

**EVI 2015**

# **EVI 07: Realidad Aumentada con Detección de Marcadores Utilizando OpenCV**



Miguel Ángel Astor  
Ana Verónica Morales

# Objetivos

- Objetivo general:
  - Introducir y practicar los conceptos fundamentales para el desarrollo de algoritmos y aplicaciones que utilicen técnicas de Realidad Aumentada en el lenguaje C++.



# Objetivos

- Objetivos específicos:
  - Presentar las nociones fundamentales de programación con OpenCV y su integración con OpenGL.
  - Presentar un algoritmo para detección de marcadores.
  - Desarrollar el algoritmo presentado en el lenguaje C++ utilizando la biblioteca OpenCV.
  - Mostrar la integración del algoritmo desarrollado con una aplicación básica de despliegue de modelos 3D.



# Programa

- 1) Introducción a la Realidad Aumentada.
- 2) Introducción a la programación con OpenCV.
- 3) Detección de marcadores.
- 4) Nociones de despliegue en Realidad Aumentada.



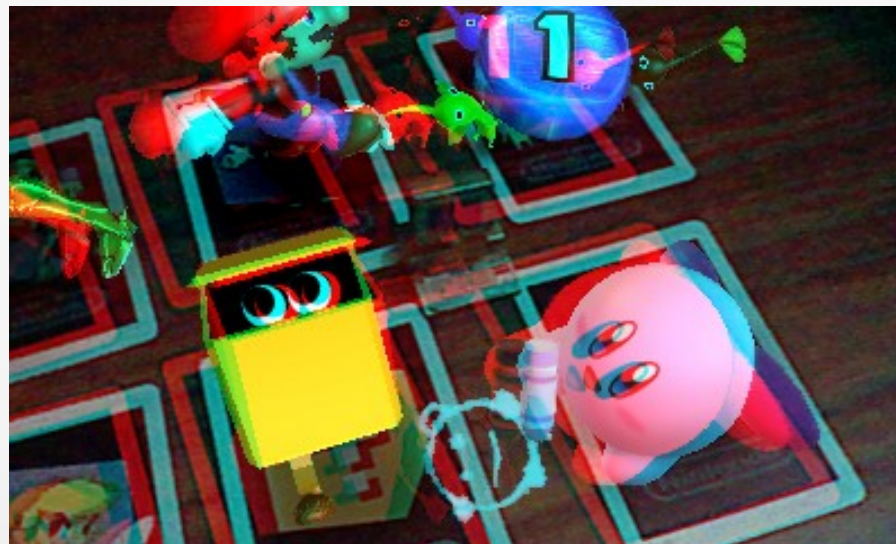
# Punto 1

## Introducción a la Realidad Aumentada



# Introducción a la Realidad Aumentada

- Realidad Aumentada:
  - Complementar la percepción real del usuario con objetos virtuales.

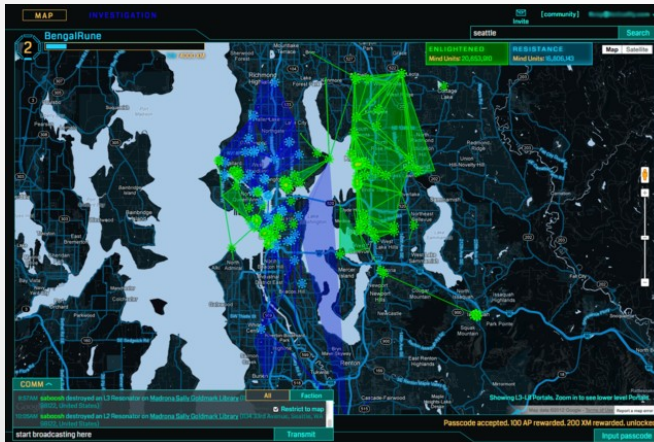


AR Games  
Nintendo 3DS

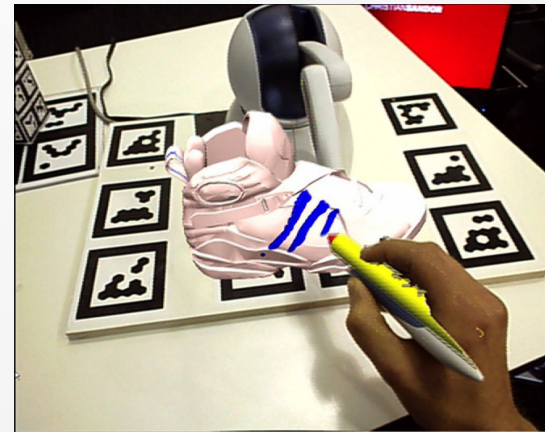
# Introducción a la Realidad Aumentada

- La Realidad Aumentada no se limita al aspecto visual:
  - Permite toda clase de sensores y efectores.
  - Sensaciones virtuales táctiles y auditivas también son posibles.

Ingress

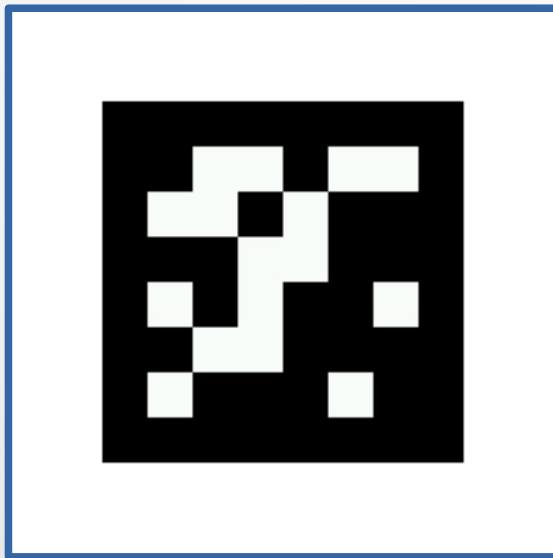


HARP



# Introducción a la Realidad Aumentada

- Tipos de Realidad Aumentada visual:
  - Realidad Aumentada Basada en reconocimiento de marcadores.
  - Realidad Aumentada sin marcadores.





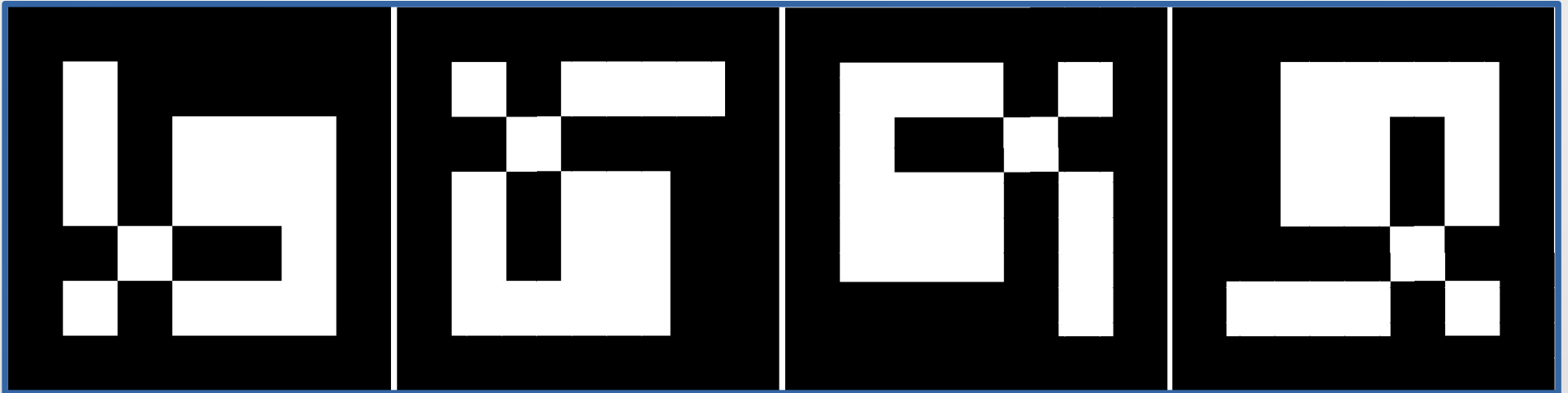
# Introducción a la Realidad Aumentada

- Realidad Aumentada basada en marcadores:
  - Los marcadores se utilizan para el registro de objetos reales.
  - Estos son fáciles y rápidos de detectar e identificar.
  - Pueden usarse para codificar información.



# Introducción a la Realidad Aumentada

- Características de los marcadores:
  - Alto contraste entre el marcador y el fondo.
  - Códigos arbitrarios.
  - Forma regular (principalmente cuadrada).
  - Invariantes a la rotación.

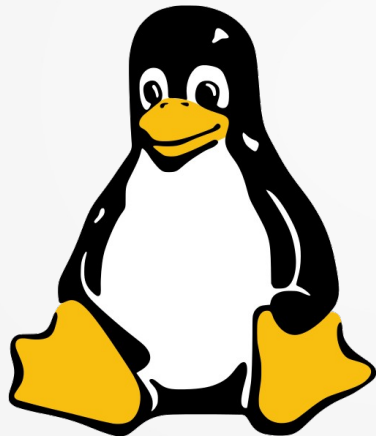


# Introducción a la Realidad Aumentada

- Solución de Realidad Aumentada con marcadores:
  - 1) Calibración de la cámara.
  - 2) Captura de imágenes.
  - 3) Para toda imagen:
    - 1) Detección de los marcadores.
    - 2) Decodificación de los marcadores.
    - 3) Estimación de pose.
    - 4) Despliegue de objetos virtuales.

## Punto 2

# Programación con OpenCV



# Programación con OpenCV

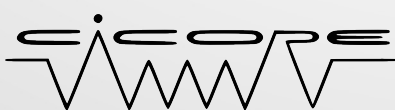
```
typedef std::vector<cv::Point2f> points_vector;

int main(void) {
    struct stat      s;
    FILE *          f;
    vector<char>     buffer;
    vector<unsigned char> out_buffer;

    ////////////////////////////////////////////////////////////////////
    // Declare the matrices.
    ////////////////////////////////////////////////////////////////////
    cv::Mat A, B, C;

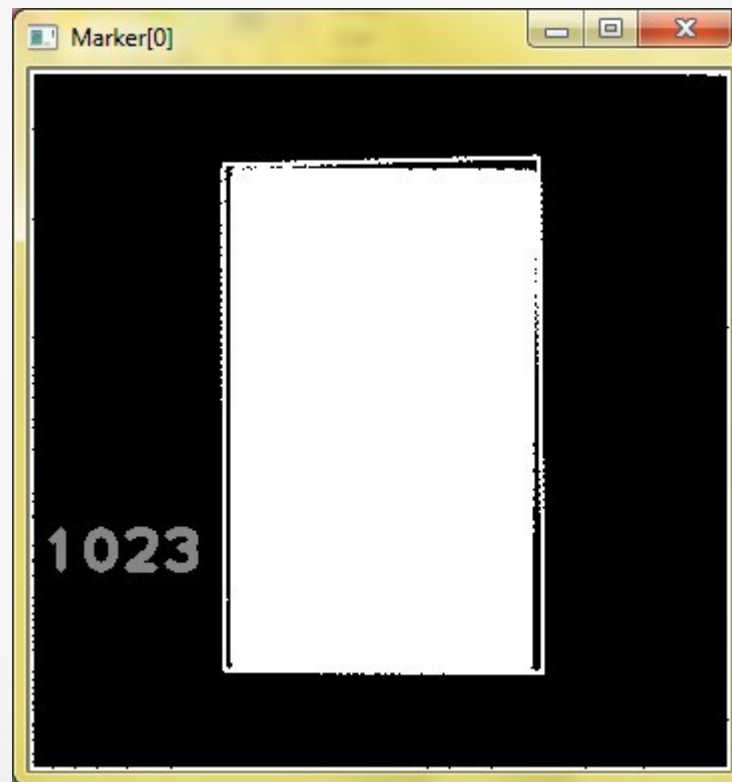
    ////////////////////////////////////////////////////////////////////
    // Load two images using the simplest method.
    ////////////////////////////////////////////////////////////////////
    A = cv::imread("queen1.jpg");
    B = cv::imread("queen2.jpg");

    ////////////////////////////////////////////////////////////////////
    // Load an image from a byte array.
    ////////////////////////////////////////////////////////////////////
    stat("queen3.jpg", &s);
```



# Punto 3

## Detección de Marcadores

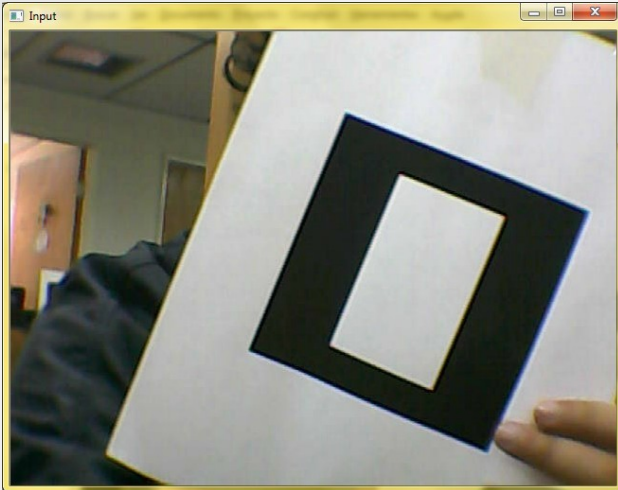


# Detección de Marcadores

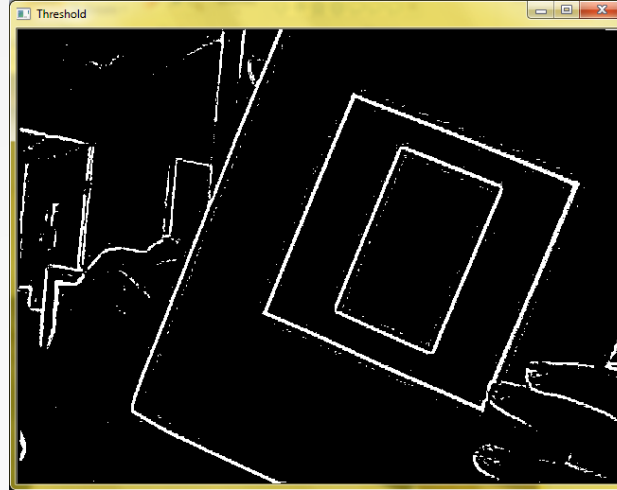
- Pasos:
  - 1) Binarizar la imagen de entrada.
  - 2) Detectar contornos en la imagen binarizada.
  - 3) Reducir contornos a polígonos.
  - 4) Seleccionar candidatos.
  - 5) Para todo candidato:
    - 1) Eliminar distorsión de perspectiva.
    - 2) Intentar decodificar.

# Detección de Marcadores

1)



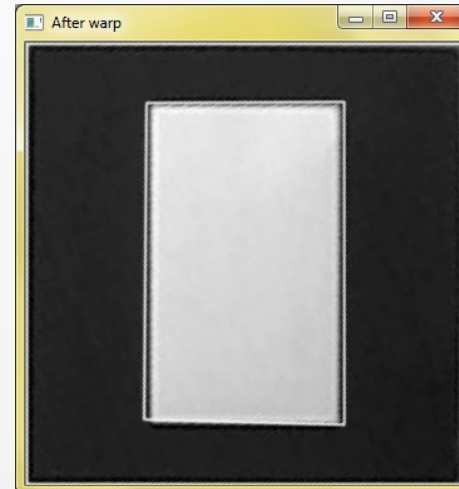
2)



3)



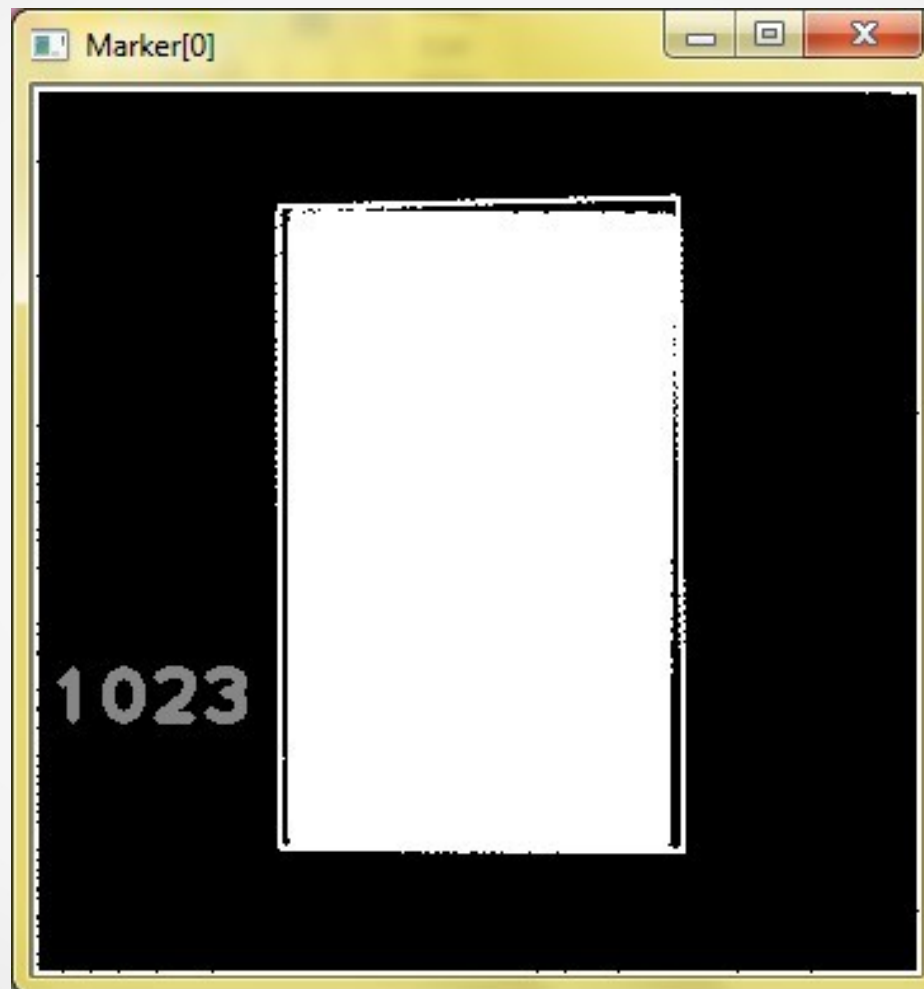
4 - 5.1)





# Detección de Marcadores

5.2)



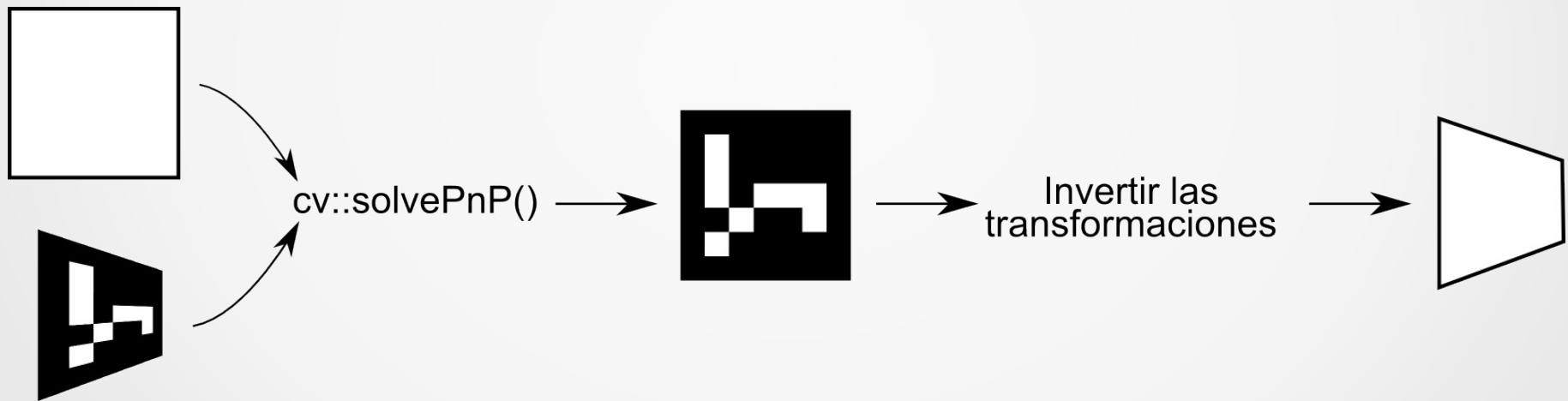
## Punto 4

# Nociones de Despliegue en Realidad Aumentada



# Nociones de Despliegue

- Estimación de pose o *Perspective N-points Problem*:
  - Vector de traslación.
  - Vector de rotación.



# Nociones de Despliegue

- Colocación de objetos virtuales:
  - Matriz de proyección en perspectiva:
    - $\mathbf{F}_{xy}$ : Punto focal de la cámara.
    - $\mathbf{C}_{xy}$ : Centro óptico de la cámara en píxeles.

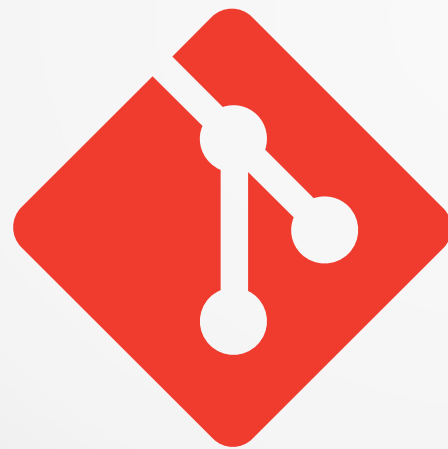
$$\begin{bmatrix} \frac{-2f_x}{w} & 0 & \frac{-2c_x}{w} - 1 & 0 \\ 0 & \frac{2f_y}{h} & \frac{-2c_y}{h} - 1 & 0 \\ 0 & 0 & \frac{-(FAR+NEAR)}{FAR-NEAR} & \frac{-2(FAR \times NEAR)}{FAR-NEAR} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

# Nociones de Despliegue

```
90 ¶
91 » » » if(cvProc.isCameraCalibrated()) {¶
92 » » » » data = cvProc.findMarkersInFrame();¶
93 ¶
94 » » » » if(data != null) {¶
95 ¶
96 » » » » » for(int i = 0; i < ProjectConstants.MAXIMUM_NUMBER_OF_MARKERS; i++) {¶
97 » » » » » » if(data.markerCodes[i] == ProjectConstants.CODE) {¶
98 » » » » » » » monkey.position.set(data.translationVectors[i]);¶
99 » » » » » » » monkey.rotation.set(data.rotationMatrices[i]);¶
100 » » » » » » » monkey.applyWorldTransform();¶
101 » » » » » » » monkey.setVisible(true);¶
102 » » » » » » }¶
103 » » » » » }¶
104 ¶
105 » » » » » focalPointX = cvProc.getFocalPointX();¶
106 » » » » » focalPointY = cvProc.getFocalPointY();¶
107 » » » » » cameraCenterX = cvProc.getCameraCenterX();¶
108 » » » » » cameraCenterY = cvProc.getCameraCenterY();¶
109 » » » » » camera.setCustomARProjectionMatrix(¶
110 » » » » » » » focalPointX,¶
111 » » » » » » » focalPointY,¶
112 » » » » » » » cameraCenterX,¶
113 » » » » » » » cameraCenterY,¶
114 » » » » » » » ProjectConstants.NEAR,¶
115 » » » » » » » ProjectConstants.FAR,¶
116 » » » » » » » ProjectConstants.W,¶
117 » » » » » » » ProjectConstants.H¶
118 » » » » » » » );¶
119 » » » » » camera.update(camera.projection);¶
120 ¶
```

# Ejemplos

- Código fuente:
  - <https://github.com/miky-kr5/EVI---AR-Demo>
  - <https://github.com/miky-kr5/EVI---AR-Prototype>



**git**

# Bibliografía

- Paul Milgram y Fumio Kishino "***A taxonomy of mixed reality visual displays.***", IEICE TRANSACTIONS on Information and Systems, 1994
- Daniel Lélis Baggio et al. "***Mastering OpenCV with practical computer vision projects.***", PAKT Publishing, 2012.

