

Demostración de Proyectos de Realidad Aumentada y Virtual

Dr. Marco Aurelio Nuño Maganda

Universidad Politécnica de Victoria
Laboratorio de Sistemas Inteligentes
mnunom@upv.edu.mx

Marzo de 2024

Contenido

- 1 Algoritmos, Programación y Teléfonos Inteligentes
- 2 Conceptos de Realidad Virtual y Aumentada
- 3 Prototipos de Realidad Virtual y Aumentada y Demostración a los Asistentes
- 4 Conclusiones

Outline

- 1 Algoritmos, Programación y Teléfonos Inteligentes
- 2 Conceptos de Realidad Virtual y Aumentada
- 3 Prototipos de Realidad Virtual y Aumentada y Demostración a los Asistentes
- 4 Conclusiones

Algoritmos y programación (1)

Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

Algoritmos y programación (1)

Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

- Las instrucciones deben ser claras (no ambiguas)

Algoritmos y programación (1)

Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

- Las instrucciones deben ser claras (no ambiguas)
- El orden debe estar definido, de tal forma que si no efectúan los pasos en el orden, no se logra el objetivo

Algoritmos y programación (1)

Algoritmo

Conjunto finito de instrucciones para resolver una tarea específica

- Las instrucciones deben ser claras (no ambiguas)
- El orden debe estar definido, de tal forma que si no efectúan los pasos en el orden, no se logra el objetivo




RECETA

TÍTULO: "Sandwich de jamón"

INGREDIENTES

- 1.2 Pan de caja
- 2.Mayonesa
- 3.Jamón
- 4.Queso
- 5.Lechuga
- 6.Tomate



PREPARACIÓN

- 1.Untar mantequilla en dos panes de caja
- 2.Colocar el jamón en un pan
- 3.Cortar en rebanada el queso, tomate
- 4.Agregar al pan una hoja de lechuga limpia y las rebanadas de queso y tomate
- 5.¡Tapa con el otro pan y ¡Listo! a comer

Algoritmos y programación (2)

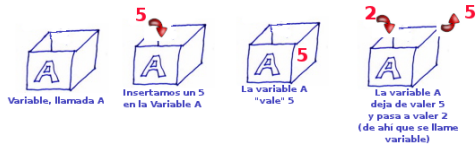
Variable

Un espacio reservado para almacenar un dato

Algoritmos y programación (2)

Variable

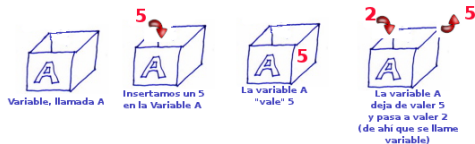
Un espacio reservado para almacenar un dato



Algoritmos y programación (2)

Variable

Un espacio reservado para almacenar un dato



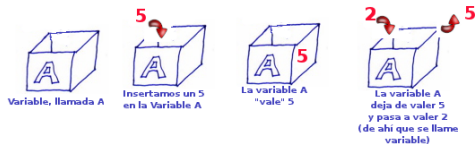
Arreglo

Una colección ordenada de variables del mismo tipo, que puede ser accesada por un índice

Algoritmos y programación (2)

Variable

Un espacio reservado para almacenar un dato



Arreglo

Una coleccion ordenada de variables del mismo tipo, que puede ser accesada por un indice

calificaciones[]							
6.8	8.3	10.0	9.5	4.5	7.7	9.4	10.0
0	1	2	3	4	5	6	7

Algoritmos y programación (3)

Obtener la mayor estatura y el lugar de la fila del niño(a) de mayor estatura



Algoritmos y programación (3)

Obtener la mayor estatura y el lugar de la fila del niño(a) de mayor estatura



- Medir o preguntar la estatura de los niños (En orden o en desorden?)

Algoritmos y programación (3)

Obtener la mayor estatura y el lugar de la fila del niño(a) de mayor estatura



- Medir o preguntar la estatura de los niños (En orden o en desorden?)
- Analizar las estaturas para obtener la mayor.

Algoritmos y programación (3)

Obtener la mayor estatura y el lugar de la fila del niño(a) de mayor estatura



- Medir o preguntar la estatura de los niños (En orden o en desorden?)
- Analizar las estaturas para obtener la mayor.
- Respalidar (escribir en alguna parte) la altura (o el lugar en la lista) del niño mas alto.

Algoritmos y programación (3)

Obtener la mayor estatura y el lugar de la fila del niño(a) de mayor estatura



- Medir o preguntar la estatura de los niños (En orden o en desorden?)
- Analizar las estaturas para obtener la mayor.
- Respalidar (escribir en alguna parte) la altura (o el lugar en la lista) del niño mas alto.

Algoritmos y programación (4)

Obtención de estaturas

```
1  numero_ninos = 8;
2  int Estaturas[numero_ninos]; //Estatura en Centimetros
3  indice = 0;
4  while (indice<numero_ninos)
5  {
6      EstaturaActual = preguntarEstatura(indice)
7      Estaturas[indice]=EstaturaActual
8      indice=indice+1
9  }
```

Algoritmos y programación (4)

Obtención de estaturas

```
1  numero_ninos = 8;  
2  int Estaturas[numero_ninos]; //Estatura en Centimetros  
3  indice = 0;  
4  while (indice<numero_ninos)  
5  {  
6      EstaturaActual = preguntarEstatura(indice)  
7      Estaturas[indice]=EstaturaActual  
8      indice=indice+1  
9  }
```

Estructura de Control Repetitiva (while)

Permite ejecutar un grupo de instrucciones por X cantidad de veces

Algoritmos y programación (4)

Obtención de estaturas

```

1  numero_ninos = 8;
2  int Estaturas[numero_ninos]; //Estatura en Centimetros
3  indice = 0;
4  while (indice<numero_ninos)
5  {
6      EstaturaActual = preguntarEstatura(indice)
7      Estaturas[indice]=EstaturaActual
8      indice=indice+1
9  }
```

Estructura de Control Repetitiva (while)

Permite ejecutar un grupo de instrucciones por X cantidad de veces

Análisis de las estaturas

```

1  numero_ninos = 8;
2  indice = 0;
3  while (indice<numero_ninos)
4  {
5      if (indice == 0){
6          mas_alto = indice
7          estatura_mas_alto=Estaturas[indice]
8      }
9      else
10     {
11         if (Estaturas[indice]>mas_alto) {
12             mas_alto = indice
13             estatura_mas_alto=Estaturas[indice]
14         }
15     }
16     indice=indice+1
17 }
```

Algoritmos y programación (4)

Obtención de estaturas

```

1  numero_ninos = 8;
2  int Estaturas[numero_ninos]; //Estatura en Centimetros
3  indice = 0;
4  while (indice<numero_ninos)
5  {
6      EstaturaActual = preguntarEstatura(indice)
7      Estaturas[indice]=EstaturaActual
8      indice=indice+1
9  }
```

Estructura de Control Repetitiva (while)

Permite ejecutar un grupo de instrucciones por X cantidad de veces

Análisis de las estaturas

```

1  numero_ninos = 8;
2  indice = 0;
3  while (indice<numero_ninos)
4  {
5      if (indice == 0){
6          mas_alto = indice
7          estatura_mas_alto=Estaturas[indice]
8      }
9      else
10     {
11         if (Estaturas[indice]>mas_alto) {
12             mas_alto = indice
13             estatura_mas_alto=Estaturas[indice]
14         }
15     }
16     indice=indice+1
17 }
```

Estructura de Control Selectiva (IF)

Permite ejecutar instrucciones basados en una condicion

Algoritmos y programación (4)

Obtención de estaturas

```

1  numero_ninos = 8;
2  int Estaturas[numero_ninos]; //Estatuta en Centimetros
3  indice = 0;
4  while (indice<numero_ninos)
5  {
6      EstaturaActual = preguntarEstatura(indice)
7      Estaturas[indice]=EstaturaActual
8      indice=indice+1
9  }
```

Estructura de Control Repetitiva (while)

Permite ejecutar un grupo de instrucciones por X cantidad de veces

Análisis de las estaturas

```

1  numero_ninos = 8;
2  indice = 0;
3  while (indice<numero_ninos)
4  {
5      if (indice == 0){
6          mas_alto = indice
7          estatura_mas_alto=Estaturas[indice]
8      }
9      else
10     {
11         if (Estaturas[indice]>mas_alto) {
12             mas_alto = indice
13             estatura_mas_alto=Estaturas[indice]
14         }
15     }
16     indice=indice+1
17 }
```

Estructura de Control Selectiva (IF)

Permite ejecutar instrucciones basados en una condicion

Algoritmos y programación (5)

Computadora

Es una máquina (electrónica) programable* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

Algoritmos y programación (5)

Computadora

Es una máquina (electrónica) programable* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

Programa de Computadora

Un conjunto de instrucciones escritas en un lenguaje de programación, que una computadora interpreta en una secuencia lógica para llevar a cabo una tarea en particular

Algoritmos y programación (5)

Computadora

Es una máquina (electrónica) programable* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

Programa de Computadora

Un conjunto de instrucciones escritas en un lenguaje de programación, que una computadora interpreta en una secuencia lógica para llevar a cabo una tarea en particular

Lenguaje de Programación

Un lenguaje que permite a un programador codificar instrucciones que serán ejecutadas por una computadora.

Algoritmos y programación (5)

Computadora

Es una máquina (electrónica) programable* que recibe y procesa datos para convertirlos en información útil. Contiene periféricos de entrada (para introducir datos) y salida (para mostrar resultados)

Programa de Computadora

Un conjunto de instrucciones escritas en un lenguaje de programación, que una computadora interpreta en una secuencia lógica para llevar a cabo una tarea en particular

Lenguaje de Programación

Un lenguaje que permite a un programador codificar instrucciones que serán ejecutadas por una computadora.

Algoritmos y programación (6)

Programa en Python para Leer estaturas

```

1  ArregloEstaturas=[0,0,0,0,0,0,0,0]
2  numero_ninos = 8;
3  indice=0
4  while (indice<numero_ninos):
5      Estatura=input("Dame la estatura del elemento "+str(indice)+" : ")
6      ArregloEstaturas[indice]=int(Estatura)
7      indice=indice+1
8
9  indice=0
10 while (indice<numero_ninos):
11     if (indice==0):
12         estatura_mas_alto=ArregloEstaturas[indice]
13         indice_mas_alto=indice
14     else:
15         if (ArregloEstaturas[indice]>estatura_mas_alto):
16             estatura_mas_alto=ArregloEstaturas[indice]
17             indice_mas_alto=indice
18     indice=indice+1
19 print ("Estura del mas alto: ",estatura_mas_alto)
20 print ("Posicion del mas alto: ",indice_mas_alto)
21

```

```

marco@marco-HP-Pavilion-dv7-Notebook-PC: ~
Archivo  EdRar  Ver  Buscar  Terminal  Ayuda
marco@marco-HP-Pavilion-dv7-Notebook-PC:~$ python3 Estaturas.py
Dame la estatura del elemento 0 : 120
Dame la estatura del elemento 1 : 130
Dame la estatura del elemento 2 : 150
Dame la estatura del elemento 3 : 145
Dame la estatura del elemento 4 : 178
Dame la estatura del elemento 5 : 119
Dame la estatura del elemento 6 : 110
Dame la estatura del elemento 7 : 147
Estura del mas alto: 178
Posicion del mas alto: 4
marco@marco-HP-Pavilion-dv7-Notebook-PC:~$

```

Algoritmos y programación (6)

Programa en Python para Leer estaturas

```

1  ArregloEstaturas=[0,0,0,0,0,0,0,0]
2  numero_ninos = 8;
3  indice=0
4  while (indice<numero_ninos):
5      Estatura=input("Dame la estatura del elemento "+str(indice)+" : ")
6      ArregloEstaturas[indice]=int(Estatura)
7      indice=indice+1
8
9  indice=0
10 while (indice<numero_ninos):
11     if (indice==0):
12         estatura_mas_alto=ArregloEstaturas[indice]
13         indice_mas_alto=indice
14     else:
15         if (ArregloEstaturas[indice]>estatura_mas_alto):
16             estatura_mas_alto=ArregloEstaturas[indice]
17             indice_mas_alto=indice
18     indice=indice+1
19 print ("Estura del mas alto: ",estatura_mas_alto)
20 print ("Posicion del mas alto: ",indice_mas_alto)
21

```

```

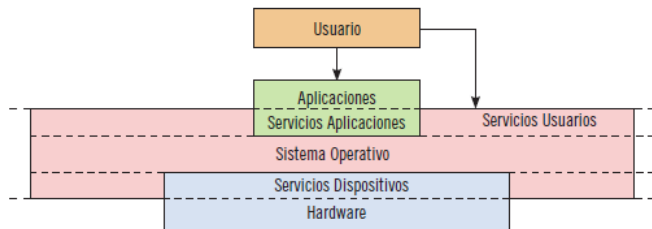
marco@marco-HP-Pavilion-dv7-Notebook-PC: ~
Archivo  EdRar  Ver  Buscar  Terminal  Ayuda
marco@marco-HP-Pavilion-dv7-Notebook-PC:~$ python3 Estaturas.py
Dame la estatura del elemento 0 : 120
Dame la estatura del elemento 1 : 130
Dame la estatura del elemento 2 : 150
Dame la estatura del elemento 3 : 145
Dame la estatura del elemento 4 : 178
Dame la estatura del elemento 5 : 119
Dame la estatura del elemento 6 : 110
Dame la estatura del elemento 7 : 147
Estura del mas alto: 178
Posicion del mas alto: 4
marco@marco-HP-Pavilion-dv7-Notebook-PC:~$

```

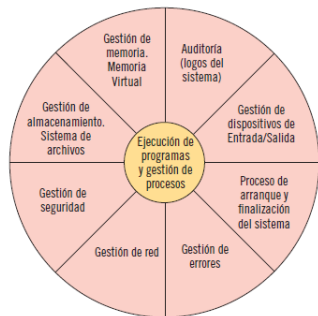
Sistema Operativo

Un Sistema Operativo (SO) es un programa (software) que al arrancar la computadora** se encarga de gestionar todos los recursos del sistema informático permitiendo así la comunicación entre el usuario y la computadora.

Estructuración de los servicios del sistema operativo



Bloques funcionales de un sistema operativo



Sistemas Operativos para PCs



Telefono Celular No-inteligente vs Telefono Celular Inteligente

Teléfono No-inteligente

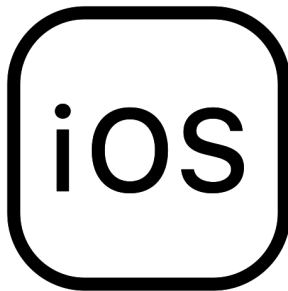
- Su funcionalidad principal era la comunicación (llamadas o mensajes) a través de la red celular (GSM)

Teléfono inteligente

- Interfaz de entrada: Pantalla Touch (a color, de alta definición)
- Conexión a Internet: WiFi, GSM (4G o 5G)
- Comunicación con otros dispositivos: Bluetooth, NFC
- Cámaras (Frontal y Posterior)

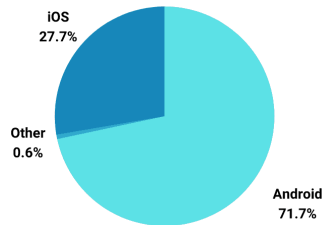


Sistemas Operativos para Telefonos Inteligentes



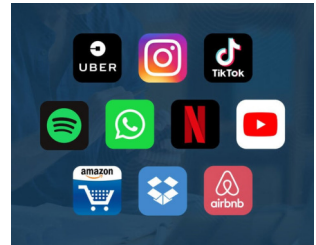
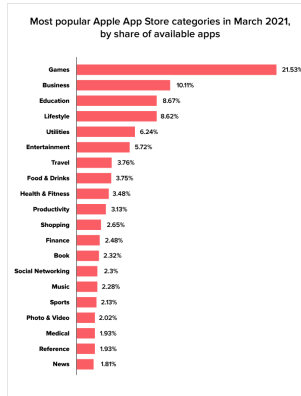
Android

- Android es un sistema operativo móvil basado en Linux
- Principalmente orientado a dispositivos de pantalla táctil (Smartphone, tablets, smartwatches, etc)
- Fue desarrollado por Android Inc (Adquirida por Google en 2005)
- Vinculado con un grupo de empresas (HTC, Sony, Motorola, Samsung, LG, Lenovo, entre otras) para la creación de un SO común para sus dispositivos
- A la fecha (Q1 2023), los teléfonos con SO Android concentran mas del 70% del mercado global.



Aplicaciones Móviles

- Ejecutadas en el teléfono
- La entrada de datos es mediante un teclado “virtual”
- El apuntador del raton es la pantalla
- Incluyen una interfaz de usuario gráfica (GUI)
- Es posible descargar miles de éstas en nuestros dispositivos



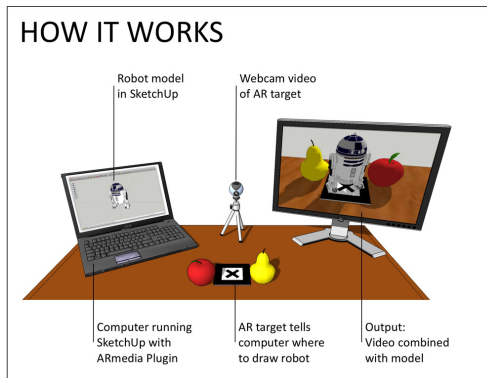
<https://www.netsolutions.com/insights/top-10-most-popular-apps-2018/>

Outline

- 1 Algoritmos, Programación y Teléfonos Inteligentes
- 2 Conceptos de Realidad Virtual y Aumentada
- 3 Prototipos de Realidad Virtual y Aumentada y Demostración a los Asistentes
- 4 Conclusiones

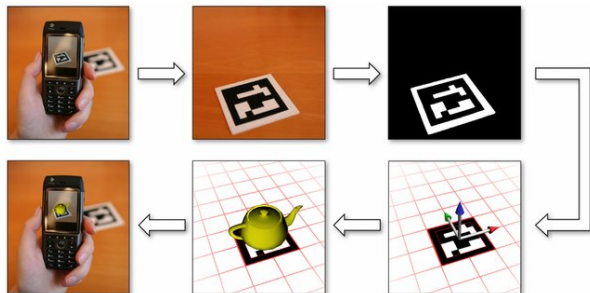
Realidad Aumentada (AR)

- La AR es una experiencia que traslapa elementos digitales (modelados por computadora) con el mundo físico del usuario (adquirido mediante una cámara).
- Los elementos digitales se combinan con las vistas del mundo real.



Pasos en la detección de marcadores de AR

- 1 Umbralización.
- 2 Detección del Marcador.
- 3 Estimación de Pose y Posición.
- 4 Empalme del modelo 3D.

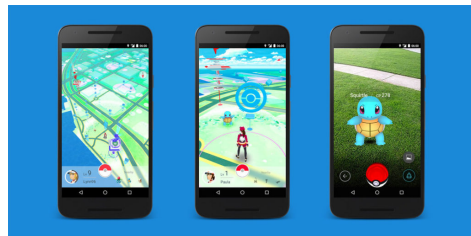


Tipos de Aplicación de AR

- Basadas en Localización. Están basada en sensores GPS para determinar la ubicación del dispositivo para crear objetos AR
- Basadas en Visión – Utilizan una cámara, aunque también es posible incorporar sensores (compass, acelerómetros, giroscopios, etc).
 - Requieren Marcadores (Marker) – Localizan un patrón o marcador QR y renderizan un objeto 3D basado en su localización en el espacio real.
 - No requieren marcadores (Markerless)– Se emplean esquinas y puntos característicos del espacio real

Aplicaciones de RA

- Aplicaciones principales: Arquitectura, Cosméticos, Contenido social, Marketing, Juegos, etc
- Houzz
- Pokemon Go

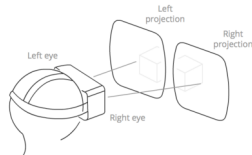
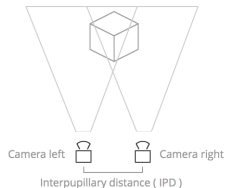


Realidad Virtual - Antecedente histórico

- Un estereoscopio proporciona imágenes separadas para cada ojo mediante lentes individuales, donde cada imagen tiene una variante en el ángulo de captura y un desplazamiento horizontal.
- El cerebro de una persona con una percepción binocular normal de la profundidad al utilizar el estereoscopio “mezcla” ambas imágenes para crear una “ventana estereoscópica”

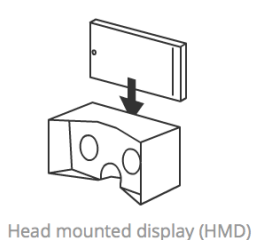


How to create stereoscopic 3D images



Realidad Virtual (VR)

- La Realidad virtual (RV) emplea modelos y simulaciones por computadora que permite a una persona interactuar con un entorno visual artificial tridimensional (3D)
- En un formato típico de RV, un usuario lleva un casco con una pantalla estereoscópica para ver imágenes animadas de un entorno simulado
- El dispositivo más económico para aplicaciones de RV es un teléfono inteligente



Costos de Dispositivos HeadSets para VR

- Meta Quest 3 (499 USD)
- Sony PlayStation VR2 (599 USD)
- Meta Quest Pro (900 USD)
- Valve Index VR Kit (1350 USD)
- HTC Vive Pro 2 (1400 USD)



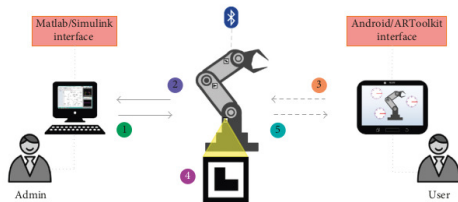
Outline

- 1 Algoritmos, Programación y Teléfonos Inteligentes
- 2 Conceptos de Realidad Virtual y Aumentada
- 3 Prototipos de Realidad Virtual y Aumentada y Demostración a los Asistentes
- 4 Conclusiones

“An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality” (1)

Componentes:

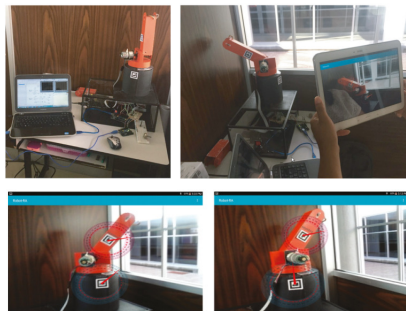
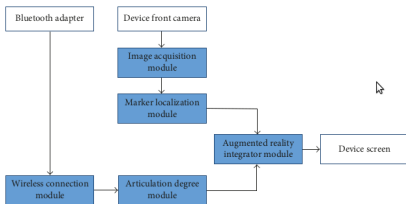
- Un sistema (Arduino) que genera los movimientos del brazo robot incluye un transmisor Bluetooth
- Una aplicación móvil que visualizar un transportador virtual encima de una articulación robótica con el ángulo en tiempo real



“An Education Application for Teaching Robot Arm Manipulator Concepts Using Augmented Reality” (2)

Funcionamiento de la aplicación:

- Se emplea un marcador ARUCO para determinar de que articulación se trata
- Mediante comandos Bluetooth se obtiene el ángulo



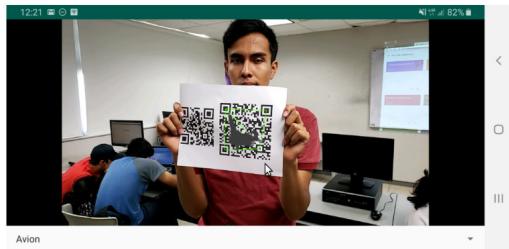
Visión por Computadora para Realidad Aumentada (1)

- Detección de Codigos QR
- Decodificación del texto codificado en el codigo QR
- Sobreposición del modelo 3D dependiendo de la posición del código QR



* Jesús Alfredo Cárdenas-Castillo, Eluis Carlo Ramos-Lucio Ana Karen Molina-Pastrana and, Linda Margarita Rodríguez-Terán, and César Jovany Vázquez-Luna. *Visión por Computadora para Realidad Aumentada*. Proyecto Final de la Asignatura Cómputo en Dispositivos Móviles. Universidad Politécnica de Victoria, 2019

Visión por Computadora para Realidad Aumentada (2)



Modelo 3D del Jaguar de la UPV sobre un código QR (1)

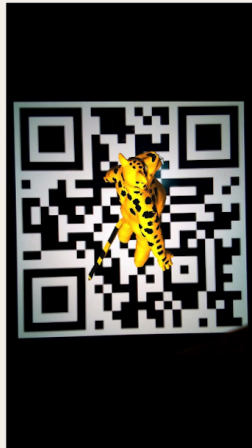
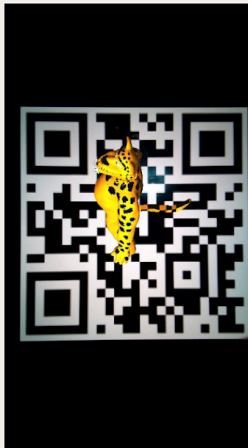
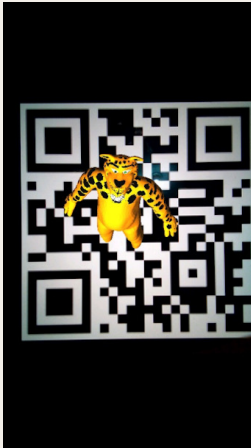
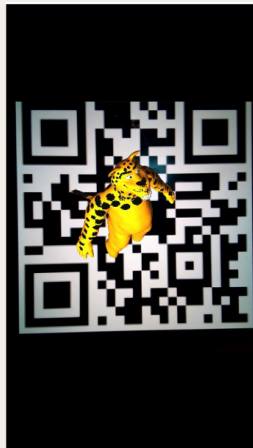
- Detección de Códigos QR
- Modelado de la mascota en 3D mediante el Software Blender
- Sobreposición del modelo 3D dependiendo de la posición del código QR



* Andrés García-González, Cristian Aléxis Lazo-García, and Damaris Mendoza-Vázquez. *Modelo 3D del Jaguar de la UPV sobre un código QR.*

Informe técnico proyecto de asignatura "Graficación por Computadora Avanzada". Universidad Politécnica de Victoria, 2020

Modelo 3D del Jaguar de la UPV sobre un código QR (2)



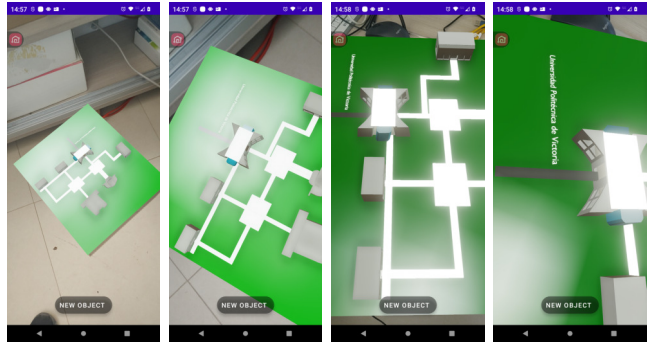
*Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria** (1)

- Se propone una aplicación de RA que despliegue un MAPA de la UPV
- Se generó el modelo utilizando Blender y se exportó a un formato GLB
- La aplicación hace uso de la cámara para desplegar el modelo en una superficie con textura

* Eymi Lizeth Andrade-Martinez, Edson Jair Hernández-Garza, Oscar Esteban Infante-González, José Luis Leal Requena, Pablo Nahúm Ochoa-Morales, Jesús Guadalupe Rangel-Turrubiates, José Manuel Rodríguez-García, and Carlos Eduardo Sánchez-Charles. *Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria*. Informe técnico proyecto de asignatura "Cómputo en Dispositivos Móviles". Universidad Politécnica de Victoria, 2022.

Desarrollo de una app móvil de realidad aumentada de la Universidad Politécnica de Victoria (2)

- Se propone una aplicación de RA que despliegue un MAPA de la UPV



Recorrido UPV Virtual en teléfonos inteligentes

Demo incremental, que emplea OpenGL ES 2.0 (compatible con el 100% de los smartphones).

- Versión 1: Solo mundo virtual (no inmersivo). El usuario se movía con presionando teclas de la interfaz de usuario
- Versión 2: Mundo virtual inmersivo (integrado a unos lentes). El usuario movía la vista mediante los datos obtenidos por el sensor giroscopio del teléfono inteligente y avanzaba usando un manos libres alámbrico.
- Versión 3: Controlado por voz. El usuario se movía dentro del entorno mediante comandos de voz.
- Versión 3.5: Controlado mediante control de videojuegos (Bluetooth o USB)

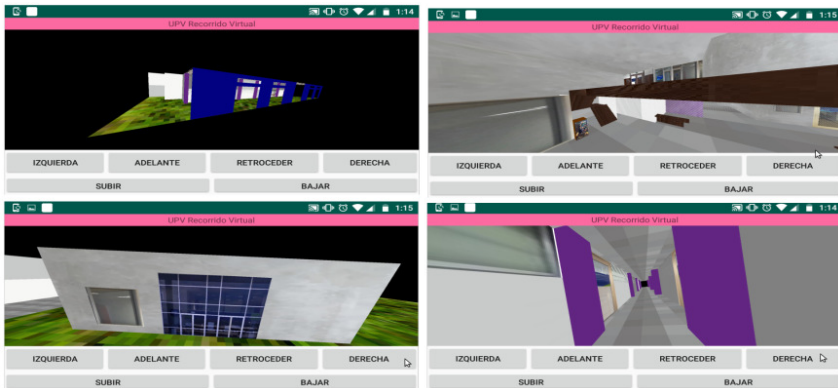
Recorrido UPV en Android (1)

- La navegación es mediante botones (adelante, atrás, izquierda, derecha, subir, bajar)
- Una potencial mejora es mediante eventos de toque en pantalla



Recorrido UPV en Android (2)

- Vistas de los diferentes edificios



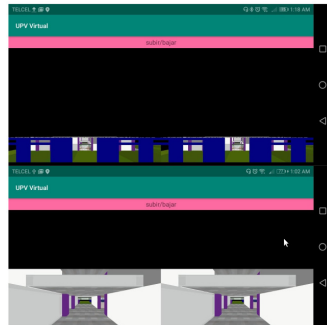
Aplicación UPV-Virtual (2)

- Se extendió el demo 1.0 para generar una vista dual, requerida para su uso en conjunto con un armazón de VR.
- La vista cambia en base a lo obtenido por el giroscopio, y el movimiento se controla mediante el botón del manos libres.



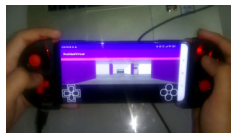
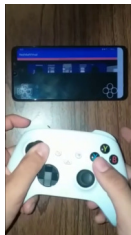
Recorrido UPV Virtual en Android con Controles de Voz. (1)

- Se eliminó el uso del botón del manos libres para incluir comandos de voz
- La aplicación respondía a los comandos de voz, de tal forma que el usuario no debería mover nada



Incorporación de Controles de Consola de Video Juego al Recorrido Virtual de la UPV (1)

- Se incorporaron varios controles de consolas de videojuego para la navegación.



* Mario Alberto Delarbre-Quintanilla, Gerardo Loperena-Bustillos, Alejandro Lumbreras-Olvera, Jorge Eduardo Monita-Rodríguez, Christian Alejandro Saldaña-Calderon, Juan Ramiro Soto-Gómez, Claudia Corina Vásquez-Pérez, Hector Yahude Vallejo-Cepeda, Andrea Medrano-Salvidar, Julio Alberto Zuniga-Gallegos, and Darianna Carreon-Gomez. *Incorporación de Controles de Consola de Video Juego al Recorrido Virtual de la UPV*. Informe técnico proyecto de asignatura "Cómputo en Dispositivos Móviles". Universidad Politécnica de Victoria, 2021.

Filtro de Cabeza de Jaguar UPV (1)

- La mascota institucional de la UPV es el Jaguar
- Se diseñó un filtro que detecta el rostro de una persona y sobrepone la imagen de un Jaguar



Mapa de Realidad Aumentada UPV (1)

- Se diseñó una aplicación de Realidad Aumentada que muestra un mapa de la UPV
- Se modeló el mapa en Blender y se integró a la aplicación



* Jorge Luis Charles-Torres, Paullette Monserrat Esparza-Gonzales, Juan Emmanuel Laurian-Mendoza, and Francisco Gael Sustaita-Reyna Humberto Erubiel Ortega-Lujano and. *Mapa de Realidad Aumentada UPV*. Proyecto Final de la Asignatura Programación Móvil. Universidad Politécnica de Victoria, 2023

Outline

- 1 Algoritmos, Programación y Teléfonos Inteligentes
- 2 Conceptos de Realidad Virtual y Aumentada
- 3 Prototipos de Realidad Virtual y Aumentada y Demostración a los Asistentes
- 4 Conclusiones**

Conclusión

- Se presentaron los fundamentos de algoritmos, computadoras y lenguajes de programación
- Se vieron conceptos relacionados con la RA y la RV
- Se presentaron proyectos de RA y RV, principalmente desarrollados por estudiantes de la UPV (Ingeniería en Tecnologías de la Información y de Maestría en Ingeniería)
- Los asistentes pudieron interactuar con algunos de los demos presentados

Agradecimiento

Se agradece al Consejo Tamaulipeco de Ciencia y Tecnologia (COTACYT) por la invitación a este evento
Comentarios o Dudas: mnunom@upv.edu.mx