



Projet de Réalité Augmentée

Incrustation d'objets virtuels sur des vidéos en temps réels

Conditions	Projet Durée : 28 h	Moyens	<ul style="list-style-type: none"> • Poste informatique sous Windows • Logiciels PROCESSING
Prérequis	TD 1 ^{er} pas en Réalité Augmentée	Niveau	Classe de Terminale S spécialité ISN
Compétences	<ul style="list-style-type: none"> • C1.1 Justifier dans une situation donnée, un codage numérique ou l'usage d'un format approprié, qu'un programme réalise l'action attendue... • C2.3 Développer une interface logicielle ou une interface homme-machine, un algorithme, un programme, un document ou fichier numérique... 		
Éléments du programme	<ul style="list-style-type: none"> • Représentation de l'information <ul style="list-style-type: none"> ○ Structuration et organisation de l'information • Algorithmes <ul style="list-style-type: none"> ○ Algorithmes simples 		

Table des matières :

1. Qu'est-ce que la réalité augmentée ?.....	2
A- Définition.....	2
B- Domaines d'applications	2
C- Principe.....	2
2. Projet ISN – Réalité Augmentée	3
A- Activité pédagogique proposée en projet.....	3
B- Outil de développement	4
a. Installation.....	4
b. Jeu d'instructions.....	5
C- Les marqueurs	6
D- Le programme développé	9
3. Pistes d'évolution du projet.....	14
Mes expérimentations.....	14

1. Qu'est-ce que la réalité augmentée ?

A- Définition

La réalité augmentée désigne les différentes méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans une séquence d'images. Le terme augmenté est évoqué car notre perception visuelle est modifiée par l'implémentation d'un modèle virtuel en 2D ou 3D dans le monde qui nous entoure, en temps réel.

B- Domaines d'applications

Le premier exemple de réalité augmentée au cinéma daterait de 1993 par le film Jurassic Park de Steven Spielberg. Des images de synthèse représentant des dinosaures étaient ajoutées en temps réel dans le décor du parc d'attraction. D'autres grands projets cinématographiques ont permis la reconnaissance de la réalité augmentée comme Le Seigneur des Anneaux ou Star Wars. Pour la petite anecdote, Georges Lucas le réalisateur de la double trilogie Star-Wars utilisait des hamburgers et balles de ping-pong sur fond vert afin de modéliser les vaisseaux spatiaux aux yeux des caméras.

(source <http://leculturographe.wordpress.com>)

De nos jours la réalité augmentée prend de plus en plus de place, s'essaye à tous les domaines possibles : jeux vidéo, marketing, Art, Cinéma, réseaux sociaux mais aussi assistance à la personne



Exemple d'application marketing

L'application permet aux utilisateurs de choisir l'un des modèles du catalogue et de le superposer, en ajustant sa taille, sur une image prise par le capteur photo de l'appareil (Smartphone).

En savoir plus :

- <http://www.larealiteaugmentee.info>
- <http://tpe-realite-augmentee.perso.sfr.fr/projetsixthsense.html>

C- Principe

Les scènes réelles sont capturées par une caméra (simple webcam ou téléphone portable) et transmettent à une unité de traitement.

Cependant pour faire de la réalité augmentée il faut localiser le virtuel dans le réel avec le même point de vue. Pour cela des marqueurs peuvent être placés dans la scène réelle. Ils permettent de calculer les coordonnées 3D à partir de 3 points spécifiques reconnus par le système.

La difficulté du traitement informatique est alors de faire coïncider une caméra virtuelle avec la position de la caméra réelle.

2. Projet ISN – Réalité Augmentée

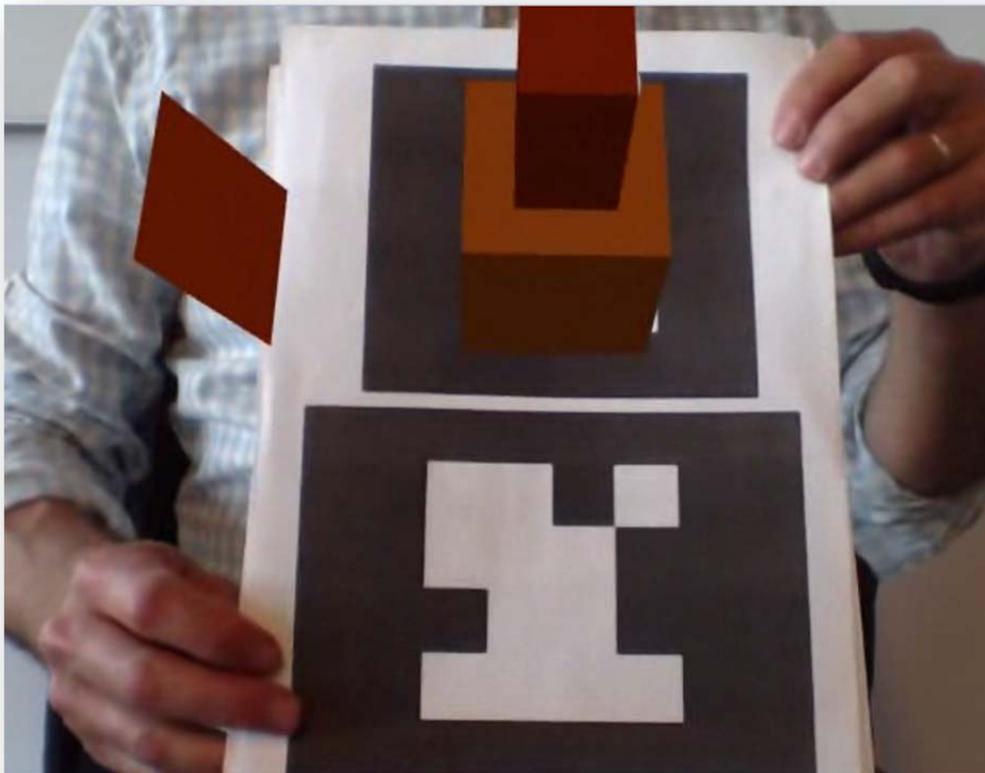
A- Activité pédagogique proposée en projet

Thème d'étude : Conception d'une vidéo en réalité augmentée

Nature de la production finale attendue : Conception d'une vidéo en réalité augmentée générée par le logiciel Processing. Cette vidéo doit présenter une visite partielle du lycée Jean Perrin avec incrustation à partir de marqueurs d'images générées par du code.

Cahier des charges :

- Cette vidéo doit comporter au minimum 4 marqueurs.
- Chaque élève doit concevoir le code d'au minimum 1 image déclenchée par 1 marqueur.
- Chaque élève doit fournir un document détaillé expliquant la démarche, les difficultés rencontrés, le détail et le rôle du code utilisé.
- Une cohérence du choix des images vis-à-vis de la production demandée est exigée.



B- Outil de développement

Ce projet utilise le logiciel Processing :

C'est un langage de programmation et un environnement de développement graphique. Son code est un dérivé de Java. Les programmes réalisés avec Processing peuvent être lus par les navigateurs internet équipés du plug-in java, mais aussi sous forme d'applications indépendantes pour Windows, Linux ou mac (en réalité n'importe quelle machine disposant d'une machine virtuelle Java).

ainsi que les librairies suivantes :

- ARToolkit nécessaire pour la reconnaissance de marker
- Java
- Opengl
- Gsvideo nécessaire à la capture et l'affichage d'un flux vidéo

a. Installation

Installer Processing-1.5.1 (uniquement cette version et non la dernière qui ne fonctionne pas avec les fichiers donnés) et la webcam extérieure (dans le cas où vous rajoutez une webcam).

1) Il faut rajouter des librairies d'Artoolkit, pour cela télécharger :

Le nyar4psg-1.1.6 qui se trouve sur ce site :

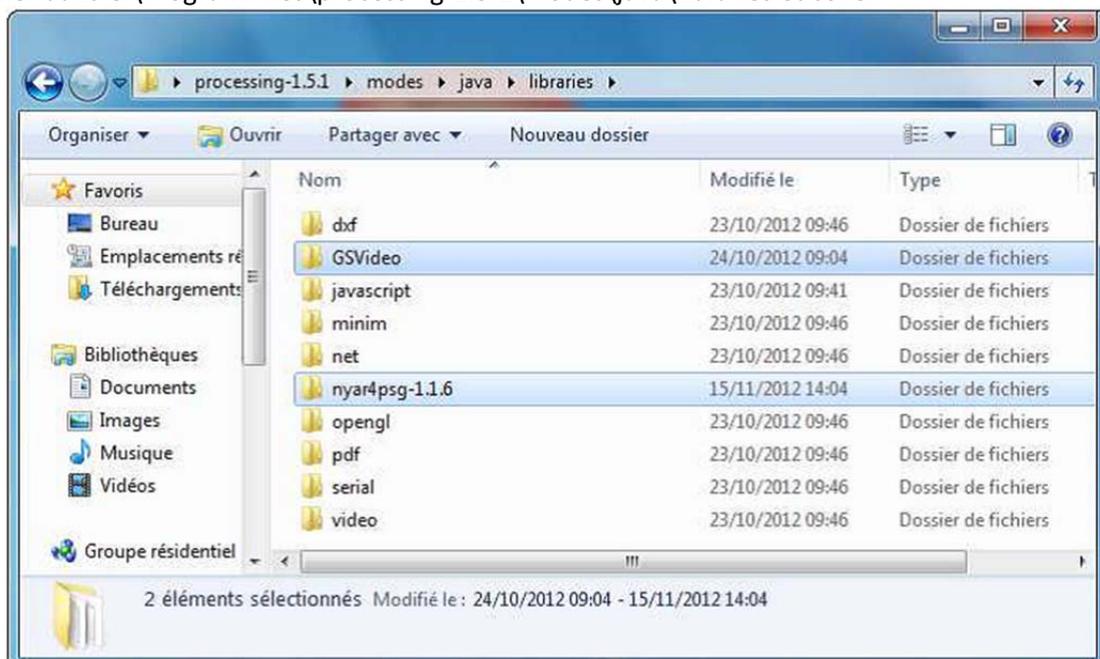
<http://es.sourceforge.jp/projects/nyartoolkit/downloads/52564/nyar4psg-1.1.6.zip/>

Le GSVideo, sur le site :

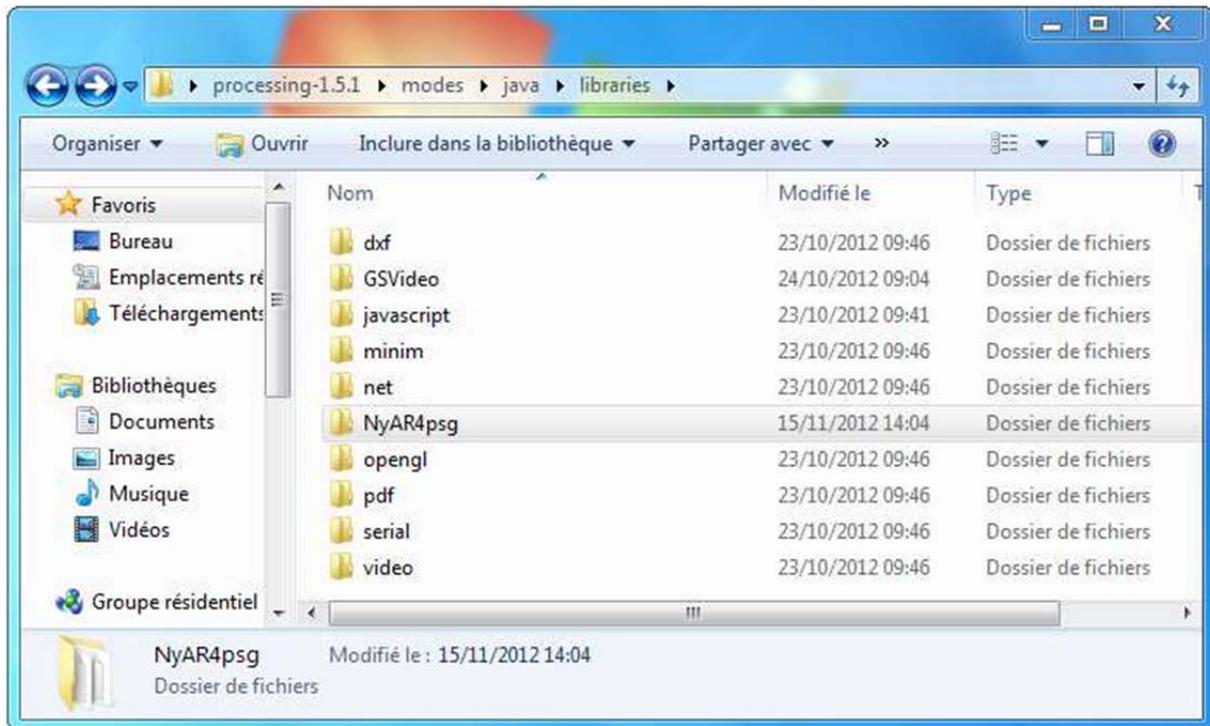
<http://sourceforge.net/projects/gsvideo/files/gsvideo/1.0/GSVideo-1.0.0-windows.zip/download/>

ou récupérer les librairies NyAR4psg et GSVideo dans la répertoire « librairies »

2) Aller dans C:\Program Files\processing-1.5.1\modes\java\libraries et coller.



3) Renommer le dossier « nyar4psg-1.1.6 » en « NyAR4psg ».



4) Vérifier que NyAR4psg et GSvideo soit bien dans la librairie de Processing.

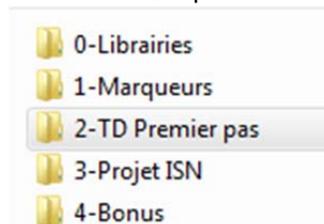
b. Jeu d'instructions

Le programme utilise un certain nombre de fonctions dont certaines reviennent assez souvent :

- **import** : permet l'importation de librairie
- **void** : indique que la fonction ne renvoie aucune valeur
- **println** : affiche dans la zone de texte l'évolution d'une, ou plusieurs variables
- **text** : fonction d'affichage de texte
- **loadImage** : fonction de chargement d'une image (appelle une image)
- **fill (R,G,B)** : permet le changement de couleur d'une fonction telle que « text »

Voir pour exemple la description d'un programme gérant

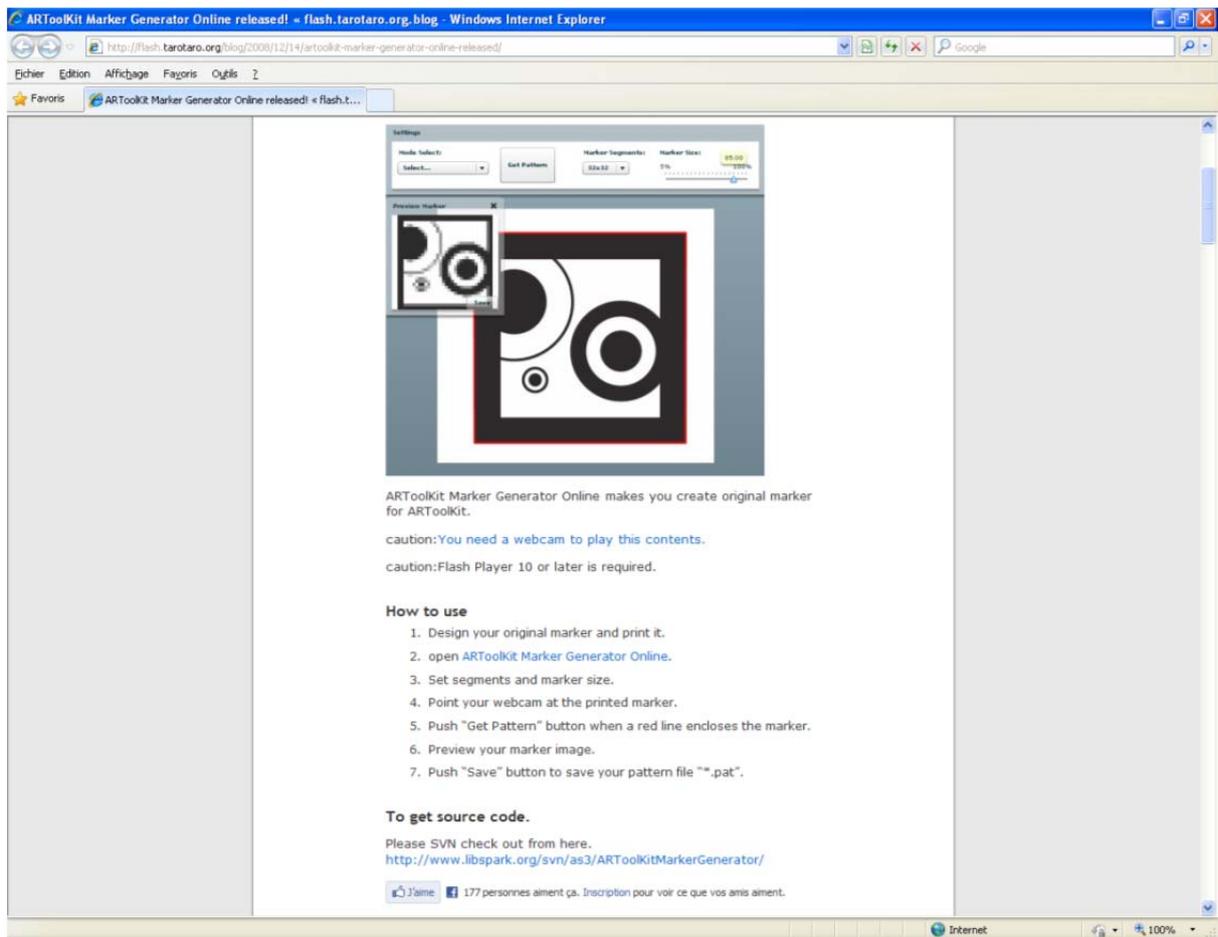
Dossier 2-TD Premiers pas



C- Les marqueurs

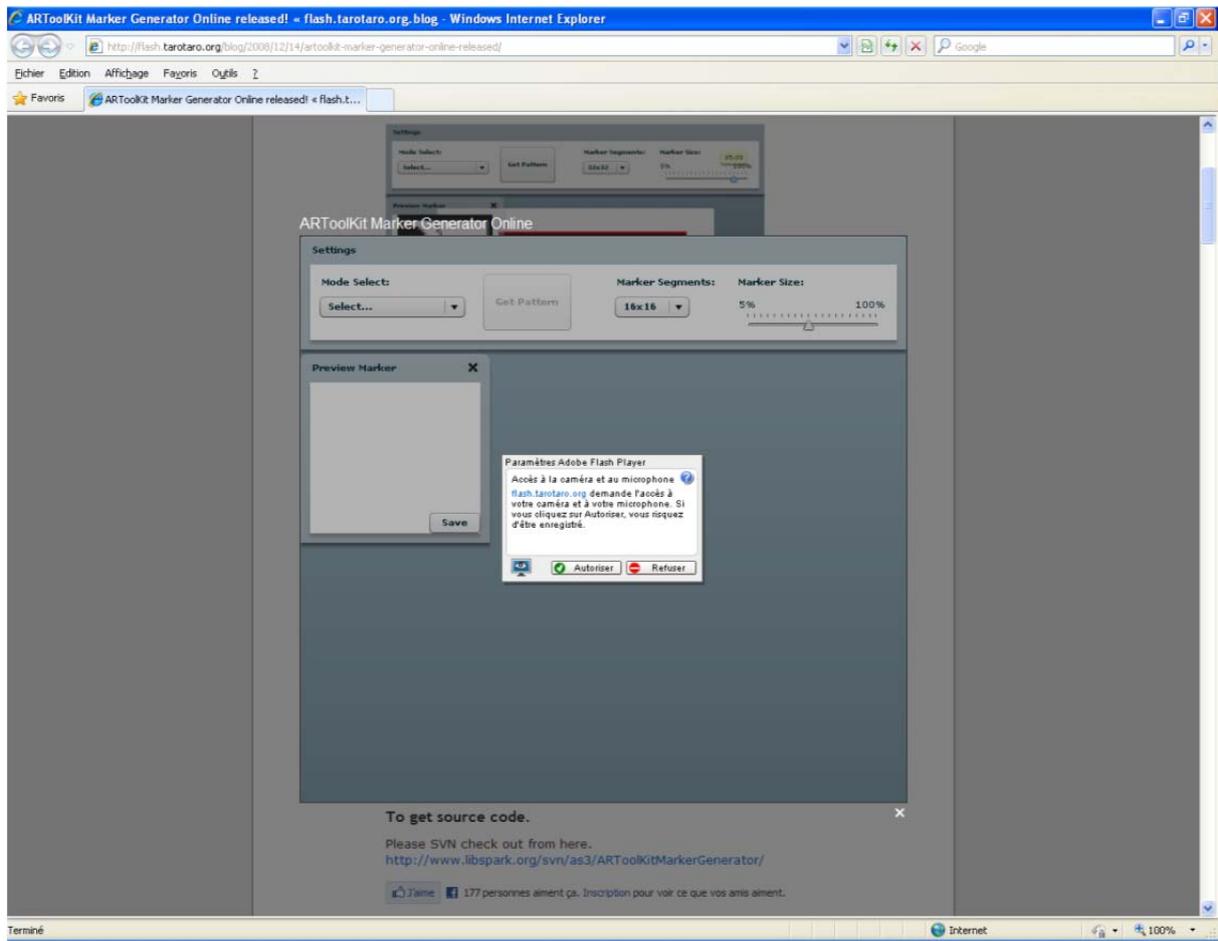
Il faut créer ou récupérer un pattern au format .gif pour le soumettre à la détection de la Webcam (vous trouverez un nombre important de .gif associé à des .patt dans le dossier « marqueur »)

En premier lieu suivre l'URL <http://flash.tarotaro.org/blog/2008/12/14/artoolkit-marker-generator-online-released/> (ctrl+clik gauche pour ouvrir l'URL)



Cliquer sur le lien présent au point n°2 « ARToolkit Marker Generator Online » <http://flash.tarotaro.org/ar/MarkerGeneratorOnline.swf> (ou suivre le lien précédent).

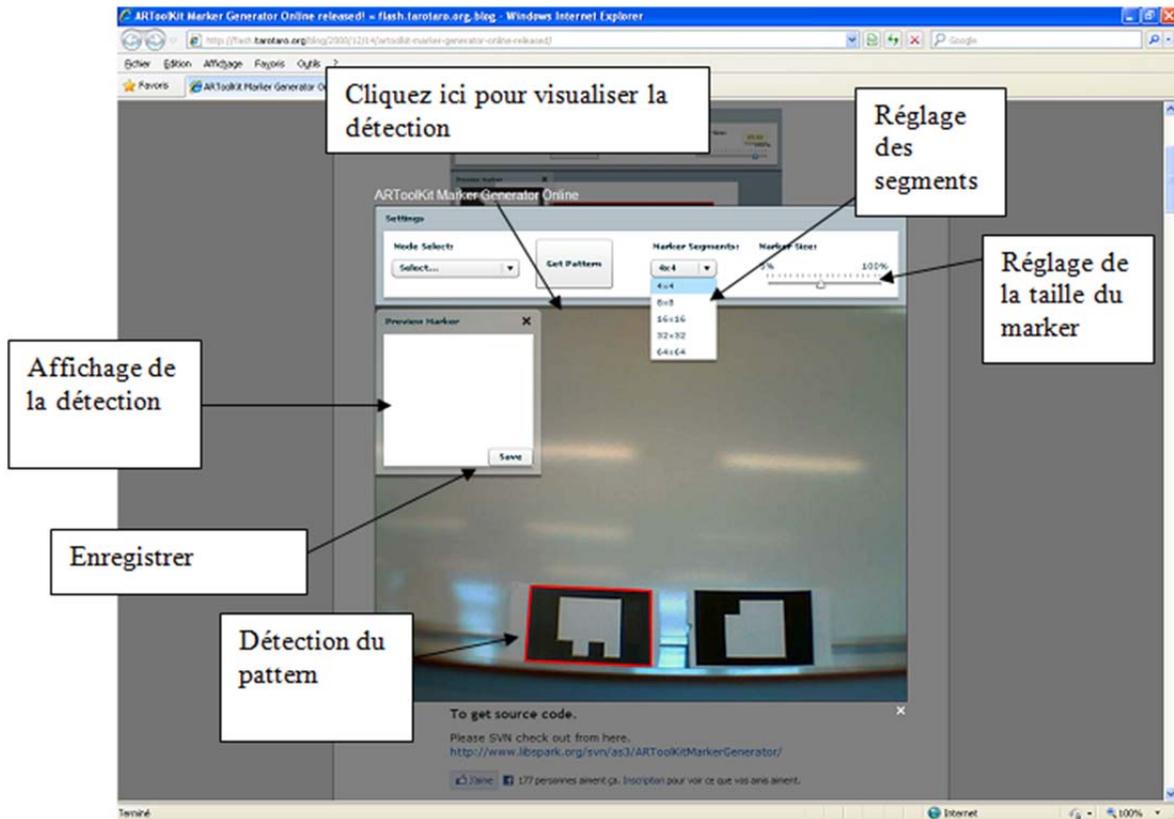
Une fenêtre flash se lance.



Cliquer sur « Autoriser » afin que la Webcam ait accès à la page flash.

Des latences peuvent survenir lorsque cette fenêtre est ouverte, il est possible de définir la mémoire allouée en faisant un clic droit sur la partie grisée et en choisissant « paramètre ».

Paramétrage du pattern



Régler « Marker Segments » sur la précision désiré (4x4 faible précision, 64x64 plus grande précision), le chiffre définissant le nombre de pixel.

Une fois le pattern détecté, c'est-à-dire entouré de rouge, cliquez sur « GetPattern », le carré « Preview Marker » affiche la reconnaissance du marker. Si elle n'est pas satisfaisante, il est possible de régler la détection en jouant sur le curseur « Marker Size ».

Enregistrer le fichier .pat dans le répertoire « data » en cliquant sur « save » et le renommer en .patt

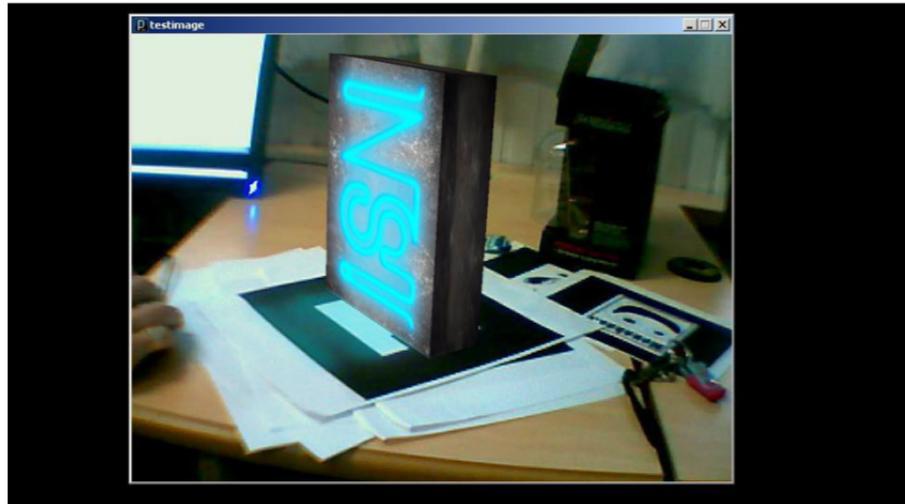
Le fichier .gif doit exclusivement être composé de noir et de blanc, réaliser sous un éditeur graphique (paint, photofiltre, gimp,...), les dimensions importent peu.

D- Le programme développé

Description du fichier « FinalCode » - Dossier : 3-Projet ISN

L'objet affiché à partir du marqueur 1, s'est construit à partir d'images nommées néon 1, 2 et 3 créés sur Photoshop le clignotement s'effectue de façon aléatoire grâce à la variable nx (si nx n'est pas égal à 2 le néon est éteint) :

```
beginShape(QUADS);
if (nx==2){
texture(neon2);
}
else{
texture(neon1);
}
vertex(-1, 3, -3,150,700);
vertex(-1, 3, 3,950,700);
vertex(-1, -3, 3,950,150);
vertex(-1, -3, -3,150,150);
endShape();
```



```
if(nx==2)
{
ny=ny+1;
if (ny==nz)
{
nx = int(random (0,5));
ny=0;
nz= int(random(2,20));
}
}
else
{
ny=ny+1;
if (ny==nz)
{
nx = int(random (1,4));
ny=1;
nz= int(random(5,10));
}
}
}
```

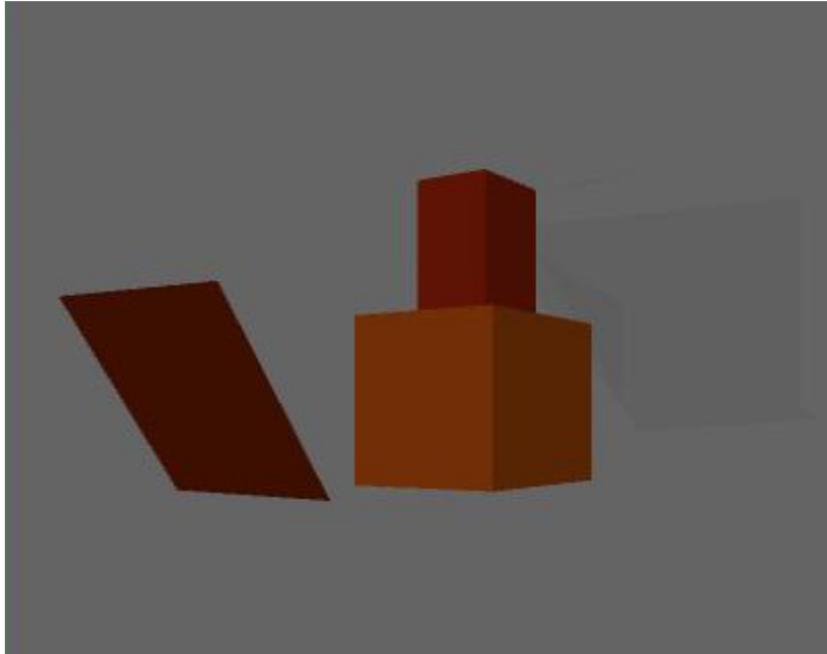
ny: variable qui augmente constamment
sin y est égal à un nombre aléatoire nz, nx est redéfini, c'est ainsi que nz produit cet effet de clignotement du néon avec un tps d'affichage aléatoire (nx nbre entier choisi entre 0 et 5)

nz: entier choisi entre 0 et 20

même principe qu'au dessus avec des paramètres modifiés.

Le clignotement du néon s'effectue de façon aléatoire, on peut régler la probabilité de clignotement en modifiant les paramètres random() de nx et de nz.

L'objet affiché à partir du marqueur 2, représente un projecteur cinématographique se déployant d'une boîte et projetant plusieurs images (décompte avant projection) sur pression d'un interrupteur en réalité augmentée



<code>void dessin(){</code>	Déclaration d'une fonction
<code> lights();</code>	Pour le rendu 3D sur les cubes et les spheres (effet d'ombre)
<code> scale(0.5);</code>	
<code> LUM=int(random(150,255));</code>	Variable LUM qui prend une valeur aléatoire entre 150 et 255
<code> translate(0,0,0);</code>	utile pour l'effet "clignotement" du projecteur
<code> //rotateZ(45);</code>	rotation sur l'axe Z
<code> rotateX(11);</code>	rotation sur l'axe X
<code> //rotateY(-90);</code>	rotation sur l'axe Y
<code> pushMatrix();</code>	
<code> fill(120,50,5,255);</code>	couleur de la boîte
<code> box(80);</code>	1er boîte : 80 pixel de côté
<code> popMatrix();</code>	
<code> pushMatrix();</code>	
<code> translate(0,TR,0);</code>	Partie mobile suivant un axe Y (le translate possède une
<code> fill(100,20,2);</code>	variable TR en y)
<code> box(40,70,40);</code>	
<code> popMatrix();</code>	
<code> pushMatrix();</code>	
<code> rotateX(TR2);</code>	
<code> rotateY(TR3);</code>	
<code> translate(-40-TR4+(TR3*20),-80,41+(TR3*40));</code>	
<code> fill(120,30,0);</code>	
<code> rect(0,41,80,80);</code>	Rectangle représentant le couvercle

```

popMatrix();
pushMatrix();
translate(186-200,-15+TR,0);
if(TR<=-69){
fill(LUM);
}
else{
  fill(200);}
sphere(10);
popMatrix();

```

mobilité en même temps que l'autre rectangle
L'animation est basée sur cette variable TR, ici le code vérifie si le projecteur est effectivement sorti de sa boîte

```

if(TR<=-68){
pushMatrix();
scale(0.2);
rotateX(0);
rotateY(33+PI);
rotateZ(0);
translate(-500+x,-900+y,1300+z);
if(nc<=10){
image(trois,0,0);
nc=nc+1;
images
}
if(nc>=10&&nc<=20){
  image(deux,0,0);
nc=nc+1;
images
}
if(nc>=20){
  image(un,0,0);
nc=nc+1;
images
}
if(nc>=30){
nc=0;
}
popMatrix();
}
else{nc=0;}
}

```

Affichage de l'image « trois » placée dans « data »
Incrémentation de la variable nc pour le décompte des

Affichage de l'image « deux » placée dans « data »
Incrémentation de la variable nc pour le décompte des

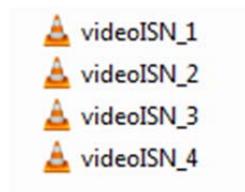
Affichage de l'image « un » placée dans « data »
Incrémentation de la variable nc pour le décompte des

```
void mouvement(){
```

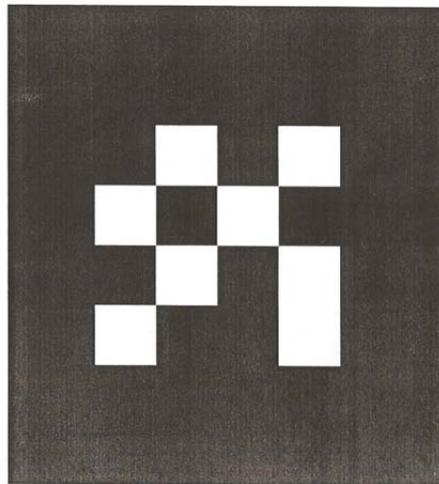
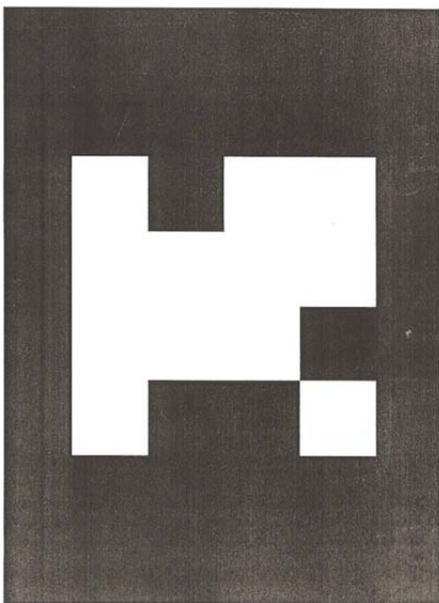
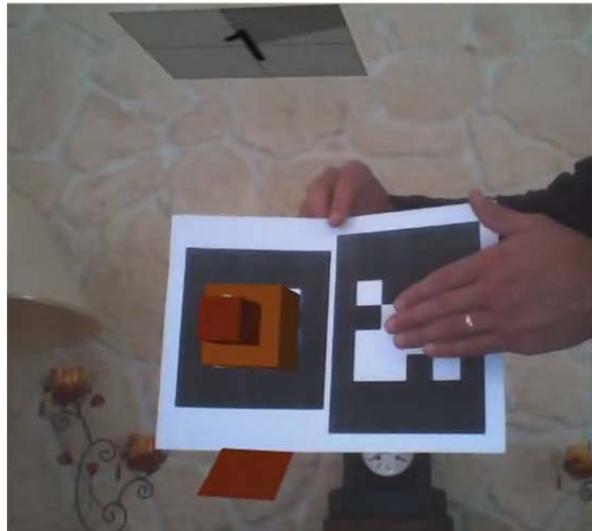
```
    if(KP>=2){  
        if(TR>=-70){  
            TR=TR-3;  
        }  
        if(TR3<=2.1){  
            TR3=TR3+0.04;  
            TR4=TR4+2.2;  
        }  
    }  
    else{  
        if(TR<=-1){  
            TR=TR+3;  
        }  
    }  
}
```

Fonction mouvement qui permet au projecteur d'avoir sa partie supérieure de relevée.

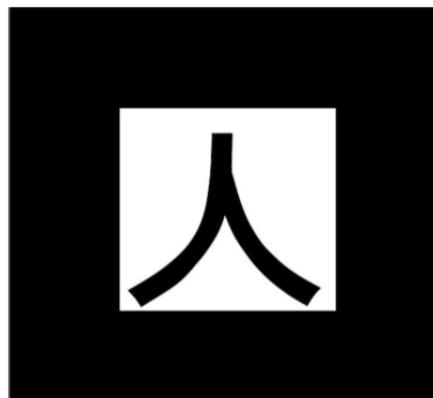
Voir les vidéos :



Marqueurs qui déclenchent l'animation 3D du projecteur avec décompte
 (il faut cacher le marqueur de droite pour déclencher l'animation) :



Marqueur qui déclenche l'affichage de l'image ISN en néon clignotant. :



3. Pistes d'évolution du projet

Mes expérimentations

Description fichier « bonus »

La vidéo qui se trouve dans le répertoire « bonus » montre un objet 3D qui apparaît sur un marqueur, et sur un autre marqueur un des scripts qui réagit avec le son.

Pour que ce fichier fonctionne il faut télécharger la librairie spécifique objet 3D nommée OBJLoader (ou la prendre dans le répertoire « librairie ») qui permet de lire des format.obj crée par des logiciels 3D comme Blender et d'afficher l'image 3D correspondante.

Placer cette librairie toujours au même endroit c'est-à-dire dans :
program files/processing1.5.1/modes/java/librairies.

Ce fichier fait appel à 2 marqueurs nommés m1.patt et m2.patt qui sont placés dans le data (voir les commentaires au niveau du fichier Processing).

