases
selected_editable_bones
selected_editable_objects
selected_editable_sequences
selected_objects
selected_pose_bones
selected_sequences
sequences
space_data
tool_settings
type_recast
user_preferences
values
vertex_paint_object
visible_bases
visible_bones
visible_gpencil_layers
visible_objects
visible_pose_bones
weight_paint_object
window
window_manager
>>> bpy.context.
```

Blender et dona també la possibilitat d'utilitzar un editor de text on podràs escriure el teu codi i executar-ho.





Fase 4_Introducció a Ultimate Cura

Aquesta és la fase on preparem l'arxiu 3d que ha creat el software de disseny perquè la màquina 3d l'entengui.

Aquest arxiu final serà un G-Code on es donaran totes les instruccions a la impressora 3D. Velocitat, gruix de capa, infill, etc.

El primer cop que obrim Cura hem d'escollir la nostra impressora, Creality Ender-3, d'aquesta forma Cura adapta els paràmetres predefinitos a la nostra impressora. Després haurem de modificar els paràmetres més importants.

Són els següents:

1. Layer Height. Si tenim un Nozzle de 0.4mm de gruix, no passarem de .25mm alçada.
2. Shell. Sempre el gruix hem d'intentar que sigui múltiple del gruix del nozzle.
3. Infill. Densitat major o menor, depenent de la resistència que necessiti la peça.
4. Speed. Major velocitat, menor qualitat.
5. Travel. Important, activar el Z hop.
6. Support. Si tenim formes amb piràmide invertida, Cura ens crearà suports addicionals.
7. Build Plate Adhesion. Les opcions per adherir la nostra peça són: skirt (falda), raft (balsa), brim (perímetre).

Fase 5_Introducció a l'impressora Creality 3D Ender-3

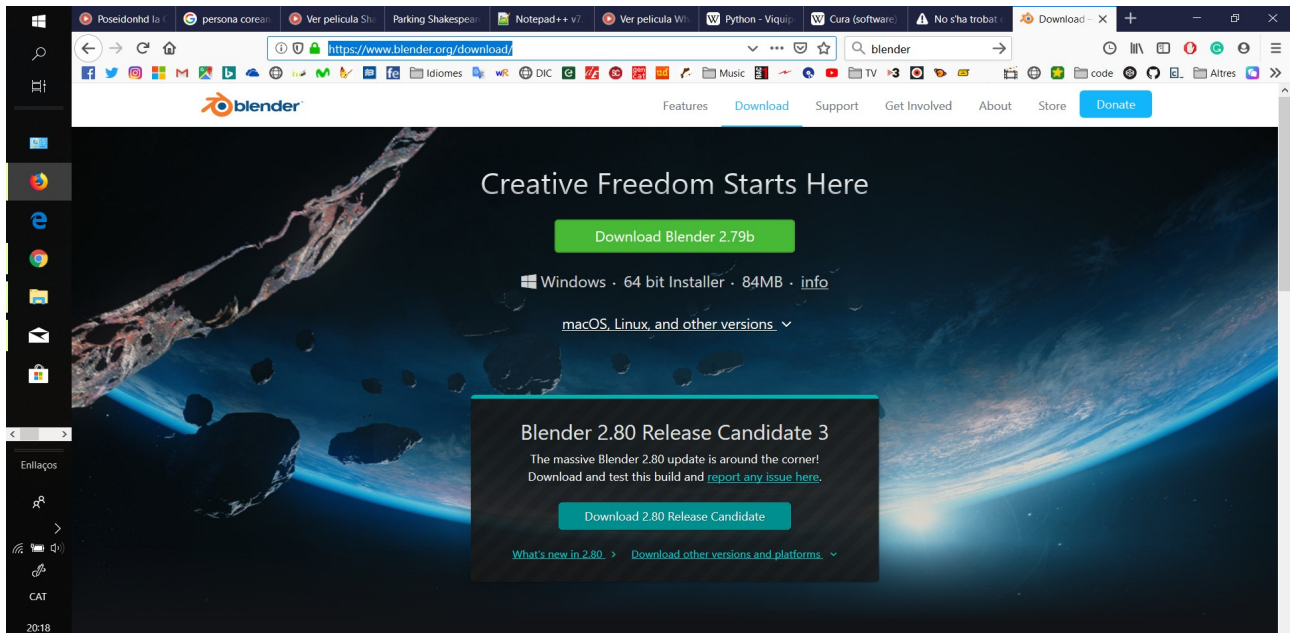
Primer de tot haurem d'anivellar la impressora. Per això és necessari fer un Auto Home i un Dissable Steppers. Després hem de poder passar un paper de gruix normal sense que es quedi embussat però que passi just entre el nozzle i el llit. Aquest procés ho farem en les quatre cantonades del llit.

Després ja podrem seleccionar des de la SD el nostre arxiu per imprimir.

5. Projecte

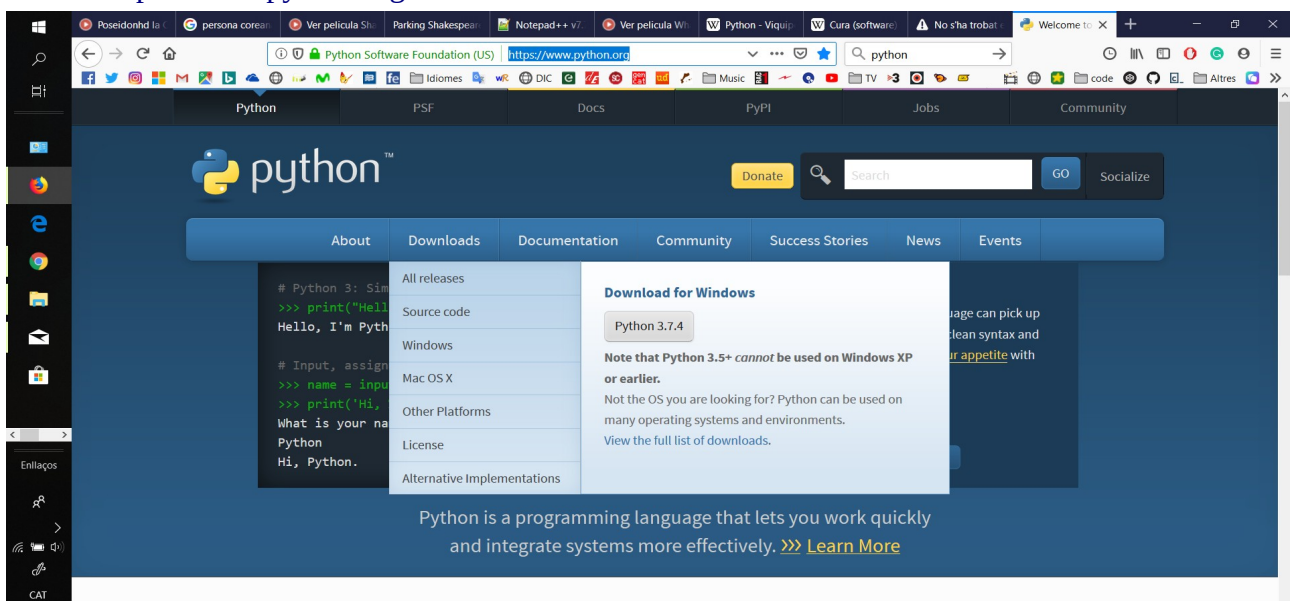
5.1. Instal·lació del Blender

<https://www.blender.org/download/>



5.2. Instal·lació del Python

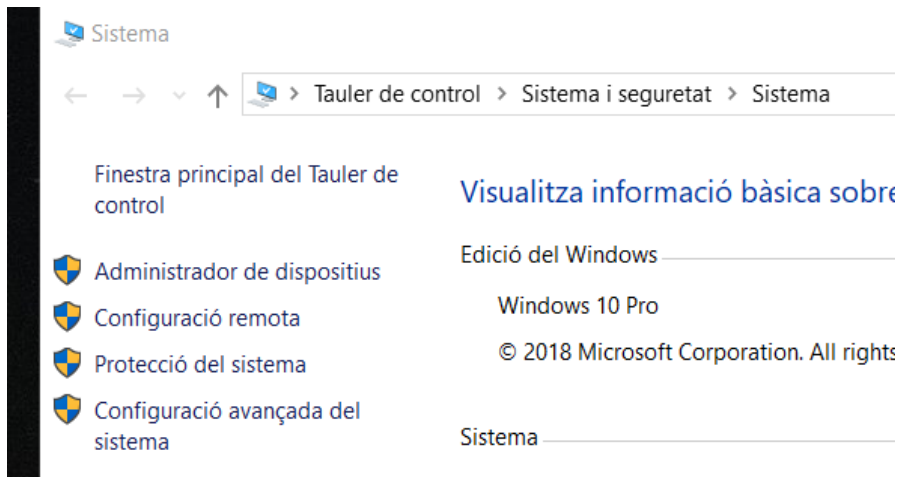
<https://www.python.org/>



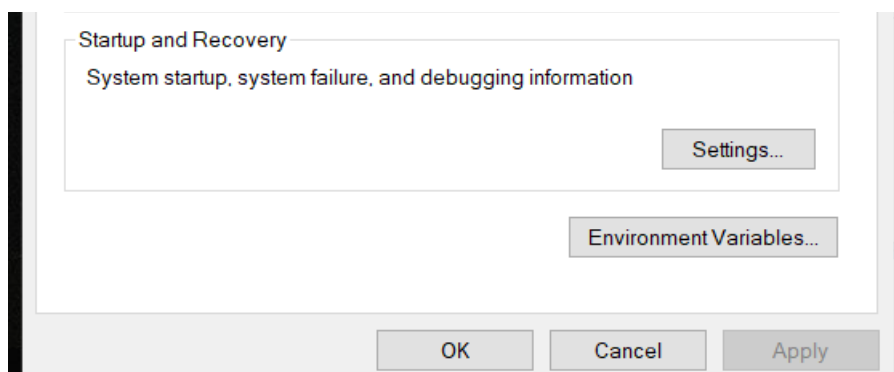
5.3. Crear el Path

Perque el programa Blender s'executi des del codi Python em de crear un Path al sistema.
Sistema/Configuració avançada del sistema/Entorn Variables../Path/C:\Program Files\Blender Foundation\Blender

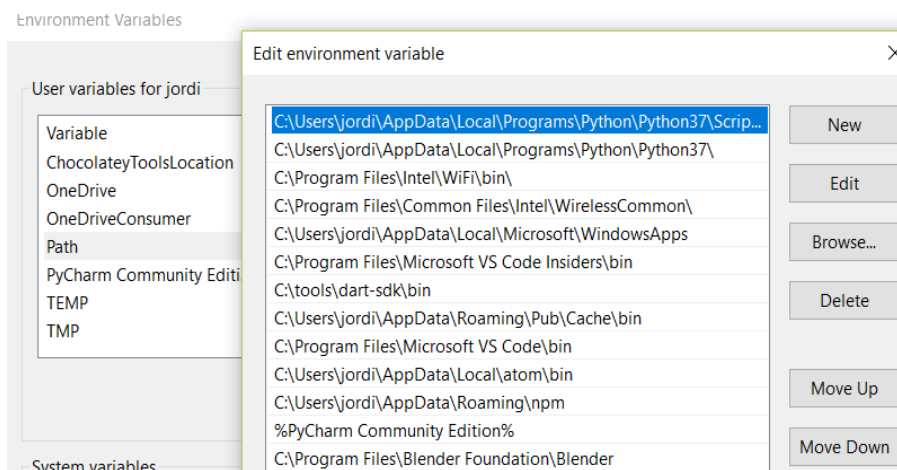
1. Configuració avançada del sistema



2. Entorn Variables



3. C:\Program Files\Blender Foundation\Blender



5.4. Creació del script

#Importem les llibreries.

```
import bpy
import io
import time
```

#Seleccionem tots els objectes de l'escena i els eliminem.

```
bpy.ops.object.select_all(action='SELECT')
bpy.ops.object.delete()
```

#S'obre el fitxer csv on les dades són emmagatzemades i es fa un readlines on el contingut del text queda guardat a la variable línies.

```
fitxer = io.open('csvmickey.csv', 'r')
linies = fitxer.readlines()
for index in range(len(linies)):
    linies[index] = linies[index].replace('\n', '')
fitxer.close()
```

Es crea una funció que generi el dibuix a sobre del qual volem que s'imprimeixin els noms personalitzats dels socis, en aquest cas un clauer amb la silueta de Mickey Mouse.

```
def dibuix(escrit):
```

Es genera el primer paralelepípede que serà la base del clauer.

```
    bpy.ops.mesh.primitive_cube_add(radius=1, location=(0, 0, 0))
    bpy.ops.transform.resize(value=(3, 3, 3), constraint_axis=(True, False,
False))
    bpy.ops.transform.resize(value=(3, 3, 3), constraint_axis=(False, True,
False))
    cub1 = bpy.context.object
```

Es generen els altres dos paral·lelepípedes que formaran les "orelles" de la figura i es col·loquen als costats del primer paral·lelepípede.

```
    bpy.ops.mesh.primitive_cube_add(radius=1, location=(3, 3, 0))
    bpy.ops.transform.resize(value=(1.5, 1.5, 1.5), constraint_axis=(True,
False, False))
    bpy.ops.transform.resize(value=(1.5, 1.5, 1.5), constraint_axis=(False,
True, False))

    bpy.ops.mesh.primitive_cube_add(radius=1, location=(3, -3, 0))
    bpy.ops.transform.resize(value=(1.5, 1.5, 1.5), constraint_axis=(True,
False, False))
    bpy.ops.transform.resize(value=(1.5, 1.5, 1.5), constraint_axis=(False,
True, False))
```

Es genera un quart paral·lelepípede que serà el forat per on posar les claus al clauer.

```
    bpy.ops.mesh.primitive_cube_add(radius=1, location=(2, 0, 0))
    bpy.ops.transform.resize(value=(.3, .3, .3), constraint_axis=(True, False,
False))
    bpy.ops.transform.resize(value=(.3, .3, .3), constraint_axis=(False, True,
False))
    cub4 = bpy.context.object
```

Mitjançant el següent procés crearem el forat al paral·lelepípede.
Creem un modificador booleà anomenat 'my_bool_mod' per al cub.

```
mod_bool = cub1.modifiers.new('my_bool_mod', 'BOOLEAN')
```

Establim el mode del modificador a DIFFERENCE.

```
mod_bool.operation = 'DIFFERENCE'
```

Definim l'objecte que s'utilitzarà per el modificador.

```
mod_bool.object = cub4
```

La funció modifier_apply function només funciona sobre l'objecte actiu.
Configurem el cub1 com a objecte actiu.

```
bpy.context.scene.objects.active = cub1
```

Apliquem la modificació.

```
res = bpy.ops.object.modifier_apply(modifier='my_bool_mod')
```

Eliminem el cub4.

```
cub4.select = True  
bpy.ops.object.delete()
```

Crear el text, extruir-lo, centrar-ho, moure-ho i rotar-lo.

```
bpy.ops.object.text_add(view_align=False, enter_editmode=False, location=(0,  
0, 0), layers=(  
    True, False, False, False, False, False, False, False, False, False, False,  
    False, False, False, False, False,  
    False, False, False, False))  
bpy.context.object.data.extrude = 1  
bpy.context.object.data.align_x = 'CENTER'  
bpy.ops.transform.translate(value=(0, 0, 0.5), constraint_axis=(False,  
False, True))  
bpy.ops.transform.rotate(value=4.71239, axis=(0, 0, 1),  
constraint_axis=(True, True, False))
```

Eliminem el text predeterminat "Text".

```
bpy.ops.object.editmode_toggle()  
bpy.ops.font.delete(type='PREVIOUS_OR_SELECTION')  
bpy.ops.font.delete(type='PREVIOUS_OR_SELECTION')  
bpy.ops.font.delete(type='PREVIOUS_OR_SELECTION')  
bpy.ops.font.delete(type='PREVIOUS_OR_SELECTION')
```

Escrivim la nova paraula.

```
chars = escrit  
for char in chars:  
    bpy.ops.font.text_insert(text=char)  
bpy.ops.object.editmode_toggle()
```

Transformar el text en mesh.

```
bpy.ops.object.convert(target='MESH')  
text1 = bpy.context.object
```

Mitjançant el següent procés crearem el forat al paral·lelepípede amb el nom escrit.
Creem un modificador booleà anomenat 'my_bool_mod' per al cub.

```
mod_bool = cub1.modifiers.new('my_bool_mod', 'BOOLEAN')
```

Establim el mode del modificador a DIFFERENCE.

```
mod_bool.operation = 'DIFFERENCE'
```

Definim l'objecte que s'utilitzarà per el modificador.

```
mod_bool.object = text1
```

La funció modifier_apply function només funciona sobre l'objecte actiu.

Configurem el cub1 com a objecte actiu.

```
bpy.context.scene.objects.active = cub1
```

Apliquem la modificació.

```
res = bpy.ops.object.modifier_apply(modifier='my_bool_mod')
```

Eliminem el text.

```
text1.select = True  
bpy.ops.object.delete()
```

#Es crea un document .stl amb cada nom que aparegui al .csv i la figura corresponent.

```
nom='./'+escrit+'.stl'  
bpy.ops.export_mesh.stl(filepath=nom)  
bpy.ops.object.select_all(action='SELECT')  
bpy.ops.object.delete()
```

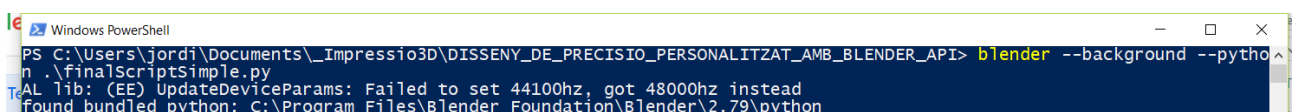
Agafa les dades que del archiu csv que s'han guardat a la variable línies i crea una figura amb cada nom dins de la variable línies .

```
for name in linies:  
    dibuix(name)
```

5.5. Executar l'script

1. A la carpeta on tenim l'script, boto esquerra+shift.
2. Escull obre la finestra de poweShell aquí.
3. Executa el programa:

blender --background --python .\finalScriptSimple.py



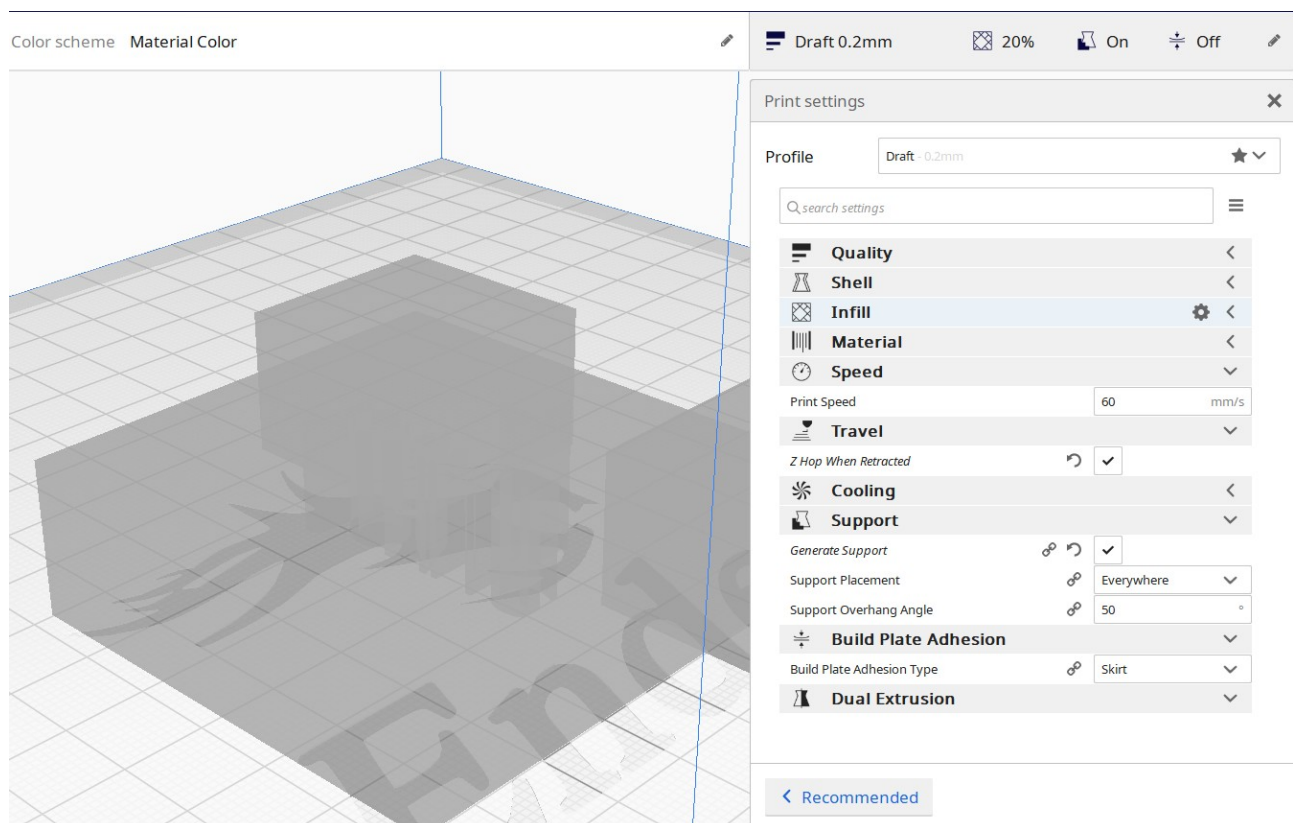
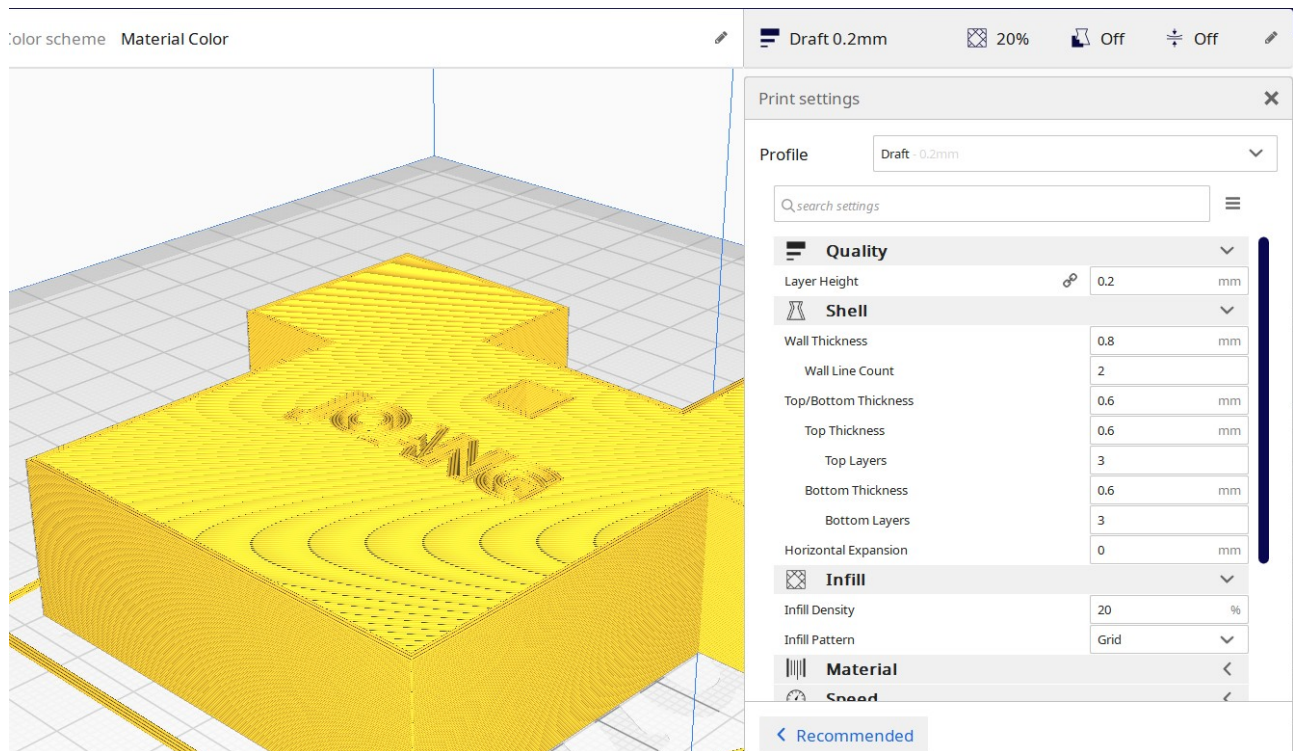


5.6. Preparació arxiu STL

1. Instal·lem el programa Ultimaker Cura:

<https://ultimaker.com/software/ultimaker-cura>

2. Obrim l'arxiu amb el programa.
3. Modifiquem els paràmetres:



5.7. Imprimir

- Pas 1. Obrir la impressora.
- Pas 2. Seleccionar menú Prepare.
- Pas 3. Seleccionar Auto Home.
- Pas 4. Seleccionar Disable Steppers.
- Pas 5. Anivellarem la impressora.
- Pas 6. Seleccionem Print from SD.
- Pas 7. Seleccionem l'arxiu que volem imprimir.

6. Conclusions

Automatitzar processos mola.

7. Annexos

- 1. Sintaxi del llenguatge: my Python summary notes.

8. Biografia

- **Python Tutorial for Beginners [Full Course] 2019.**
https://www.youtube.com/watch?v=_uQrJ0TkZlc.
- **Intro to Python Scripting in Blender - Workshop to automate tasks for artists.**
<https://www.youtube.com/watch?v=KNa5kJd2Epo>