

**Rapport de stage**

**Name: Joe Geib**

**Firma: securitymadein.lu**

**Zeitperiode: 2. Januar 2018 – 9. Februar 2018**



Inhaltsverzeichnis

[1) Einleitung 2](#_Toc505953037)

[2) Hauptteil 2](#_Toc505953038)

[1) Woche 1: 2. Jan – 5. Jan 2](#_Toc505953039)

[2) Woche 2: 8. Jan – 12. Jan 4](#_Toc505953040)

[3) Woche 3: 15. Jan – 19. Jan 5](#_Toc505953041)

[4) Woche 4: 22. Jan – 26. Jan 6](#_Toc505953042)

[5) Woche 5: 29. Jan – 2. Feb 7](#_Toc505953043)

[6) Woche 6: 5. Feb – 9. Feb 8](#_Toc505953044)

[3) Schlussfolgerung 9](#_Toc505953045)

[4) Anhang: Ubuntu Installation 10](#_Toc505953046)

[5) Anhang: Python Dokumentation 12](#_Toc505953047)

[6) Anhang: Scratch Game 14](#_Toc505953048)

[7) Anhang: Light Painting 49](#_Toc505953049)

# Einleitung

Mein Praktikum findet in der Firma „security made in lu“ statt. Ich arbeite hier vom 2. Januar bis zum 9. Februar und in diesem Zeitraum ist es meine Aufgabe mit meinen Mitarbeitern gemeinsam an verschiedenen Aufgaben und Problemen zu arbeiten und diese zu lösen.

„Security made in lu“ ist eine Sicherheitsfirma, die Ihren Sitz in der HSBC Bank in Luxemburg Stadt hat und Ihr Ziel ist es, wie der Name schon sagt, Sicherheit zu gewährleisten.

„Security made in lu“ wurde am 5. Mai 2010 unter dem Namen SMILE (Security made in Lëtzebuerg) gegründet. Im Februar 2015 wurde „Security made in lu“ aus einer Initiative von SMILE gegründet und besteht aus 3 Arbeitsbereichen:

* Circl: Computer Incident Response Center Luxembourg
* Cases: Cyberworld Awareness and Security Enhancement Services
* C3: Cybersecurity Competence Center

Jeder dieser Bereiche behandelt leicht unterschiedliche Aufgaben, aber alle haben das gleiche Ziel: Online Sicherheit.

Ob, Phishing E-Mails, Identitätsdiebstahl, Kreditkarten Informationsdiebstahl, Materialsicherheit oder einfacher physikalischer Diebstahl, das Ziel von „Security made in lu“ ist es, die Sicherheit dieser Dinge zu sichern.

# Hauptteil

## Woche 1: 2. Jan – 5. Jan

Im Beginn der ersten Woche wurde mir eine recht einfache Aufgabe gegeben, bei der ich das Ubuntu Betriebssystem auf mehreren Laptops installieren sollte. Dies wurde jedoch von einem bootable USB stick aus gemacht. Da bootable USB sticks für Linux anders funktionieren als bei Windows, musste Ich erst herausfinden wie man einen bootable USB sick für Linux macht.

Nach einiger Zeit habe ich einen Linux Befehl gefunden, mit dem man solch einen bootable USB stick machen kann.

Nach der Installation des Linux Betriebssystem musste ich noch die üblichen Updates und Upgrades machen und dann noch das Zeichenprogramm Gimp installieren.

Anschliessend, musste Scratch Offline Editor, welches Adobe Air benötigt installiert werden.

*(Für meine selbstgemachte Cheatsheet siehe „Anhang: Ubuntu Installation“)*

Während den Installationen bekam ich eine kleine Einführung zu Botnets und wie eine der grössten DDoS Attacken von einem Schadprogramm names Mirai durchgeführt wurde.

Anschliessend wurden mir Backdoors etwas näher gebracht. In diesem Fall waren es Physikalische Backdoors. Diese findet man meist an Reisekoffern, da man solche haben muss falls man in die USA reisen möchte. Der Grund dafür sind die Gesetze der USA, da diese solche Backdoor Schlösser verlangen.

Diese Backdoors sind genormt von TSA 001 -TSA 007. Es gibt also gerade mal 7 verschiedene Schlüssel um fast alle Kofferschlösser der Welt aufzumachen. Die Sicherheitslücke hier ist, dass diese Schlüssel ganz leicht aus einem 3D Drucker rausgedruckt werden können. Die Modelle hierfür findet man in 1 Minute über Google.



( Bild Quelle: <http://enterthecore.net/tsa007/> )

Die Ausmassen der 3D Drucktechnologie sind weitaus schlimmer. So konnte man zum Beispiel, mit 3 Fotos eines Schlüssels in der Hand eines Mannes, aus leicht verschiedenen Winkeln von hunderten von Metern aus ein Modell des Schlüssels anfertigen, welches das Schloss öffnen konnte.

Nachdem die Linux Installationen fertig waren, wurde mir ein besonderes Projekt anvertraut:

Das Ziel war es Musik herzustellen, mithilfe des tcpdump Befehls.

Um Kindern über Internetsicherheit aufzuklären, wurde beschlossen, Ihnen mit Musik den Datenverkehr zu erklären.

Die einzige Hilfe die Ich für dieses Projekt bekam war eine vordefinierte Python Datei.

Den Rest der Woche verbrachte Ich damit, herauszufinden wie Ich das Projekt ausführen sollte.

## Woche 2: 8. Jan – 12. Jan

In der zweiten Woche arbeitete Ich zum Grösstenteil an dem tcpdump Projekt.

Mir gelang den output des tcpdump Befehls an das Python skript zu schleusen.

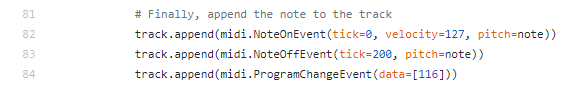
Da der Klang im Terminal selbst aber nicht immer einwandfrei funktioniert hat habe ich dem tcpdump die Eigenschaft zugewiesen, dass der Befehl nachdem er 1000 Pakete erhalten hat aufhört und dann eine MIDI Datei erstellt, die die Musik enthält. Die Finale Zeile, die Erfolg zeigte, war folgende:

C:\Users\Windblows\Desktop\Screenshot from 2018-01-16 11_04_25.png

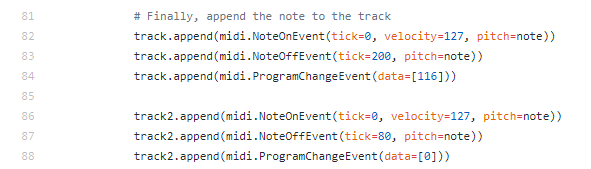
Nun hatte ich eine MIDI Datei die ein Piano Rythmus spielte, abhängig vom Datenverkehr.

Um etwas mehr Leben in die Melodie zu bringen, habe ich mich über Python informiert und anschliessend versucht, das Python skript selbst zu modifizieren. So habe ich herausgefunden wie man ein neues Instrument hinzufügt und wie man die Geschwindigkeit mit der das Instrument gespielt wird beeinflussen kann.

Der Code wie er am Anfang war:



Das zweite Instrument das Ich hinzugefügt habe:



Wobei es sich bei „data=[116]“ (Steel Drums) und „data=[0]“ (Acoustic Grand Piano) in der Zeile 84 und 88 um die benutzten Instrumente handelt.

*(Für eine genaue Dokumentation des von mir veränderten Python Codes siehe „Anhang: Python Dokumentation“. Ausserdem kann eine Beispiels Datei einer solchen MIDI File auf meinem github Profil gefunden werden, unter JeyAl -> Repositories -> internship -> tcpdump\_tunes -> test\_3)*

Mir wurde dann noch eine kleine Einführung in die Virtuelle Realität (kurz: VR) gegeben.

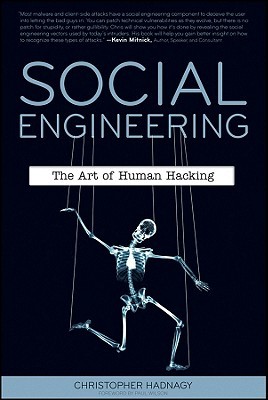
Ein Mann der zu Besuch kam hat ein VR Spiel demonstriert, bei dem ein Cyberangriff simuliert wird und man den Schaden so gering wie möglich halten muss. Nach 3 Minuten ist das Spiel vorbei. Ausserdem hat er für die BIL als Promotion ein VR Horrorspiel entwickelt, das er mitgebracht hatte und das ausprobiert werden konnte.

## Woche 3: 15. Jan – 19. Jan

In Woche 3 habe ich versucht, ein 3D Modell, in meinem Fall einer der ausdruckbaren TSA Schlüssel, in sketch up zu importieren und diesen dann auf Modelo (modelo.io) als 3D objekt anzuzeigen.

Dieses Objekt sollte mit Hilfe meines Smartphones, welches eine VR Brille simulieren sollte, vor mir sichtbar sein. Dies funktionierte auch, jedoch existierte keine Tiefe, das Objekt schwebte also lediglich vor mir, Ich konnte mich ihm nicht nähern.

Später gab man mir ein Buch das Ich mir anschauen sollte.



( Bild Quelle: <https://www.goodreads.com/book/show/9068044-social-engineering> )

Im Buch Social Engineering wird beschrieben wie man Menschen manipulieren oder beeinflussen kann, um Informationen von Ihnen zu bekommen. Ein recht bekanntes Beispiel das auch im Buch erwähnt wird ist der erfolgreiche Nigerianische Prinz Scam, bei dem man die Gier des Menschen ausnutzt um Ihr vertrauen zu gewinnen. Unter anderem erwähnt das Buch auch, wie leicht es zu schwerwiegendem Daten oder Informationsverlust kommt.

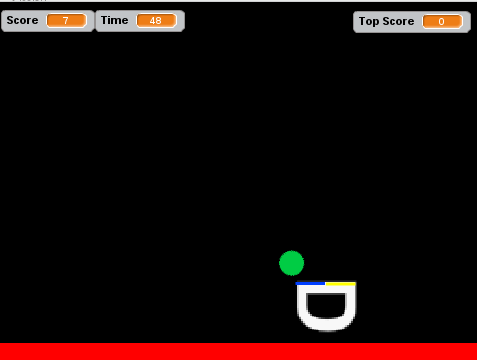
Im Buch wird ein Beispiel genannt bei dem man die Kreditkarte und die Kreditkartennummer einer Person bekommt indem man zuerst die Tasche mit Kreditkarte stiehlt, dann, wenn das Opfer den Verlust merkt, die Panik des Opfers ausnutzt um die Nummer zu bekommen. In diesem Fall gab sich der Dieb als Arbeiter der Bank aus bei der die Frau angemeldet ist, sagt Ihr das er die Karte deaktivieren könne indem sie während eines Gesprächs mit seiner „Kollegin“ auf seinem Smartphone den PIN auf der Tastatur eingibt.

Der Trick hierbei ist, dass eine Person nie einfach den PIN irgendwo eingeben würde, jedoch hier die Erfolgschancen hoch sind, da durch die Panik, das rationelle Denken des Opfers ausgeschaltet ist.

Das Lesen des Buches beanspruchte fast den ganzen Rest der Woche und es hat mich sehr viel gelehrt.

## Woche 4: 22. Jan – 26. Jan

Mir wurde ein neues Projekt anvertraut.

Dieses Projekt hat zum Teil mit dem Ersten zu tun. Diesmal geht es nur darum, Kindern das programmieren näher zu bringen.Ich sollte ein „Juggling Game“ in Scratch programmieren, ein Spiel bei dem man mit einer Platform am untern Rand den Ball, der durch das Spiel Fenster fliegt, davon abhalten muss, den Boden zu berühren.

Dies war recht einfach, hat jedoch sehr viel Zeit in Anspruch genommen, so dass ich hiermit die ganze Woche beschäftigt war, da das ganze Kinderfreundlich und in 2 Sprachen dokumentiert werden musste.

*( Für die Englischsprachige dokumentation siehe „Anhang: Scratch Game“)*

## Woche 5: 29. Jan – 2. Feb

Während der fünften Woche, habe ich mit einer vorprogrammierten Platine, die mit LEDs bestückt war versucht, mit Hilfe einer Kamera ein Lichtbogen aufzunehmen, welcher eine Schrift bzw. Bild in der Luft erzuegen soll. Hier ein Beispiel wie es aussehen sollte:



( Bild Quelle: <https://learn.adafruit.com/assets/16176> )

Dies stellte sich als relativ schwierig raus, da die Platine gleichmässig in einer bestimmten Geschwindigkeit bewegt werden musste. Ausserdem musste ich mit den Einstellungen der Kamera die mir gegeben wurde herumspielen um die richtige Einstellungen herauszufinden.

Da ich jedoch noch nie mit Kameras gearbeitet hatte, stellte sich dies als eine Herausforderung heraus.

Das Ziel war es, die richtige Filmlänge einzugeben, die richtige Lichtintensität und der richtige Fokus.

Gleichzeitig, habe ich nach und nach meine Scratch Dokumentation mit Hilfe einer Arbeitskollegin überarbeitet und ins französische übersetzt.

Anschliessend habe ich noch eine kleine Powerpoint über die Evolution der Videospielgrafik gemacht, welche durch eine Zeitlinie anzeigt, wie sich die Grafik in Videospielen über die Jahre entwickelt hat und besser geworden ist.

## Woche 6: 5. Feb – 9. Feb

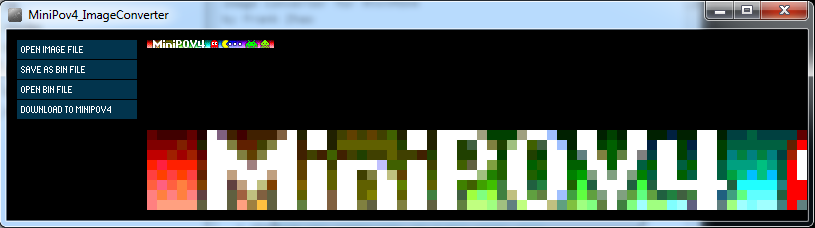
In der letzten Woche wurden mir von einem Mitarbeiter einige Tools gezeigt, mit denen sie hier in der Bank verschiedene Links, E-Mails, etc... auf Schadsoftware überprüfen. Das Prinzip ist relativ ähnlich zu dem der bekannten Webseite virustotal.com, jedoch werden bei diesen Tools die Resultate nicht öffentlich angezeigt so wie bei virustotal.com.

Ein anderes Tool, überprüft ebenfalls Links, E-Mails, etc.. auf verschiedene Wörter. So kann man als Schlüsselwort meinen Namen eingeben und das Tool überprüft ob z.B. in einem Link irgendwo die Buchstabenkombination „Joe“ steht.

Eine andere Aufgabe die Ich diese Woche behandelt habe, ist die Re-programmierung der Light Painting Platine.

Das Farbmuster ist vorprogrammiert und man wollte, dass Ich es ändere.

Um dies Umzusetzen habe ich ein Programm gefunden, das ein Bild in ein solches Farbmuster umwandelt und habe dann mein eigenes Farbmuster hochgeladen.



Hierzu habe Ich auch eine Dokumentation gemacht.

*(siehe Anhang: „Light Painting“)*

# Schlussfolgerung

Das Praktikum fand Ich sehr gut. Ich habe extrem viel gelernt, in Bereichen die in der Schule nicht behandelt werden, genauso wie in Bereichen die wir momentan als Fächer haben.

Vor allem mochte Ich, wie meine Arbeit extrem Abwechslungsreich war und Ich die Wahl hatte, ob ich etwas anderes machen oder sehen wollte, sollte mir das momentane Thema nicht gefallen.

Dadurch habe Ich viele verschiedene Dinge gesehen, die mir im späteren Leben sehr wahrscheinlich nützen werden.

Die Arbeit war relativ schwierig, jedoch habe Ich jedes Hindernis überwältigt und dazu gelernt, wie z.B. Das von mir modifizierte Python skript, da Ich noch nie zuvor Python gesehen hatte.

Alles in allem war das Praktikum sehr lehrreich und zufriedenstellend.

# Anhang: Ubuntu Installation

Bootable USB with Ubuntu:

→ gksudo usb-creator-gtk

The command lets me make a startup disk.

Formatting the USB stick via make startup disk program, leaving settings on default which are:

“When starting up from this disk, documents and settings will be stored in reserved extra space → 1 GB”

The USB Drive is now ready to install the OS on any device.

On the device the OS is going to be installed on, the USB drive has to be assigned as boot device.

→ Press F2 on boot up

→ Switch to boot tab

→ Add new boot option

→ Path for boot option > USB device > EFI > BOOT > BOOTx64.EFI

→ Add boot option > enter name

→ create

→ Switch to Save & Exit tab

→ choose the created boot option

Ubuntu settings:

Timezone → Luxembourg

Keyboard → Switzerland (French)

Username → Scratch

Password → scratch is fun

Updates:

→ sudo apt-get update

→ sudo apt-get upgrade

Gimp:

→ sudo apt-get install gimp

Scratch Offline Editor:

→ install Adobe AIR:

→ wget -O adobe-air.sh http://drive.noobslab.com/data/apps/AdobeAir/adobe-air.sh

→ chmod +x adobe-air.sh;sudo ./adobe-air.sh

→ Download and Install Sratch Offline Editor from https://scratch.mit.edu/download

(Root password had to be changed. Terminal → sudo passwd root)

# Anhang: Python Dokumentation

Documentation of the python script

Explanation of changed code

* The octave multiplier set to 5 (note = note + octave \* 5) (Line 76) -> Sounds a bit darker than the natural instrument sound, but different pitches sound better.  
  -> The pitch that seems to be the most reoccuring is rather high pitched and therefore sounds less bad with this octave setting.  
  -> Additionally, a max octave value has to be set, to prevent extremely high pitched notes (Line 15).  
  -> Next, a if statement has been set, that whenever the result of (note = note + octave \* 5) is higher than 85, it is being limited to 85.
* Added a second instrument (Line 35/36/86/87/88/96) -> The MIDI File consists of one single pattern, but the pattern can consist of multiple tracks.  
  -> In order to add a second track with a different instrument to the pattern we declare a second track first (Line 35).  
  -> Then we append the created track to the pattern (Line 36).  
  -> Just like with the first track, a NoteOnEvent, NoteOffEvent and ProgramChangeEvent (Instrument) has to be assigned (Line 68-88).  
  -> Finally a EndOfTrackEvent has to be assigned to the newly created track(Line 96).  
  -> More instruments can easily be added with the same principle.

Short Explanation of specific lines of code

* MAX\_OCTAVE

The highest possible value the octave can have. The pitch of a note will never go above this value. Decrease this to force a darker sound. Increase this to allow higher pitched sound.

* note = note + octave \* 5

The multiplicator in this variable can be used to change the pitch of the sound. Lower value means darker sound, higher value means higher pitched sound. Keep in mind that if the multiplicator is increased, the pitch will still not go above the MAX\_OCTAVE value.

* track.append(midi.NoteOnEvent(tick=0, velocity=70, pitch=note))

The tick determines when the key is being hit. Increasing this value means it takes longer after a key hit until the next key is being hit.  
The velocity determines how hard the key of a piano for example is being hit, in other words this determines the base volume (max. value: 127).  
The pitch determines the pitch of the note. This is set to the (note = note + octave \* 5) variable. Changing said variable therefore changes the pitch.

* track.append(midi.NoteOffEvent(tick=300, pitch=note))

The tick determines how long the key is being pressed. Lowering this value means notes are being played faster, increasing this value means it takes longer until the next note is being played.  
The pitch is the same as stated before.

* track.append(midi.ProgramChangeEvent(data=[0]))

This function determines which instrument is being played. Change the data value to change the instrument.  
Which instrument equals which data parameter can be found here <https://en.wikipedia.org/wiki/General_MIDI#Program_change_events>  
The value of the instrument is always -1 than shown on wikipedia!

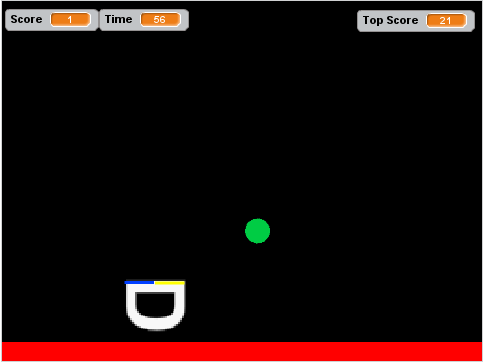
* Depiction of midi notes  
  <http://www.electronics.dit.ie/staff/tscarff/Music_technology/midi/midi_note_numbers_for_octaves.htm>

In the link above it is stated that C5 is the middle C note, but sometimes there are alternatives where a software interprets C3 as middle note, shifting the table upwards by 2 octaves.  
This means that additionally, in Line 62, 66, 68, 70, 72 the midi.x\_3 can be changed to alter the sound that way as well.  
Currently they are set to C\_3, D\_3, G\_3, F\_3, so in the middle of the piano to guarantee a neutral base sound.

# Anhang: Scratch Game

Juggling

**This is a simple guide on how to program your own juggling minigame with Scratch**  
**Note: While doing this, make sure to save your project often to prevent dataloss! There will be checkboxes in front of every step. This doesn't mean that every step has something to do. Sometimes it's just an explanation of the previous step for example. In that case, only check the box if you have truly understood the explanation**

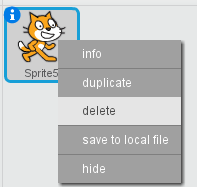
[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Introduction_Screen.png)

Chapter 1: First steps

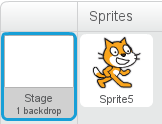
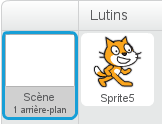
**Step 1: Setting up a background**  
The first step will teach you how you use a sprite as a platform to juggle the ball

**Activity Checklist**

*  Start a new Scratch and delete the cat sprite so that your project is empty. Right click the cat sprite in the sprite area below the project area

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Del_Cat_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Del_Cat_franc.png)

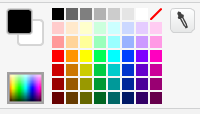
*  Change the background of the stage to black. In order to do this, click on the stage to the left of the sprite area.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Stage_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Stage_fr.png)

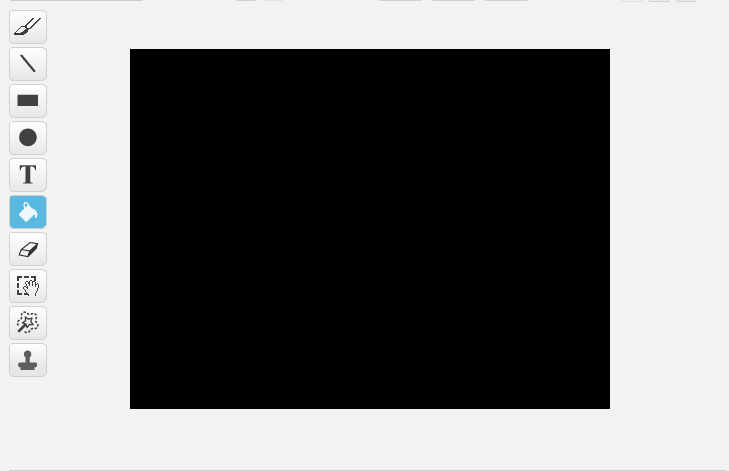
*  Next, pick the bucket tool in the backdrop area.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Palette.png)

*  Make sure that black is chosen as colour. You have a colour palette below the area you're working in right now.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Colours.png)

*  Now, with the bucket tool and black as colour selected, simply click into the big white area that you see on the right side of your screen. The result should look like this.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Backdrop.png)

**Step 2: Creating the platform**

*  The juggling game needs a platform to bounce the ball off of. In this step we're creating this platform, which in our case simply consists of the letter D. This letter, when the game starts, will go into position and act as platform!
*  Create a new sprite by clicking on the brush symbol above the sprite area.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/New_Sprite_en.png) [title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/New_Sprite_fr.png)

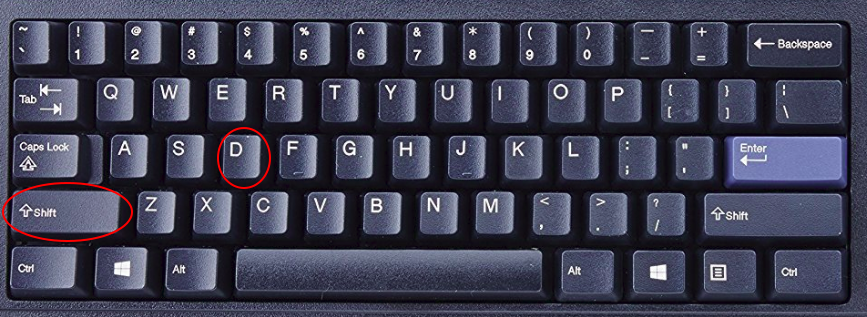
*  Now in the backdrop area on the right (The same area, where you painted the background black), you want to select the Text tool.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Backdrop_Text.png)

*  Select white as coulour in the colour palette. (Both squares on the left of the colour palette are white)

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Colour_White.png)

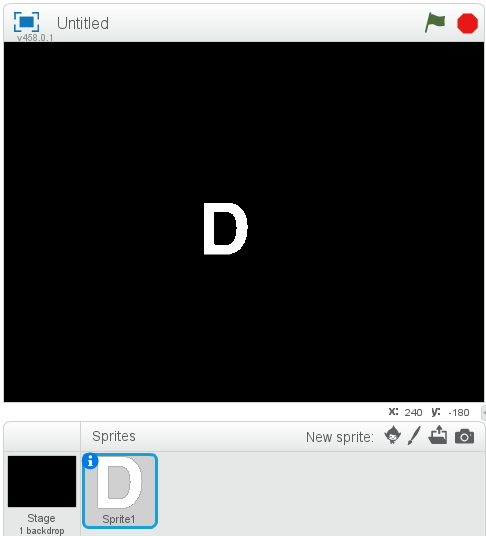
*  Now click into the area in the middle to open a text box. Enter a capital D. You do this by holding shift and then pressing the letter D.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Keyboard.png)

*  Since the D sprite is white it may seem invisible at first but it's there. Now, unselect the D sprite by clicking anywhere else in this area. If you do this, a bunch of rectangles and dots are going to appear on the D sprite.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/D_Grid.png)

*  Select, the diagonal rectangles and drag them to make the D sprite bigger. Should you accidently unselect the D sprite or you're unsatisfied with the size, click on the little undo arrow that is pointing to the left, at the top part of this area.  
    [title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Undo.png)
*  If you're satisfied with the size of the D sprite, click anywhere in the area again to unselect it and make it appear in your project area on the left side.
*  Your project should now look like this.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Project1.png)

**Step 3: Creating the Ball**

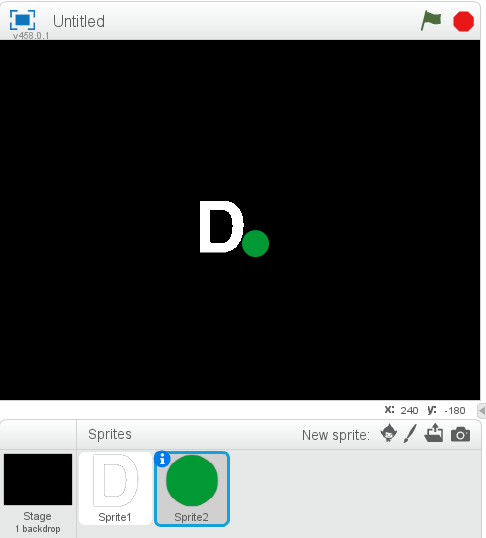
*  The game obviously needs a ball which we are going to create now. Create a new sprite, just like you did with the letter D.
*  Before, we used the bucket and the text tool. This time we are going to use the circle tool.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Select_Ball.png)

*  Select a green colour in the colour palette. To the left of the colour palette you also want to select the "fill" option for the ball. This means that we aren't just drawing the outlines, but the full ball.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Ball_Fill.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Colour_Green.png)

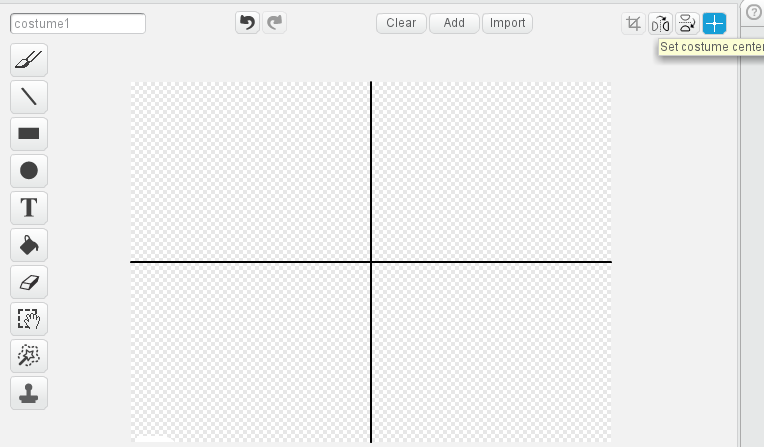
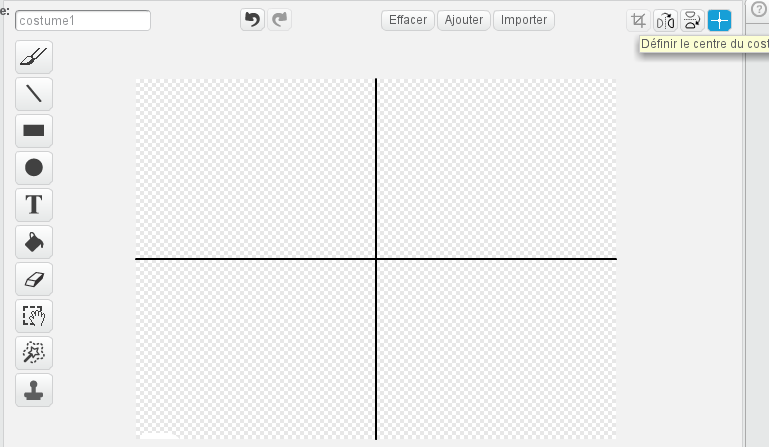
*  Now, inside the area on the right, hold shift and the left mouse button, then drag the mouse to create a ball. The reason why you want to hold shift, is because it forces a perfectly round ball, without shift it may turn into an egg!
*  From here on out it's the same as with the D sprite, release the mouse button and click somewhere in this area to finish the creation of the ball, just like with the D sprite, the ball should appear inside the project area on the right as well.
*  Your project should look similar to this.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Project2.png)

*  Before we move on to the programing, there's one more step required later on, that we should configure now.

**Step 4: Sprite Center**

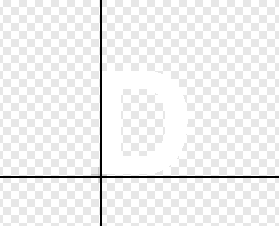
*  We have to set the costume center. Select the D sprite, then, in the far right corner, you will see a cross symbol. select it.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Costume_Center_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Costume_Center_fr.png)

*  Go into the area with the cross, then hold your left mouse button and drag the cross in the bottom left corner of the D sprite. If you can't see the D sprite well enough or if you're shaking too much, try zooming further in, by clicking on the magnifying glass in the bottom right of this area.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Magn_Glass.png)

*  You want to put the cross right here.

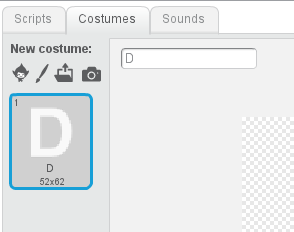
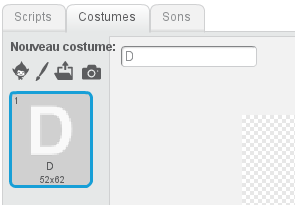
[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/D_Cross.png)

*  Now let go off of the left mouse button, the position of the D will change but that's fine. This happens because you changed where the center of the D sprite is.
*  Remember to use the undo function if something happened (i.e. The D disappeared).
*  Next, do the same with the Ball. The difference here is that you want to put the cross right in the middle of the ball. This should be easier than with the D sprite.
*  Now we got a propper set-up. Let's move on to the programing part shall we?

Chapter 2: Basic juggling game

**Step 1: Transform the D sprite into a platform**

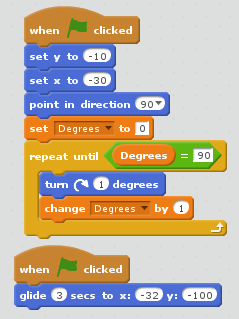
*  Once the game starts (When you click on the green flag above the project area). We want the D sprite to float torwards the bottom and become the platform we juggle the ball with.
*  We were working in the costume tab of the sprites, now we're changing to the script tab. This is where we put the code.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Tabs_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Tabs_fr.png)

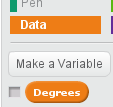
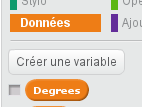
*  The script window should look like this:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Scripts_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Scripts_fr.png)

*  Now, let's make the code, that will let the D sprite spin and float. Select the D sprite and copy the following code:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/D_Turn_Code_fr.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/D_Turn_Code_en.png)

*  In order to completely write this code, a new variable has to be made. Click on Make a Variable under the Data option and call it "Degrees". Make sure to uncheck the variable after you created it, otherwise we will always see its value on screen and we don't want that.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Variable_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Variable_fr.png)

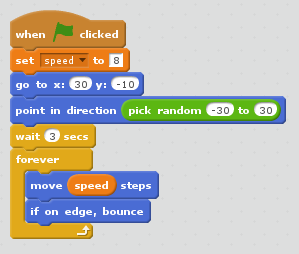
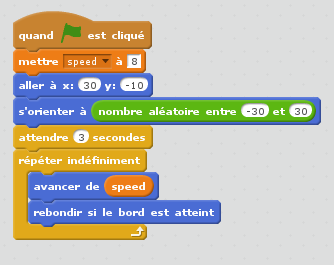
*  Once you click on the green flag above the project window, the D sprite should float slightly torwards the bottom of the area and at the same time to a 90 degree spin to the right.
*  Explanation of the code: When we click on the green flag, we set the D sprite to a specific x and y position which is somewhere in the middle. The x axis is basically the horizontal axis, where as the y axis is the vertical axis. If you move your mouse through the project area, you can see the axis of your mouse in the bottom right of the project area.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/axis.png)

*  The pointing into direction part means it's pointing straight right. 0 would mean it would face straight upwards.
*  Then we set the "Degrees" variable we previously created to 0, meaning we reset it every time we reset the game.
*  The big "repeat until" thing that surrounds other code is a loop, meaning whatever is inside is being repeated until the criteria is met. In other words, we repeatedly turn the D sprite by 1 degree to the right and then add 1 to the "Degrees" value until the variable "Degrees" equals 90. Simply put, we turn the D sprite 90 times by 1 degree really fast. After 90 times, the variable "Degrees" has the value 90, so the loop criteria is met and it stops.
*  From a seperate starting point we then let the D sprite glide for 3 seconds to a predetermined point closer to the bottom of the project area. This has to be seperate because if we add it to the first block of code it will either turn or glide first, then do the other. This way it does both at the same time.

**Step 2: Make a bouncing ball**

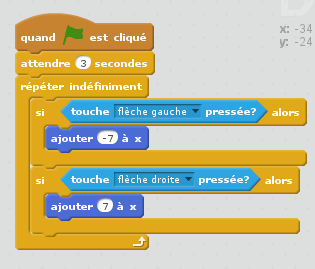
*  Select the ball sprite and add the following code to let your ball freely jump around:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Base_Ball_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Base_Ball_fr.png)

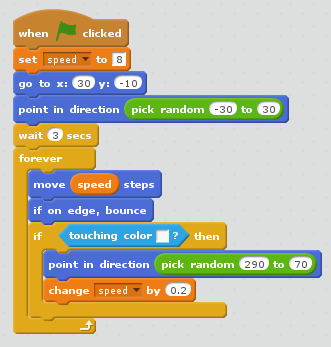
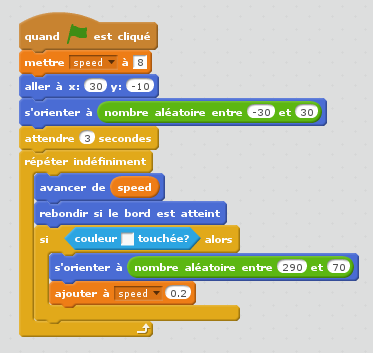
*  Explanation of the code: When the flag is being clicked, we set the speed to the base value 8, you probably already guessed that this is a new variable you have to create, just like you did with "Degrees".
*  Then, just like with the D sprite, you assign a fixed position, that the ball will always return to when clicked on the green flag.
*  We let the ball face a random directon from -30 degrees to +30 degrees, which is basicly a cone facing straight up.
*  Then we wait 3 seconds. We do this so the ball doesn't start bouncing before the D sprite is in position.
*  Inside another loop, simply a loop that lasts forever as it says on it, we let the ball move at "speed" steps, so in other words 8, since we set the "speed" to 8 in the beginning.
*  And of course we let it bounce with the next line of code whenever it hits a wall.

**Step 3: Collision & Movement**

*  Let's program controls for the platform and that whenever the ball touches the platform it bounces off.
*  Select the D sprite and add the following code to add controls to the platform, so you can move it left and right:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Controls_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Controls_fr.png)

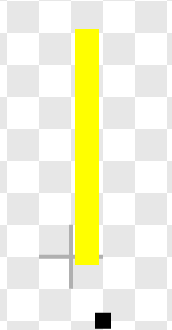
*  Explanation of the code: After the game starts we wait 3 seconds first, to let the platform glide into position.
*  Now in a forever loop, we use a if statement, meaning that if something happens, something else is being executed. In this case we check "forever" since it's in a loop that if "left arrow" is being pressed, we change x, so the horizontal axis by -7 to move the platform to the righ. It's obviously the exact opposite if we want to move to the right, just that the x value here has to be positive.
*  Select the ball sprite and add the following code to the code from before to let the ball bounce off of the platform:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Ball_Bounce_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Ball_Bounce_fr.png)

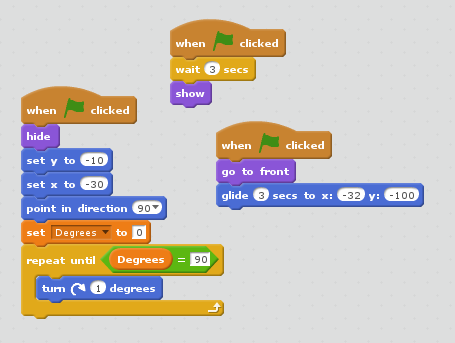
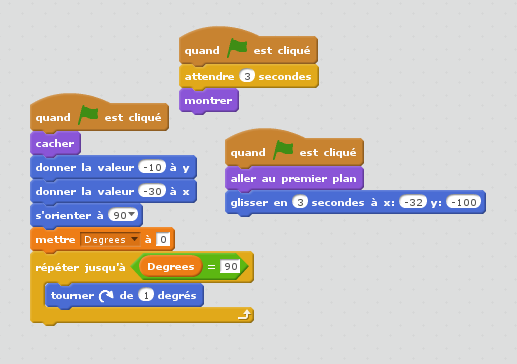
*  The game should be playable now.... Somewhat. The Ball is probably going to react strangely upon collision with the platform or even the edge. The reason for this is because we check for collision with something white and the edge of the arena is white too.
*  Explanation of the code: We added a if statement that if the colour of the D sprite (white) is being touched then we change the direction to a random number somewhere in a cone facing upwards, additionally, we increase the speed slightly, so the ball becomes faster and faster.

**Step 4: Refining ball bouncing**

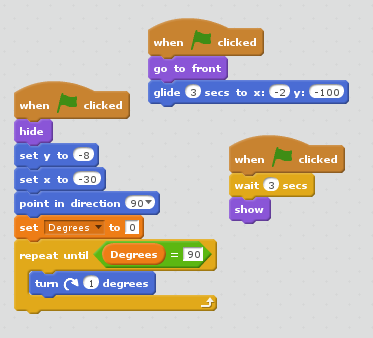
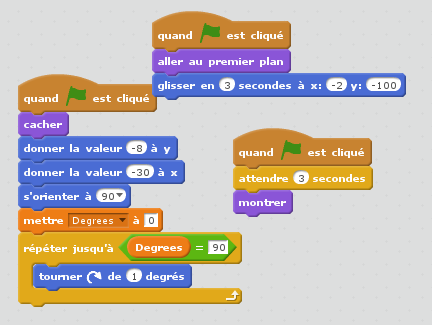
*  Since the bouncing ball seems to be a bit out of control, let's add some additional code, to make the collision and bouncing a bit smoother.
*  Add a new sprite. This new sprite should be a yellow line that should be about 1 pixel wide and 7 pixel long. Pixels are shown as grey/white rectangles in the costumes window if you zoom in as shown in the picture below.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Yellow_Bar.png)

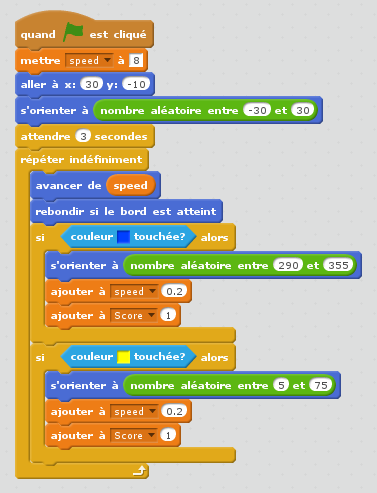
*  Next, do the same as above, but this time make it a blue bar. For both, set the costume center cross to their bottom right just like you did with the D sprite.
*  Now, we want to add some code to those bars. Add the following code to the blue bar.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Blue_Bar_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Blue_Bar_fr.png)

*  You probably noticed that this code is similar to the one that lets the D sprite turn and float down, this code essentially does the same with the blue bar, just that the bar is first invisible and appears once the platform is in position.
*  Now you want to add the same code to the yellow bar, since this bar isn't at the exact same position, we have to change the output position slightly as shown here:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Yellow_Bar_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Yellow_Bar_fr.png)

*  Next up, we change the previous collision detection from white to blue, and then add another collision detection for yellow.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Bounce_Bars_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Bounce_Bars_fr.png)

*  The only difference between these 2, is the angle. The different numbers mean that if the yellow bar is hit, the ball bounces off more torwards the right and if the blue one is hit, the ball bounces off more to the left.
*  The last thing missing, are the controls for these, since they're essentially stuck to the D sprite, we literally just copy the controls from the D sprite and paste them into the yellow and blue bar sprite code section (use drag and drop for this).
*  You should have a basic juggling game now!

Chapter 3: Upgrade

**Step 1: Lava floor**

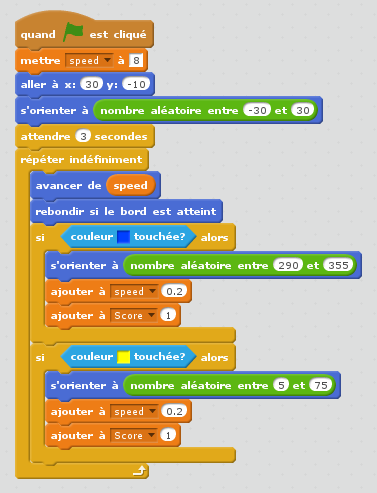
*  We have the base now, but there is no goal or purpose to the game just yet.
*  Usually, in these kind of games, you have to make sure the ball doesn't pass the platform or else you lose. So let's start by programing a dead zone.
*  Create a new sprite just like you did before. Pick the rectangle tool.

[title](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Rectangle.png)

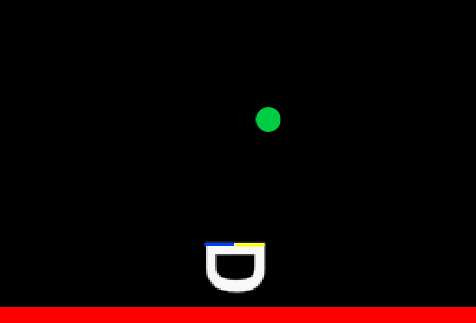
*  Since the dead zone is supposed to be dangerous, pick the colour red to make it look like lava.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Rectangle_Colour.png)

*  Also, remember to make it a full rectangle, not just the outlines.

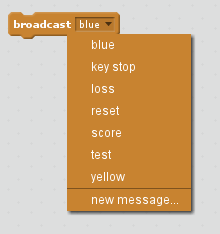
[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Bounce_Bars_fr.png)

*  Now, draw a rectangle, just like you drew all the figures before.
*  Once you drew the rectangle you'll see the little squares again to adjust the size, make it as big as possible to make sure it'll fill the screen. Once you unselect the rectangle it will appear in the project area on the left, drag it torwards the bottom, to make it look like there's a lava floor.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Lava_Floor.png)

**Step 2: Game Over**

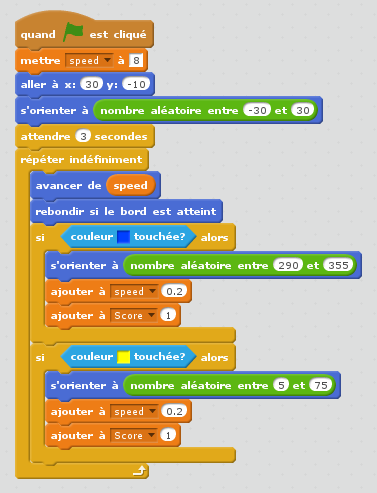
*  Since the game has to end at one point, let's add code, which when the ball touches the lava, initiates a Game Over screen and ends the game.
*  In order to do the next bit, we have to look into broadcasts first:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Broad_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Broad_fr.png)

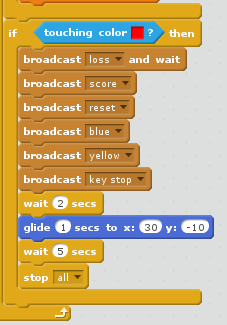
*  This lets you braodcast specific signals. Broadcasting means, you are sending the message stating in the broadcast throughout your entire code. Then on any sprite, you can program code, that will only get executed, once a specific broadcast signal is received.

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Broad2_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Broad2_fr.png)

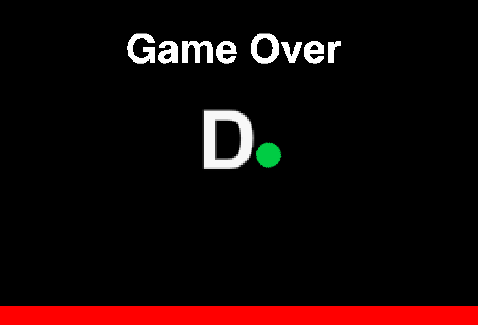
*  Go back to this code on the ball sprite:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Bounce_Bars_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Bounce_Bars_fr.png)

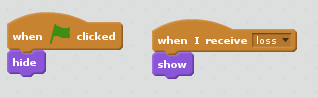
*  Now after the 2 if statements but still inside the forever loop, add the following code:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Touch_Lava_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Touch_Lava_fr.png)

*  We're broadcasting a bunch of different statements which we are going to use soon. We then wait 2 seconds and then the ball will glide back to its beginning position then we wait another 5 seconds for everything to end and then we end the entire code.
*  Add a new sprite. Make it white Text like you did before with the D sprite. Write "Game Over" in white. You can make this bigger too. Then put it in the middle of the screen.
*  It should look like this:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/End.png)

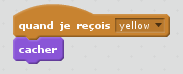
*  Now choose that new Game Over sprite and add this code to it:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Loss_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Loss_fr.png)

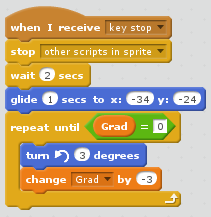
*  Explanation of the code: The way this works is that we programmed in the ball sprite, that if the ball hits the colour red, so in other words the lava, it sends a broadcast with the message "loss" to every part of the code. Now in the Game Over sprite, we programmed that when the message "loss" is received it makes Game Over visible. So the Game Over is always there, it's just invisible until you actually lose, then it appears.
*  With this code, when the game ends, the platform is still movable, let's change that.
*  Go back to the script in the ball sprite. See the "key stop", "blue" and "yellow" broadcasts? We are going to use those now. To stop everything after the game ends.
*  Go to the blue bar sprite and add this code:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Blue_End_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Blue_End_fr.png)

*  Do the same for the yellow bar:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Yellow_End_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Yellow_End_fr.png)

*  And add this code to the D sprite:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/D_End_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/D_End_fr.png)

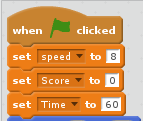
*  Explanation of the code: Once it receives the broadcast it stops all the other blocks of code in this sprite. This means that the block where we programmed the controls is going to stop working. Then it waits 2 seconds and finally puts the D sprite back into its initial position.

**Step 4: Score and Timer**

*  Finally, lets add a timer, the current score and the top score to the game.
*  Create 3 new variables: Time, Score, Top Score. Make sure the box next to them is checked..

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Fin_Var.png)

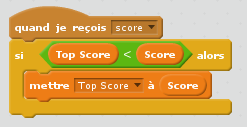
*  They will now appear in your project area. Move them wherever you want.
*  Now go back to the big block of code in the ball sprite, and add this to the top of the code:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Ball_Ini_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Ball_Ini_fr.png)

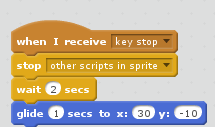
*  This sets the score to 0 and the timer to 60 when a new game begins.
*  Now we add 3 more blocks of code to the ball sprite.
*  The first one is the following:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Timer_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Timer_fr.png)

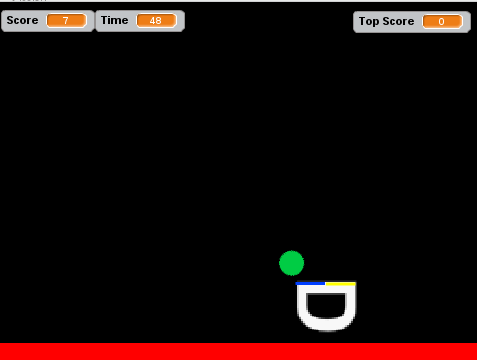
*  This code is responsible for how the timer works.
*  Explanation of the code: After 3 seconds the timer starts ticking down, until the ball hits the lava. Inside the loop we wait 1 second and then change the time by -1. Since it's a loop this means that every second the timer is going down by 1. Then the if statement if "Time" is equal to 0, so if the timer hits 0 it executes the same broadcasts that end the game as if we touched the lava, so it's saying we lost basically.
*  Next we save the top score with the following code:

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Score_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Score_fr.png)

*  Explanation of code: The if statement checks that when the game ends, if the current score is higher than the top score, the top score will become what the current score was.
*  The final bit of code is this:

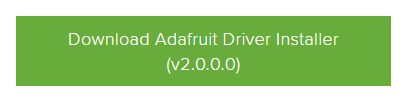
[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Stop_en.png) [](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Stop_fr.png)

*  All the extra functions should work now. Your complete juggling game is now done!

[](https://github.com/JeyAl/internship/blob/master/Scratch_Images/Final.png)

# Anhang: Light Painting

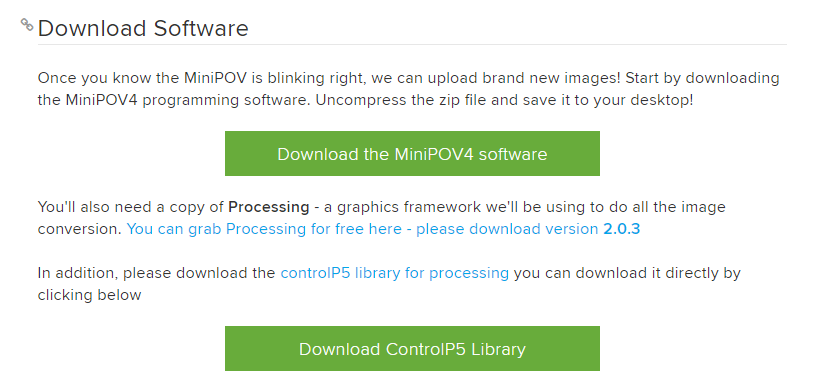
MiniPOV4 Configuration Guide

Fist, you want to visit the following Website:  
<https://learn.adafruit.com/minipov4-diy-full-color-persistence-of-vision-light-painting-kit?view=all>  
On this Website, scroll down to the "Install Windows Drivers" section.  
Here, you want to click on this link:  
[title](https://github.com/JeyAl/internship/blob/master/images/drivers_link.PNG)  
and then download the required drivers:  
[](https://github.com/JeyAl/internship/blob/master/images/drivers_download.PNG)

Follow the installation instructions of the drivers and install the drivers that are selected by default.  
After you installed the driver, plug in the MiniPOV4 while being turned off, then when it is plugged in, turn it on. In your device Manager, there should be a USBTiny device.

[title](https://github.com/JeyAl/internship/blob/master/images/USBTiny.PNG)

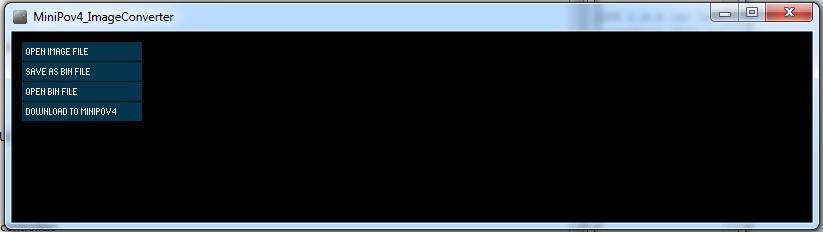
Afterwards, scroll a bit further down to the "Download Software" section.  
Download the 2 highlighted files, shown as green buttons:

[](https://github.com/JeyAl/internship/blob/master/images/downloads.PNG)

And download Processing by clicking on the blue button in the same section:

[title](https://github.com/JeyAl/internship/blob/master/images/processing.PNG)

Unpack all these files into a Folder on your Desktop.  
When you're done downloading, open up processing and quit again.  
This will create a folder "Processing" in your Documents.  
Inside this folder, create a folder "libraries" if it doesn't exist yet.  
Then put the folder "controlP5" that you previously downloaded in the "libraries" folder.  
So the path should be "My Documents/Processing/libraries/controlP5/library.properties".  
Now open Processing again. Click on "File" and then "Open..." and then select the MiniPov4\_ImageConverter Folder that you previously downloaded and unpacked and then select the file with the same name.  
Next, click on the run button in the top left below the "File" tab.  
The following window should appear:

[](https://github.com/JeyAl/internship/blob/master/images/converter.PNG)

Now click "open image file" and select your desired image. Then click on "download to minipov4" and it should successfully save the pattern to the minipov!