

KINEMATICSAPP: APLICACIÓN EDUCATIVA BASADA EN REALIDAD AUMENTADA
PARA ENSEÑAR CINEMÁTICA

DUVÁN IGNACIO GIRALDO MONTOYA

UNIVERSIDAD CATÓLICA DE ORIENTE

INGENIERÍA DE SISTEMAS

RIONEGRO

2020

KINEMATICSAPP: APLICACIÓN EDUCATIVA BASADA EN REALIDAD AUMENTADA
PARA ENSEÑAR CINEMÁTICA

DUVÁN IGNACIO GIRALDO MONTOYA

Trabajo de grado para optar al título de Ingeniero de Sistemas

Asesor técnico

Lukas Mauricio Restrepo Suárez

Asesor metodológico

Juan Fernando Garzón Álvarez

UNIVERSIDAD CATÓLICA DE ORIENTE

INGENIERÍA DE SISTEMAS

RIONEGRO

2020

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Rionegro, 09 de julio del 2020

CONTENIDO

	Pág.
INTRODUCCIÓN	7
ANTECEDENTES	8
1. PLANTEAMIENTO DEL PROBLEMA	11
2. PREGUNTA DE INVESTIGACIÓN	13
3. JUSTIFICACIÓN	14
4. OBJETIVOS	15
4.1. Objetivo general	15
4.2. Objetivos específicos	15
5. MARCO TEÓRICO	16
6. DISEÑO METODOLÓGICO	22
6.1. Análisis	22
6.2. Diseño	26
6.3. Implementación	27
7. RESULTADOS	35
8. TRABAJO FUTURO	40
9. CONCLUSIÓN	42
REFERENCIAS BIBLIOGRÁFICAS	43
ANEXOS	45

LISTA DE FIGURAS

	Pág.
Figura 1. Diagrama de abstracción de la información de un marcador para Vuforia	17
Figura 2. <i>Tracking</i> o captura de la información	18
Figura 3. Distribución de paquetes a nivel de código	22
Figura 4. Modelos cañón y cursor para el tiro parabólico	23
Figura 5. Método que controla el tiro parabólico dado un ángulo inicial	24
Figura 6. Método que controla el tiro parabólico dada una velocidad inicial	24
Figura 7. Método de disparo en el módulo de tiro parabólico	25
Figura 8. Clase que controla la simulación del arco en el módulo de tiro parabólico	26
Figura 9. Clase encargada del control del cursor	28
Figura 10. Clase encargada del enfoque de la cámara	29
Figura 11. Clase para el control de flash light de la cámara	29
Figura 12. Clase encargada del diálogo informativo que tiene cada módulo	30
Figura 13. Clase para controlar la interfaz informativa	31
Figura 14. Clase para controlar el cambio de las imágenes	32
Figura 15. Clase utilitaria para todas las interfaces	33
Figura 16. Clase para el control del sonido	34
Figura 17. Logo de KinematicsApp	35
Figura 18. Interfaz de bienvenida de KinematicsApp	36
Figura 19. Interfaz principal de KinematicsApp	37
Figura 20. Interfaz informativa de KinematicsApp	37
Figura 21. Opción de movimiento rectilíneo uniforme de KinematicsApp	38
Figura 22. Opción de movimiento rectilíneo acelerado de KinematicsApp	39

Figura 23. Movimiento parabólico en la opción de ángulo inicial	39
Figura 24. Movimiento parabólico en la opción de velocidad inicial	40
Figura 25. Interfaces y módulos versión 1 KinematicsApp	45
Figura 26. Interfaces y módulos versión 2 KinematicsApp	46
Figura 27. Representación de <i>ground plane</i>	46

INTRODUCCIÓN

Las herramientas tecnológicas que apoyan pedagógicamente a los procesos formativos de las personas, tanto estudiantes como profesores, intervienen de forma directa gracias al uso de la tecnología como instrumento primordial y de uso cotidiano en la actualidad.

La Realidad Aumentada (RA) es una tecnología que permite crear elementos virtuales basados en componentes reales y cotidianos como imágenes u objetos tridimensionales, lo cual extiende las posibilidades de construir desarrollos que unan el mundo real con el mundo virtual y aplicativos más amigables con las personas y sus necesidades educativas o de ocio.

Con el fin de apoyar los procesos de enseñanza aprendizaje en la Universidad Católica de Oriente, se desarrolla una aplicación móvil basada en RA. Esta aplicación se denomina *KinematicsApp*, está destinada a aprender los conceptos de movimiento rectilíneo uniforme, movimiento uniforme acelerado y movimiento parabólico en la rama de la cinemática, llamados en el presente trabajo como “temas de investigación”. Trata de acercar a las personas a una interacción entre lo real y lo virtual, donde los estudiantes muestren un mayor interés en la experimentación, no solo desde un laboratorio sino en cualquier parte del mundo.

Para el desarrollo de la aplicación se utilizó el motor de videojuegos Unity 3D, la librería Vuforia para el control del dispositivo y los procesos que intervienen con RA, y herramientas externas como Photoshop, Visual Studio y algunos scripts y modelos del proyecto *ProjectileShooting* publicado en la plataforma GitHub por el usuario IronWarrior. KinematicsApp es un proyecto de código abierto que ha sido publicado en GitHub en la dirección github.com/duvangiraldom/kinematics-app.

ANTECEDENTES

En la actualidad, el avance tecnológico crece de manera exorbitante. Según la revista Network World, se calcula que para el 2021 la cifra de dispositivos conectados a Internet tocará los 36.130 millones (Network World, 2019), no es necesario trasladarse de un lugar a otro para comunicarse con las personas, la información es accesible en cualquier momento o lugar del mundo y las técnicas de enseñanza y aprendizaje cada vez necesitan más de la tecnología. “El uso de recursos educativos digitales se convierte en la metodología del futuro para la transmisión de conocimiento; porque aprovecha los recursos digitales para lograr que el alumno aumente su interés en el aprendizaje” (Cuendet, Bonnard, Do-Lenh, & Dillenbourg, 2013).

Son indispensables diferentes tecnologías que sirven como base para la elaboración y facilidad de creación de los recursos digitales; entre ellas se encuentra Unity, un motor de videojuegos multiplataforma creado por Unity Technologies, que está disponible para múltiples sistemas operativos y con una versión gratuita, y Vuforia, un kit de desarrollo de software que permite la creación de aplicaciones en RA, también disponible para distintos sistemas operativos y en una versión gratuita.

Entre los desarrollos basados en RA que sirven como recurso digital educativo se encuentran algunos ejemplos tales como:

- El Instituto de Tecnología de Software y Sistemas Interactivos de la Universidad de Tecnología de Viena en Austria desarrolla una herramienta educativa llamada *Construct3D*. El desarrollo se basa en interactuar y ver cosas en tercera dimensión. “Puede mejorar la comprensión de la geometría tridimensional por parte de un estudiante” (Kaufmann, 2000). Aunque los resultados del estudio muestran que los estudiantes no utilizan la herramienta con regularidad, se

plantean futuras evaluaciones con profesores, estudiantes y colegas del instituto para inspeccionar las dificultades del desarrollo y proponer mejoras en el aplicativo.

- *Elect3D* es una aplicación para lectura de planos eléctricos, donde los estudiantes tienen una experiencia más realista en el aprendizaje de símbolos propios de la ingeniería eléctrica. Gutiérrez, Fabiani, Benesova y Meneses (2015) afirman:

De manera colaborativa y autónoma, este trabajo combina cada proceso de aprendizaje del curso de máquinas eléctricas en el grado de ingeniería eléctrica. Permite el estudio interactivo y autónomo, así como la realización colaborativa de prácticas de laboratorio con otros estudiantes y sin la ayuda de un profesor. (p.1)

- *ElectARmanual*, se usa en el entrenamiento AR para instalaciones y prácticas con máquinas eléctricas. *Elect3D* y *ElectARmanual* apoyan el aprendizaje colaborativo y autónomo en la educación superior según los resultados del nivel de usabilidad de las aplicaciones.

- Se desarrolla una herramienta basada en AR para verificar la autenticidad de la nueva familia de billetes colombianos. “La aplicación que apoyaría las estrategias de difusión de la nueva familia de billetes, debía ser desarrollada con la característica que la diferenciaría, a través de la AR, considerando que la empresa ya había adoptado este tipo de desarrollo innovador” (Cárdenas, 2017).

- Se analiza una experiencia de innovación universitaria de RA con 117 estudiantes que cursan asignaturas de Tecnología de la Información y la Comunicación en la Universidad Pablo de Olavide de Sevilla (España). Los estudiantes pueden desarrollar los objetivos y competencias de las asignaturas con el uso de aplicaciones didácticas desarrolladas con RA orientadas al contexto educativo que acompañan la parte teórica de los cursos al utilizar marcadores impresos en los

libros guías y procesados con las aplicaciones generan un entorno más amigable y didáctico para los estudiantes. El estudio muestra que, con el uso de RA en el ámbito formativo, los estudiantes evidencian un mayor interés por el aprendizaje y ellos mismos consideran que determinadas aplicaciones con RA estimulan su atención, concentración, memoria inmediata y el razonamiento según la encuesta realizada posteriormente a la implementación del material en las aulas de clase.

1. PLANTEAMIENTO DEL PROBLEMA

Existe una deserción de estudiantes universitarios en el proceso formativo por la falta de técnicas que favorecen el aprendizaje de una manera didáctica, el uso de metodologías de enseñanza como tiza o marcador y tablero, clases magistrales, evaluaciones con una hoja de papel y lápiz a lo largo de todo el semestre sin tener cambios o alteraciones que no estimulan el aprendizaje moderno, poco didácticas, donde los alumnos pierden interés, atención y concentración y principalmente por la escasez de uso de herramientas tecnológicas que faciliten, apoyen, sean transversales en el desarrollo educativo y asequibles para todos de manera gratuita.

El estudio o abordaje de los procesos de aprendizaje en los temas de investigación ocurre desde diferentes perspectivas metodológicas dentro de un laboratorio, donde las medidas son inexactas, debe haber mucho margen de error, las personas deben trasladarse de su lugar de estudio hasta el laboratorio, los materiales que se utilizan para las diferentes practicas no son suficientes lo que limita el proceso de aprendizaje y comúnmente no se utilizan materiales tecnológicos que ayuden a tener interés, gusto y motivación por aprender a causa de los altos costos de la actualización permanente del material de experimentación disponible.

La RA permite hacer la experimentación de diferentes fenómenos físicos uniendo elementos del mundo real y componentes virtuales para comportarse de manera complementaria; así, se incita a la investigación y avance educacional de las personas. Al utilizar tecnología de última generación tanto en el aspecto metodológico, pedagógico y didáctico se tiene un avance científico en múltiples ramas de la ciencia y aportes para hacer del mundo un lugar donde lo real se une con lo virtual.

Una solución tecnológica basada en RA capaz de llevar la enseñanza-aprendizaje de manera didáctica es indispensable porque apoya directamente el proceso educativo del estudiante, facilita los materiales necesarios para los diferentes laboratorios, se pueden generar pruebas donde lo

teórico se evidencia de manera práctica y permite realizar ensayos en cualquier lugar o momento determinados de los temas de investigación, solo con la ayuda de una imagen o marcador que puede estar en un dispositivo o impresa y un smartphone con las características de hardware requeridas para superponer los elementos virtuales al detectar esta imagen.

2. PREGUNTA DE INVESTIGACIÓN

¿Cómo el uso de una aplicación educativa basada en realidad aumentada facilita el proceso de enseñanza-aprendizaje de la cinemática?

3. JUSTIFICACIÓN

En el proceso educativo de los estudiantes de ingeniería son necesarias herramientas tecnológicas didácticas que apoyen el proceso formativo para generar mejores curvas de aprendizaje y desarrollar habilidades de comprensión desde un salón de estudios, casa, laboratorio o cualquier lugar del mundo.

El estudiante contemporáneo necesita darle uso a la tecnología para los procesos de aprendizaje. El internet juega un papel importante, según la revista Forbes, “El aprendizaje en línea hace que la educación esté disponible para aquellos incluso en áreas remotas, así como también facilita el intercambio de planes de estudio a través de las fronteras.” (Marr, 2020). Es necesario para el alojamiento y distribución de contenido digital para el aprendizaje.

Muchas veces los costos de operación, construcción y mantenimiento de los laboratorios, materiales y demás recursos necesarios para llevar a cabo las diferentes prácticas, superan lo que lleva construir tecnología capaz de facilitar los procesos de enseñanza-aprendizaje de una manera más didáctica y cómoda tanto para el estudiante como para el docente. Por lo tanto, se considera una herramienta tecnológica capaz de visualizar de forma práctica los temas de investigación desde cualquier momento o lugar determinados.

4. OBJETIVOS

4.1. Objetivo general

Construir una aplicación basada en realidad aumentada para acompañar el proceso de enseñanza-aprendizaje de los conceptos básicos de la cinemática.

4.2. Objetivos específicos

Planear los requisitos, necesidades y expectativas que dan la trazabilidad a los requerimientos funcionales y no funcionales de la aplicación.

Diseñar el esquema conceptual, de apariencia y funcionalidad de la aplicación.

Implementar el diseño conceptual, de apariencia y funcionalidad de la aplicación.

5. MARCO TEÓRICO

En el presente apartado se dan a conocer los diferentes principios, conceptos, técnicas y herramientas que se deben tener en cuenta para una mejor comprensión sobre el desarrollo del recurso didáctico mediado por tecnología. Se aborda sobre la metodología de ingeniería del software educativo, el aprendizaje a través del juego y los conceptos básicos sobre cinemática, movimiento rectilíneo uniforme, movimiento uniformemente acelerado y movimiento parabólico.

Realidad Aumentada (RA): Es la asociación del mundo real y lo virtual, con ella se pueden mezclar objetos del entorno con elementos creados por las personas mismas. Es posible definir el límite hasta el cual llega cada uno de ambos mundos. La RA permite al usuario interactuar con el mundo real y con elementos o aumentos digitales (Unity, 2019).

Un ejemplo muy popular de aplicaciones que implementan RA es *Pokemon Go*, en este, con la señal GPS, utiliza el mapa real mientras superpone el contenido del videojuego. El hardware RA puede estar en dispositivos como smartphones, televisores o cualquier otro que presente sensores como giroscopio, de proximidad y capaz de correr algún tipo de software que los controle.

Marcador o *target*: Es la imagen que se captura con la aplicación RA para tener un punto de referencia en la superposición de los objetos virtuales y la abstracción de la información del mundo real. En RA, los marcadores son imágenes u objetos que actúan como activadores de información la cual es escaneada por el aplicativo desarrollado (Unity, 2019).

Existen marcadores bidimensionales que son imágenes, pero también se pueden tener marcadores en tercera dimensión. Para este último, es necesario un software especializado en escanear objetos 3D y un hardware con mayores especificaciones, capaz de realizar el *tracking* o detección de este como se muestra en la Figura 2.

Vuforia: Es un kit de desarrollo multiplataforma que permite desarrollar aplicaciones de RA, detectar marcadores del mundo real, superponer objetos virtuales, detección de oclusión localizada mediante “botones virtuales”, detección de superficies planas (*Ground Plane*) y creación de *Smart terrain* (véase el anexo C). En la documentación oficial de Unity (2019), se afirma que:

Las aplicaciones RA desarrolladas con Vuforia son compatibles con una amplia gama de dispositivos móviles, incluidos los teléfonos y tabletas iPhone, iPad y Android que ejecutan Android OS versión 2.2 o superior y un procesador ARMv6 o 7 con capacidad de procesamiento de unidades de punto flotante (FPU).

Vuforia permite registrar ambos tipos de objetos, tanto planos como tridimensionales con *Vuforia Scanner*, para ser utilizados por la aplicación. En su página oficial, Vuforia tiene un listado con los dispositivos que permiten su implementación (véase el anexo A).

El potente kit de desarrollo es capaz de leer la información de los marcadores por medio de la cámara del dispositivo, hace que la aplicación cree o renderice objetos virtuales sin tener que preocuparse por el sistema operativo bajo el cual se está corriendo. La Figura 1 muestra un diagrama de abstracción de la información según Vuforia.



Figura 1. Diagrama de abstracción de la información de un marcador para Vuforia (Oliver,

Unity: “Unity es un motor de desarrollo para la creación de juegos y contenidos 3D interactivos, con las características que es completamente integrado y que ofrece innumerables funcionalidades para facilitar el desarrollo de videojuegos” (Luttecke, 2014).

La plataforma para el desarrollo de videojuegos Unity posee alta facilidad de uso en la configuración de entornos 3D, desarrollo de aplicaciones móviles y la integración de *plugins* o complementos con librerías externas como Vuforia. El kit de desarrollo se integra al motor de videojuegos con el fin de controlar los periféricos del dispositivo móvil y la superposición de los objetos virtuales.

Tracking: El seguimiento es un concepto propio de RA, también llamado *tracking device* y es la manera en que el dispositivo captura la información de los targets, hace seguimiento a la disponibilidad de la imagen e incluso cuando el objetivo ya no está en el campo de visión de la cámara como muestra la Figura 2.

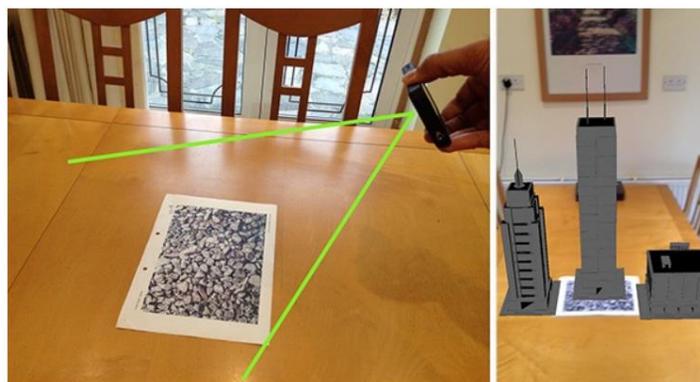


Figura 2. *Tracking* o captura de la información (Vuforia, 2019).

Visual Studio: Es una plataforma que facilita la edición, depuración y creación de código que luego se publica y convierte en lo que se conoce como una aplicación. Según (Microsoft, 2019):

Un entorno de desarrollo integrado (IDE) es un programa rico en funciones que se puede utilizar para muchos aspectos del desarrollo de software. Además del editor y depurador estándar que proporcionan la mayoría de los IDE, Visual Studio incluye compiladores, herramientas de depuración de código, diseñadores gráficos y muchas más funciones para facilitar el proceso de desarrollo de software.

Adobe Photoshop: Es una herramienta de creación y edición de imágenes de manera profesional. Se utiliza para establecer el ícono del aplicativo móvil.

Se aprovechan las versiones gratuitas y licenciadas tanto de Unity, Vuforia, Visual Studio y Photoshop para el desarrollo del aplicativo móvil.

A nivel metodológico, el presente escrito se basa en la enseñanza-aprendizaje a través de juegos. Lawrence Robinson et al. (2019) afirma:

Si bien el juego es crucial para el desarrollo de un niño, también es beneficioso para personas de todas las edades. El juego puede agregar alegría a la vida, aliviar el estrés, acelerar el aprendizaje y conectarte con los demás y con el mundo que te rodea.

El juego no solo trae beneficios a la salud, lo más importante en sí es que acelera el aprendizaje, ya que es algo espontáneo y liberador, es fundamental para la supervivencia y la paz en la sociedad, no es algo actual y desde niños se involucra en temas de enseñanza al igual que la tecnología.

“La herramienta utilizada es solo un medio para despertar el interés, mantener la motivación y la participación en el proceso de enseñanza-aprendizaje” (Fernández Aedo, Server García, & Cepero Fadruga, 2001).

Las herramientas tecnológicas con las que se puede dar un proceso de enseñanza-aprendizaje sirven como apoyo para considerar los beneficios y conservar el gusto por la adquisición del conocimiento. Con la participación de juegos en las herramientas tecnológicas se producen aplicaciones educativas que estimulan el aprendizaje autónomo a través del juego, se fomenta la motivación y se promueve el afán de superación, lo cual genera constancia y entendimiento de forma divertida.

Los temas que se estudian en la herramienta tecnológica son el movimiento rectilíneo uniforme, movimiento uniformemente acelerado y movimiento parabólico que pertenecen a la rama de la cinemática.

Cinemática: Es la rama de la física mecánica que estudia el movimiento de los cuerpos.

La cinemática tiene como objetivo proporcionar una descripción de la posición espacial de los cuerpos o sistemas de partículas materiales, la velocidad a la que se mueven las partículas (velocidad) y la rapidez a la que cambia su velocidad y aceleración (The Editors of Encyclopaedia Britannica, 2017).

Movimiento Rectilíneo Uniforme: Es un tipo de movimiento dentro de la cinemática en el cual un cuerpo se mueve con velocidad constante a medida que pasa el tiempo.

Cuando requerimos solo un eje de coordenadas junto con el tiempo para describir el movimiento de una partícula, se dice que está en movimiento lineal o rectilíneo. Algunos ejemplos de movimiento lineal son un desfile de soldados, un tren que se mueve en línea recta y muchos más (Byjus, 2019).

Movimiento Uniformemente Acelerado: Como su nombre lo indica, es el movimiento donde existe variación de la distancia, la velocidad y el tiempo, pero no de la aceleración.

Movimiento Parabólico: Es la reunión del movimiento rectilíneo uniforme y el movimiento uniforme acelerado; a nivel de ecuaciones y fórmulas se puede tratar el movimiento como dos y cada uno por separado.

El movimiento de proyectil es uno de los ejemplos más populares de movimiento en un plano con aceleración uniforme. En caso de movimiento de proyectil, la única aceleración que actúa sobre la partícula es la aceleración debida a la gravedad (g). En dirección horizontal, o la velocidad en la dirección x será constante, ya que no hay aceleración. Por lo tanto, podemos aplicar las ecuaciones de movimiento por separado en dos direcciones para obtener resultados (Byjus, 2019).

6. DISEÑO METODOLÓGICO

Se realiza una investigación de tipo cuantitativo, estudio exploratorio e investigación aplicada dividida en tres fases principales que iteraran a lo largo del desarrollo: Análisis, diseño e implementación.

6.1. Análisis

Se identifican las tecnologías requeridas para el desarrollo como Unity, Vuforia, Visual Studio, la distribución de paquetes y algunos Scripts y modelos del proyecto *ProjectileShooting* publicado en la plataforma GitHub por el usuario IronWarrior para el movimiento y diseño del cañón que se utiliza en el módulo de tiro parabólico.

El aplicativo sigue la distribución de paquetes a nivel de código según la Figura 3.

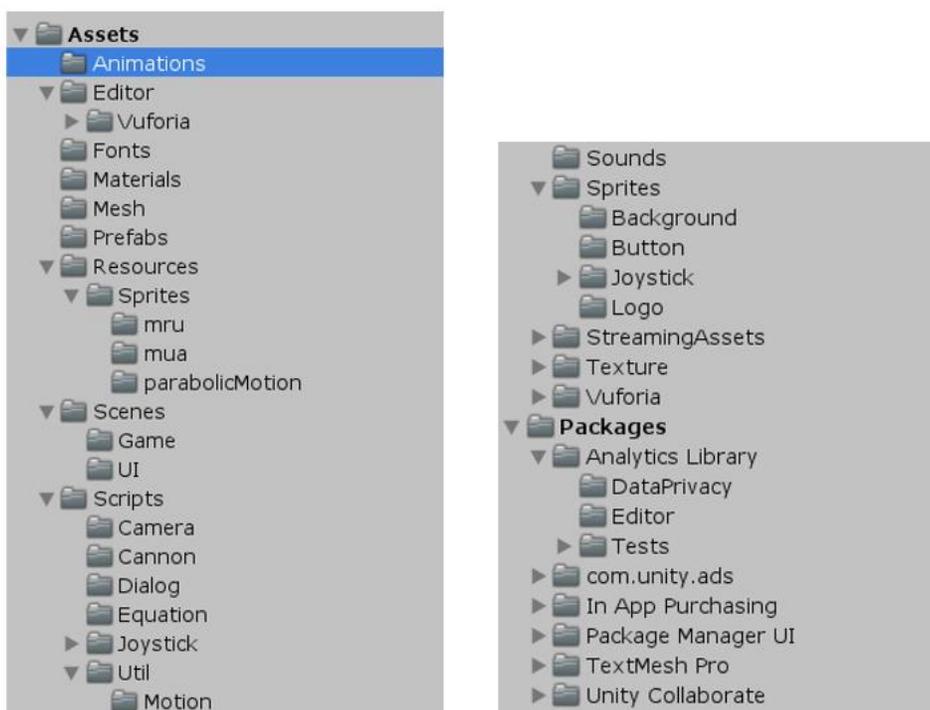


Figura 3. Distribución de paquetes a nivel de código

Del proyecto publicado en GitHub *ProjectileShooting* se utilizan y modifican las siguientes clases y modelos que se adaptan al aplicativo para facilitar el proceso de disparo en el módulo de tiro parabólico:

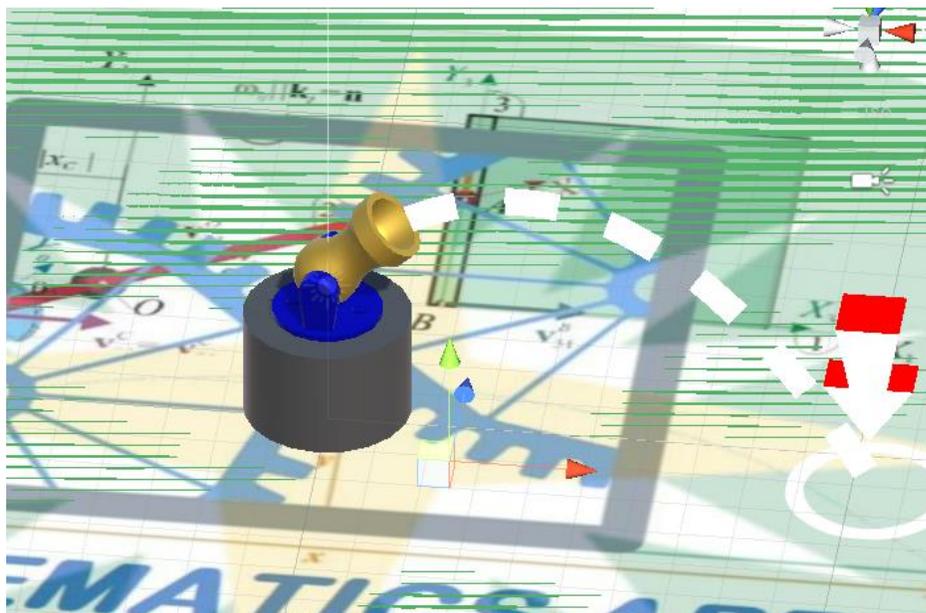


Figura 4. Modelos cañón y cursor para el tiro parabólico

Los modelos tridimensionales para los objetos del cañón y cursor y la animación para proyectar el posible tiro que se muestran en la Figura 4, son posibles gracias a las siguientes clases tomadas del proyecto *ProjectileShooting*.

La Figura 5 muestra el método para controlar el tiro en el módulo de movimiento parabólico cuando se selecciona un ángulo inicial. Se genera una distancia, velocidad, tiempo de vuelo y el llamado a la simulación o prolongación del tiro.

```

public void SetTargetWithAngle(Vector3 point, float angle)
{
    useAngle = true;
    currentAngle = angle;

    Vector3 direction = point - firePoint.position;
    float yOffset = -direction.y;
    direction = Math3d.ProjectVectorOnPlane(Vector3.up, direction);
    float distance = direction.magnitude;

    currentSpeed = ProjectileMath.LaunchSpeed(distance, yOffset, Physics.gravity.magnitude, angle * Mathf.Deg2Rad);

    projectileArc.UpdateArc(currentSpeed, distance, Physics.gravity.magnitude, currentAngle * Mathf.Deg2Rad, direction, true);
    SetTurret(direction, currentAngle);

    currentTimeOfFlight = ProjectileMath.TimeOfFlight(currentSpeed, currentAngle * Mathf.Deg2Rad, yOffset, Physics.gravity.magnitude);
}

```

Figura 5. Método que controla el tiro parabólico dado un ángulo inicial

La Figura 6 muestra el método que controla el tiro en el módulo de movimiento parabólico cuando se selecciona una velocidad inicial como información de entrada. Se genera una distancia, velocidad, tiempo de vuelo y el llamado a la simulación o prolongación del tiro.

```

public void SetTargetWithSpeed(Vector3 point, float speed, bool useLowAngle)
{
    useAngle = false;
    currentSpeed = speed;

    Vector3 direction = point - firePoint.position;
    float yOffset = direction.y;
    direction = Math3d.ProjectVectorOnPlane(Vector3.up, direction);
    float distance = direction.magnitude;

    float angle0, angle1;
    bool targetInRange = ProjectileMath.LaunchAngle(speed, distance, yOffset, Physics.gravity.magnitude, out angle0, out angle1);

    if (targetInRange)
        currentAngle = useLowAngle ? angle1 : angle0;

    projectileArc.UpdateArc(speed, distance, Physics.gravity.magnitude, currentAngle, direction, targetInRange);
    SetTurret(direction, currentAngle * Mathf.Rad2Deg);

    currentTimeOfFlight = ProjectileMath.TimeOfFlight(currentSpeed, currentAngle, -yOffset, Physics.gravity.magnitude);
}

```

Figura 6. Método que controla el tiro parabólico dada una velocidad inicial

En la Figura 7 se muestra el método de disparo en la clase que controla el cañón, el cual se encarga de generar una instancia del proyectil con el ángulo y velocidad determinados y realizar la llamada a la animación de formar una parábola.

```
31 public void Fire()
32 {
33     if (Time.time > lastShotTime + cooldown)
34     {
35         GameObject p = Instantiate(projectilePrefab, firePoint.position, Quaternion.identity);
36         p.GetComponent<Rigidbody>().velocity = turret.up * currentSpeed;
37
38         Instantiate(cannonFirePrefab, smokePuffPoint.position, Quaternion.LookRotation(turret.up));
39
40         lastShotTime = Time.time;
41         lastShotTimeOfFlight = currentTimeOfFlight;
42
43         anim.Rewind();
44         anim.Play();
45     }
46 }
```

Figura 7. Método de disparo en el módulo de tiro parabólico

La clase que se evidencia en la Figura 8 es la encargada de realizar la animación del arco y validar, si con la velocidad inicial dada se puede llevar al objetivo o en un caso diferente renderizar los puntos del arco de color rojo.

```

    * @author https://roystan.net/
    */
public class ProjectileArc : MonoBehaviour
{
    [SerializeField]
    int iterations = 20;

    [SerializeField]
    Color errorColor;

    private Color initialColor;
    private LineRenderer lineRenderer;

    void Awake()
    {
        lineRenderer = GetComponent<LineRenderer>();
        initialColor = lineRenderer.material.color;
    }

    public void UpdateArc(float speed, float distance, float gravity, float angle, Vector3 direction, bool valid)
    {
        Vector2[] arcPoints = ProjectileMath.ProjectileArcPoints(iterations, speed, distance, gravity, angle);
        Vector3[] points3d = new Vector3[arcPoints.Length];

        for (int i = 0; i < arcPoints.Length; i++)
        {
            points3d[i] = new Vector3(0, arcPoints[i].y, arcPoints[i].x);
        }

        lineRenderer.positionCount = arcPoints.Length;
        lineRenderer.SetPositions(points3d);

        transform.rotation = Quaternion.LookRotation(direction);

        lineRenderer.material.color = valid ? initialColor : errorColor;
    }
}

```

Figura 8. Clase que controla la simulación del arco en el módulo de tiro parabólico

6.2. Diseño

Se definen los requisitos que deben cumplir las interfaces de usuario, los cuales se desarrollan paulatinamente con un mayor enfoque en diseños simplistas, colores representativos de la universidad, orientación *landscape* y *responsive*, que se resumen en los requisitos funcionales y no funcionales de la aplicación.

6.2.1. Requisitos funcionales

RF1. La aplicación hace uso del sistema internacional en cuanto al manejo de las unidades.

RF2. La aplicación cuenta con los módulos de movimiento rectilíneo, movimiento acelerado y tiro parabólico.

RF3. En cada módulo se encuentra una opción de información para dar instrucciones de uso.

RF4. La aplicación tiene un botón de inicio con el fin de dar comienzo al juego.

RF5. La aplicación posee un sonido de fondo para la ambientación y una opción para quitarlo si el usuario lo desea.

RF6. Cuando sea necesario el uso de la cámara del dispositivo, existe una opción para habilitar el flashlight.

RF7. La aplicación cuenta con una opción de información donde se presente el objetivo del juego.

6.2.2. Requisitos no funcionales

RNF1. La herramienta es una aplicación móvil.

RNF2. La aplicación tiene orientación landscape.

RNF3. La aplicación es responsive, con el fin de tener una única presentación de los componentes en diferentes resoluciones de pantalla.

RNF4. La aplicación se ejecuta en dispositivos con sistema operativo Android 4.1 o superior.

RNF5. La aplicación posee un logo que represente a la Universidad Católica de Oriente.

RNF6. La aplicación está escrita en inglés con el fin de ser internacional.

RNF7. La aplicación implementa RA para iniciar proyectos con este tipo de tecnología en la Universidad Católica de Oriente.

6.3. Implementación

Con la ayuda de Unity se realiza la creación de las interfaces de usuario, los scripts para el flujo y manejo de la app son elaborados en el lenguaje de programación C#, se importan las librerías

externas de Vuforia y algunos scripts y modelos que se modifican y adaptan del proyecto ProjectileShooting publicado en GitHub como las clases y modelos de la Figura 4 a la Figura 8.

En el código se encuentran ciertas clases principales implementadas en el juego. Algunas desarrolladas para funcionalidades en específico y otras que son transversales a todo el aplicativo.

La Figura 9 muestra la clase para el control del posicionamiento del cursor en el módulo de movimiento parabólico y se valida que no supere el límite con el menú principal de la pantalla.

```
using UnityEngine;

/**
 * Clase para el control del cursor
 * @author Duvan Giraldo
 */
public class Cursor : MonoBehaviour
{
    void Update()
    {
        if (Input.mousePosition.y < 530)
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

            RaycastHit hit;
            if (Physics.Raycast(ray, out hit, float.MaxValue, 1 << LayerMask.NameToLayer("Ground")))
            {
                transform.position = hit.point;
            }
        }
    }
}
```

Figura 9. Clase encargada del control del cursor

La clase que se muestra en la Figura 10 sirve para controlar el autofocus de la cámara cuando se ejecuta el *tracking* de Vuforia.

```

public class CameraFocusController : MonoBehaviour
{
    void Start()
    {
        var vuforia = VuforiaARController.Instance;
        vuforia.RegisterVuforiaStartedCallback(OnVuforiaStarted);
        vuforia.RegisterOnPauseCallback(OnPaused);
    }

    private void OnVuforiaStarted()
    {
        CameraDevice.Instance.SetFocusMode(
            CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
    }

    private void OnPaused(bool paused)
    {
        if (!paused) // resumed
        {
            // Set again autofocus mode when app is resumed
            CameraDevice.Instance.SetFocusMode(
                CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUSAUTO);
        }
    }
}

```

Figura 10. Clase encargada del enfoque de la cámara

La Figura 11 muestra la clase para activar o inactivar el flash del dispositivo.

```

using UnityEngine;

/**
 *Clase para controlar el flash del dispositivo
 * @author Duvan Ignacio Giraldo Montoya
 */
public class FlashCamera : MonoBehaviour
{
    public GameObject btnON;
    public GameObject btnOFF;

    void Start()
    {
        Vuforia.CameraDevice.Instance.SetFlashTorchMode(false);
        FlashOff();
    }

    public void FlashOn()
    {
        Vuforia.CameraDevice.Instance.SetFlashTorchMode(true);
        btnOFF.SetActive(true);
        btnON.SetActive(false);
        Debug.Log("Flash on");
    }

    public void FlashOff()
    {
        Vuforia.CameraDevice.Instance.SetFlashTorchMode(false);
        btnOFF.SetActive(false);
        btnON.SetActive(true);
        Debug.Log("Flash off");
    }
}

```

Figura 11. Clase para el control de flashlight de la cámara

La clase de la Figura 12 controla los mensajes que se muestran en la opción de información en cada módulo, se controla la animación y los diálogos a mostrar.

```
public class DialogueManager : MonoBehaviour {  
  
    public Text nameText;  
    public Text dialogueText;  
  
    public Animator animator;  
  
    private Queue<string> sentences;  
  
    // Use this for initialization  
    void Start () {  
        sentences = new Queue<string>();  
    }  
  
    public void StartDialogue (Dialogue dialogue) ...  
  
    public void DisplayNextSentence () ...  
  
    IEnumerator TypeSentence (string sentence) ...  
  
    public void EndDialogue()  
    {  
        animator.SetBool("isOpen", false);  
    }  
}
```

Figura 12. Clase encargada del diálogo informativo que tiene cada módulo

La Figura 13 muestra el control de los mensajes y opciones que se presentan en la interfaz de información de la Figura 20.

```

public class AboutController : MonoBehaviour
{
    public GameObject textInfo;
    public GameObject textControls;

    void Start()
    {
        setTextInfoToAbout();
    }

    public void setTextInfoToAbout()
    {
        textInfo.SetActive(true);
        textControls.SetActive(false);
    }

    public void setTextInfoToMoreInfo()
    {
        textInfo.SetActive(false);
        textControls.SetActive(true);
    }

    public void OpenUrlDownloadIconKinematicsApp()
    {
        Application.OpenURL("https://i.ibb.co/Pz416gP/icon-Kinematcs-App.png");
    }
}

```

Figura 13. Clase para controlar la interfaz informativa

Cada módulo de la aplicación, a medida que se recogen monedas u objetos en el caso del movimiento parabólico, se renderizan algunas imágenes informativas acerca del movimiento que se realiza, con el fin de dar una mayor información al jugador como fórmulas y explicación teórica a cerca del movimiento. La clase encargada de controlar el cambio de una imagen a otra se presenta en la Figura 14.

```

/**
 * Class for image change in the Motion Information of URM and UAM
 */
public class ChangeImage : MonoBehaviour
{
    public Image UIImage;
    public string route;

    void Start()
    {
        UIImage = GameObject.Find("ChangedImage").GetComponent<Image>();
        UIImage.gameObject.SetActive(false);
    }

    void OnCollisionEnter(Collision col)
    {
        if (col.gameObject.name == "Player")
        {
            UIImage.sprite = Resources.Load<Sprite>(route);
            UIImage.gameObject.SetActive(true);
        }
    }
}

```

Figura 14. Clase para controlar el cambio de las imágenes

La clase que se expone en la Figura 15 es transversal en todas las escenas o interfaces del aplicativo. Es la encargada de mostrar la barra de progreso, controlar si existe información de tiro, el tiempo que demora el título al iniciar un módulo y un método para salir de la aplicación.

```

void Start()
{
    StartCoroutine(ActivationInfoLevelInit());
}

public void changeScene(string nameScene)
{
    StartCoroutine(LoadAsynchronously(nameScene));
}

IEnumerator LoadAsynchronously(string nameScene)
{
    AsyncOperation operation = SceneManager.LoadSceneAsync(nameScene);
    loadingScreen.SetActive(true);
    while (!operation.isDone)
    {
        float progress = Mathf.Clamp01(operation.progress / .9f);
        slider.value = progress;
        textProgress.text = System.Math.Round(progress, 2) * 100f + "%";
        yield return null;
    }
}

public void ShowShotInfo(bool showInfoShot){
    if (showInfoShot){
        infoShotComponent.SetActive(true);
    }
    else{
        infoShotComponent.SetActive(false);
    }
}

public void QuitGame()
{
    Application.Quit();
}

private IEnumerator ActivationInfoLevelInit()
{
    buttonInit.SetActive(true);
    yield return new WaitForSeconds(3);
    buttonInit.SetActive(false);
}
}

```

Figura 15. Clase utilitaria para todas las interfaces

La clase para la asignación y habilitación del sonido se presenta en la Figura 16.

```
public class SoundConstant : MonoBehaviour {  
    private static SoundConstant instance = null;  
    public AudioSource sound;  
    public static SoundConstant Instance  
    {  
        get { return instance; }  
    }  
    void Awake()  
    {  
        if (instance != null && instance != this)  
        {  
            Destroy(this.gameObject);  
            return;  
        }  
        else  
        {  
            instance = this;  
        }  
        DontDestroyOnLoad(this.gameObject);  
    }  
}
```

Figura 16. Clase para el control del sonido

Además del desarrollo del código para una versión 1 del aplicativo, se realizan pruebas de caja negra para validar el correcto funcionamiento y visibilidad de la aplicación en dos dispositivos con diferentes resoluciones, 1280x720 y 2340x1080 respectivamente, lo cual garantiza un diseño adaptable a cualquier dispositivo Android superior a 4,1, y un target de 15cmx15cm para una mejor visualización de los objetos.

7. RESULTADOS

Se crea el logo de la aplicación que se muestra en la Figura 17 con el programa *Photoshop*, teniendo en cuenta la representación de colores, la estrella de la Universidad Católica de Oriente y los temas de movimiento rectilíneo uniforme, movimiento uniformemente acelerado y movimiento parabólico.

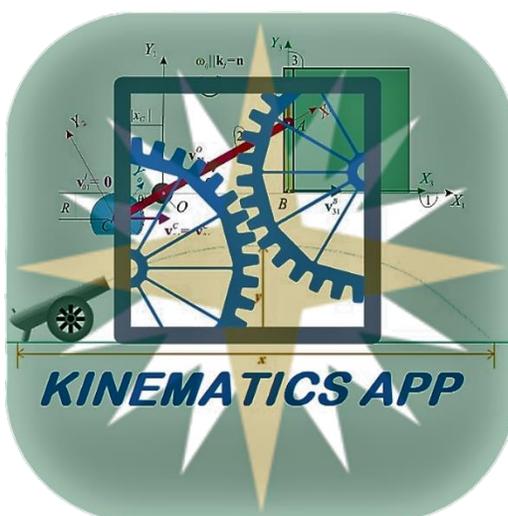


Figura 17. Logo de KinematicsApp

Se desarrolla un aplicativo móvil RA con el nombre *KinematicsApp* en el motor de videojuegos Unity con la ayuda de la librería Vuforia para la integración de la RA que facilita y sirve como apoyo del proceso formativo en los temas de movimiento rectilíneo uniforme, movimiento uniformemente acelerado y movimiento parabólico.

Se construyen dos versiones de la aplicación móvil, una en español y la otra en inglés, con tres opciones principales, cada una de ellas representa los temas de investigación. El usuario puede experimentar las diferencias que existen entre los temas de investigación de una manera didáctica.

KinematicsApp consta de las interfaces que se muestran de la Figura 18 a la Figura 24. Se considera un diseño simple, adaptable y fácil de entender por el usuario.

Interfaz de bienvenida: Es la pantalla inicial del aplicativo, muestra el título, logo y un botón de inicio que lleva a la página principal como se presenta en la Figura 18.



Figura 18. Interfaz de bienvenida de KinematicsApp

Interfaz principal: Consta de los tres módulos del aplicativo, Uniform Rectilinear Motion, Uniform Accelerated Motion y Parabolic Motion. Además, un botón para ver la información general de la aplicación, uno para salir del juego y otro para habilitar o deshabilitar el sonido como se muestra en la Figura 19.

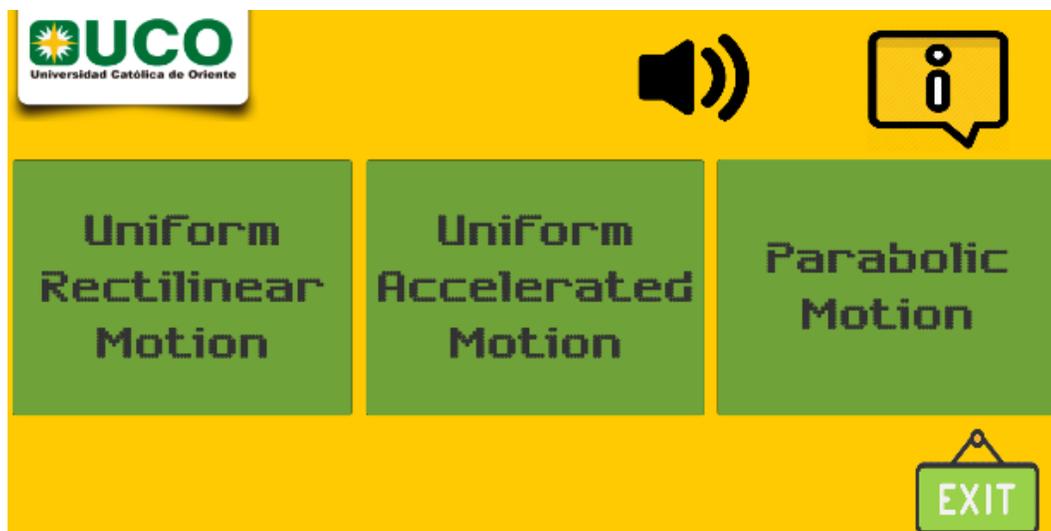


Figura 19. Interfaz principal de KinematicsApp

Interfaz informativa: Se visualiza el logo de la Universidad Católica de Oriente, cuatro botones, para visualizar el target, para ver la función de los controles que existen en la aplicación y de retroceso a la página principal y una información básica del juego como url de descarga del marcador y a quién va dirigida la aplicación como se muestra en la Figura 20.

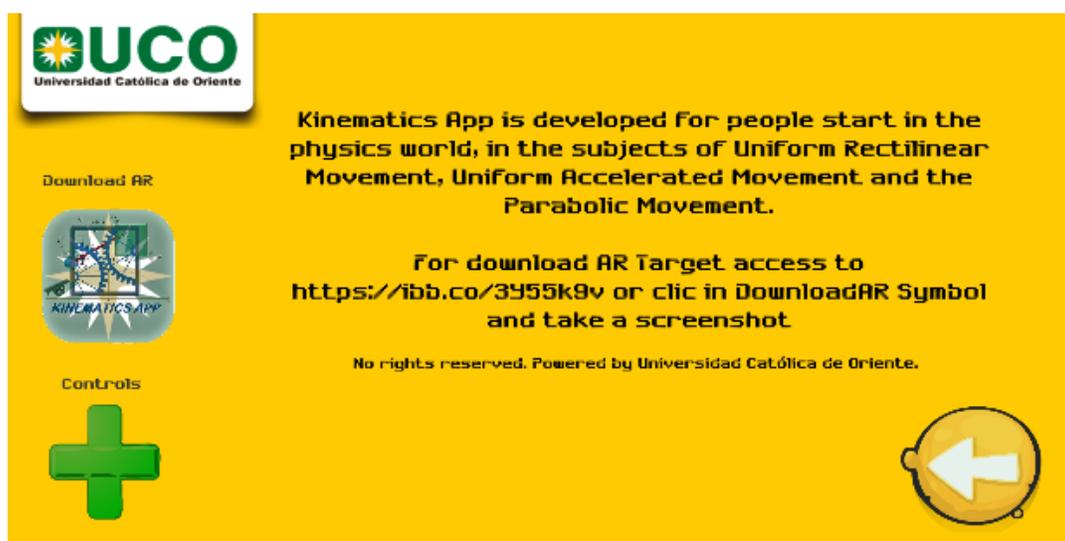


Figura 20. Interfaz informativa de KinematicsApp

Uniform Rectilinear Motion: Al dirigirse a la opción de Uniform Rectilinear Motion del menú principal, se despliega el módulo de movimiento rectilíneo uniforme, donde se encuentran cuatro botones, para ir al menú principal, ver la información de uso, activar/inactivar el flash de la cámara y restaurar la interfaz respectivamente; también, existe un campo para variar la velocidad de la esfera, un texto para visualizar el puntaje o número de monedas recogidas y una palanca para mover la esfera en cualquier parte la pantalla fuera del menú como se muestra en la Figura 21.

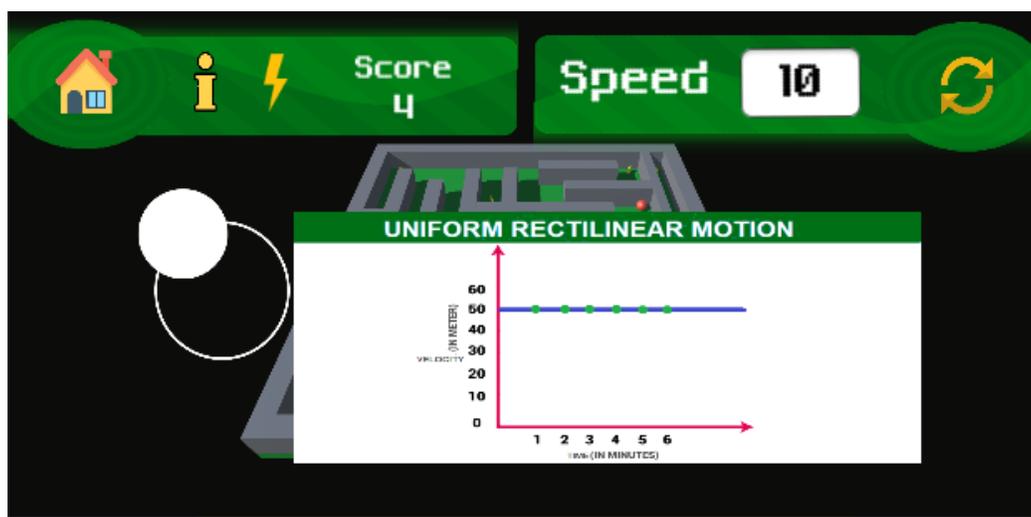


Figura 21. Opción de Movimiento rectilíneo uniforme de KinematicsApp

Uniform Accelerated Motion: Al dirigirse a la opción de Uniform Accelerated Motion del menú principal, se despliega el módulo de movimiento uniforme acelerado, donde se encuentran cuatro botones, para ir al menú principal, ver la información de uso, activar/inactivar el flash de la cámara y restaurar la interfaz respectivamente; también, existe un campo para variar la aceleración de la esfera, un texto para visualizar el puntaje o número de monedas recogidas, otro para ver la variación de la velocidad y una palanca para mover la esfera en cualquier parte la pantalla fuera del menú como se muestra en la Figura 22.

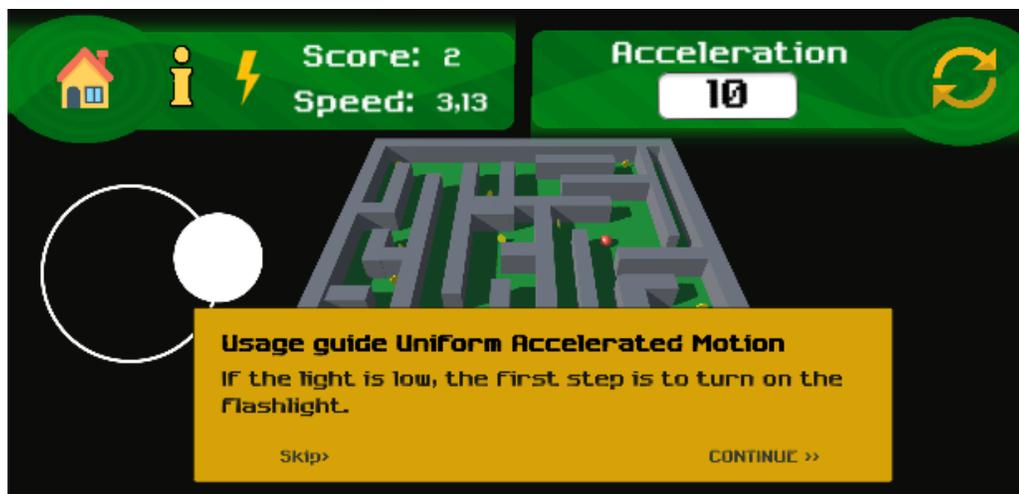


Figura 22. Opción de Movimiento rectilíneo acelerado de KinematicsApp

Parabolic Motion: Al dirigirse a la opción Parabolic Motion del menú principal, se despliega el módulo de movimiento parabólico, donde se encuentran cinco botones, para ir al menú principal, ver la información de uso, activar/inactivar el flash de la cámara, usar la velocidad inicial o usar el ángulo inicial y restaurar la interfaz respectivamente; también, existe un campo para variar la velocidad o el ángulo de la esfera según sea el caso elegido como se muestra en la Figura 23 y Figura 24.



Figura 23. Movimiento Parabólico en la opción de ángulo inicial



Figura 24. Movimiento Parabólico en la opción de velocidad inicial

8. TRABAJO FUTURO

A lo largo del desarrollo del aplicativo se tienen diferentes opciones para las interfaces y el modo de juego (véase el Anexo B) las cuales avanzan y actualizan paulatinamente a lo largo de su codificación.

Una funcionalidad muy pertinente que se requiere implementar en el desarrollo del aplicativo es capturar el plano en el que está el juego de manera automática sin la necesidad de un marcador, con el fin de tener un ambiente más real y una interacción virtual del videojuego más amigable.

A nivel práctico, la captura de planos no se agrega a la aplicación, ya que el dispositivo físico en el que corre el aplicativo debe garantizar la detección de planos físicos; así que no se pudo realizar el Ground Plane (véase el Anexo C) para la detección de planos. Vuforia, en su página oficial recomienda una serie de dispositivos para correr las aplicaciones creadas con el kit de desarrollo y los dispositivos que soportan la funcionalidad (véase el Anexo A). Otra razón muy

importante por la cual no se implementa, es por el coste de los dispositivos que la soportan y no se garantiza la versión mínima de Android 4.1 lo que ocasiona un nivel menor de personas que pueden tener la aplicación funcional de manera óptima en su dispositivo.

De igual forma, como trabajo futuro o una segunda versión para KinematicsApp, se propone implementar *ground plane* (véase el Anexo C) con librerías propias de Unity como ARCore, ARKit o AR Foundation para tener una mayor interacción entre el mundo real y virtual al disparar, mover una esfera o cañón a cualquier lugar real y poder reflejar los cambios o alteraciones en el dispositivo.

Aunque la aplicación sirve de una manera óptima en orientación landscape (Horizontal), es recomendable adaptarla para usarle tanto landscape como portrait (Vertical) haciéndola más flexible para el usuario.

Para una futura versión, también se pueden integrar nuevos temas con relación a la cinemática como el movimiento circular, pendular y un módulo para el tema vectorial; lo que genera un aumento en la cantidad de componentes integrales que sirven de ayuda en el proceso de enseñanza-aprendizaje de la cinemática.

9. CONCLUSIÓN

Las personas cuentan con un recurso didáctico llamado KinematicsApp, que permite visualizar los movimientos rectilíneo uniforme, acelerado y movimiento parabólico y apoya el proceso de enseñanza-aprendizaje del estudiante y profesor. KinematicsApp sirve como base tecnológica para futuros desarrollos o mejoras con RA especialmente en la Universidad Católica de Oriente, donde actualmente se promueve el desarrollo de aplicaciones fabricadas con RA. Finalmente, se determina que se pueden crear aplicativos capaces de realizar muchas más funcionalidades y una mayor complejidad, pero el coste puede ser un elevado número de recursos, no serán tan accesibles hoy en día y muchas personas posiblemente no están dispuestos a pagarlo. La RA acerca a las personas a mundos quizás poco posibles, al recrear cosas u objetos del mundo real para ser modificados y trasladados a un mundo virtual.

REFERENCIAS BIBLIOGRÁFICAS

- Byjus. (2019). *Rectilinear Motion of Particles*. Obtenido de Byjus: <https://byjus.com/physics/rectilinear-motion-of-particles/>
- Byjus. (2019). *Uniformly Accelerated Motion - Constant Acceleration*. Obtenido de Byjus: <https://byjus.com/physics/uniformly-accelerated-motion/>
- Cárdenas, N. (2017). *Desarrollo de una aplicación móvil en realidad aumentada para la nueva familia de billetes*. Obtenido de Repository UCC: https://repository.ucc.edu.co/bitstream/20.500.12494/4147/1/2017_desarrollo_aplicacion_movil.pdf
- Cuendet, S., Bonnard, Q., Do-Lenh, S., & Dillenbourg, P. (2013). *Designing augmented reality for the classroom*. *Computers & Education*.
- Fernández Aedo, R. R., Server García, P. M., & Cepero Fadruga, E. (2001). *El aprendizaje con el uso de las nuevas tecnologías de la información y las comunicaciones*.
- Gutiérrez, J. M., Fabiani, P., Benesova, W., & Meneses, M. D. (2015). *Augmented reality to promote collaborative and autonomous learning in higher education*.
- Kaufmann, H. (2000). *Construct3D: An Augmented Reality Application for Mathematics and Geometry Education*. Obtenido de ResearchGate: https://www.researchgate.net/publication/2570081_Construct3D_An_Augmented_Reality_Application_for_Mathematics_And_Geometry_Education
- Lawrence Robinson, M. S. (2019). *The Benefits of Play for Adults*. Obtenido de HelpGuide: <https://www.helpguide.org/articles/mental-health/benefits-of-play-for-adults.htm>
- Luttecke, C. (2014). *¿Sabes que es UNITY?* Obtenido de Zenva: <https://deideaaapp.org/sabes-que-es-unity-descubrelo-aqui/>
- Marr, B. (06 de 2020). *The Top 5 Tech Trends That Will Disrupt Education In 2020 - The EdTech Innovations Everyone Should Watch*. *Forbes*.
- Microsoft. (2019). *Welcome to the Visual Studio IDE*. Obtenido de Microsoft Docs: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- Network World. (2019). *El IoT moverá 7 trillones de dólares en 2025*. Obtenido de networkworld: <https://www.networkworld.es/convergencia/el-iot-movera-7-trillones-de-dolares-en-2025#:~:text=El%20n%C3%BAmero%20de%20dispositivos%20de,que%20significa%20127%20al%20segundo.>
- Oliver, A. (2016). *Creación de un videojuego de Realidad Aumentada multisensorial para niños de entre 3 y 11 años*.
- The Editors of Encyclopaedia Britannica. (2017). *Kinematics*. Obtenido de Encyclopaedia Britannica: <https://www.britannica.com/science/kinematics>

Unity. (2019). *What is AR, VR, MR, XR, 360?* Obtenido de Unity: <https://unity3d.com/what-is-xr-glossary>

Vuforia. (2019). *Device Tracking*. Obtenido de Vuforia Developer Library: <https://library.vuforia.com/articles/Training/device-tracking.html>

Vuforia. (2020). *Ground Plane User Guide*. Obtenido de Vuforia Developer Library: <https://library.vuforia.com/content/vuforia-library/en/articles/Training/ground-plane-guide.html>

ANEXOS

Anexo A

Para visualizar la lista de dispositivos con los que es compatible la librería Vuforia, se debe dirigir a la página oficial del portal de desarrollador de Vuforia developer.vuforia.com o library.vuforia.com/content/vuforia-library/en/platform-support/vuforia-engine-recommended-devices.html

Anexo B

Evolución de las interfaces de usuario y modo de juego del aplicativo KinematicsApp

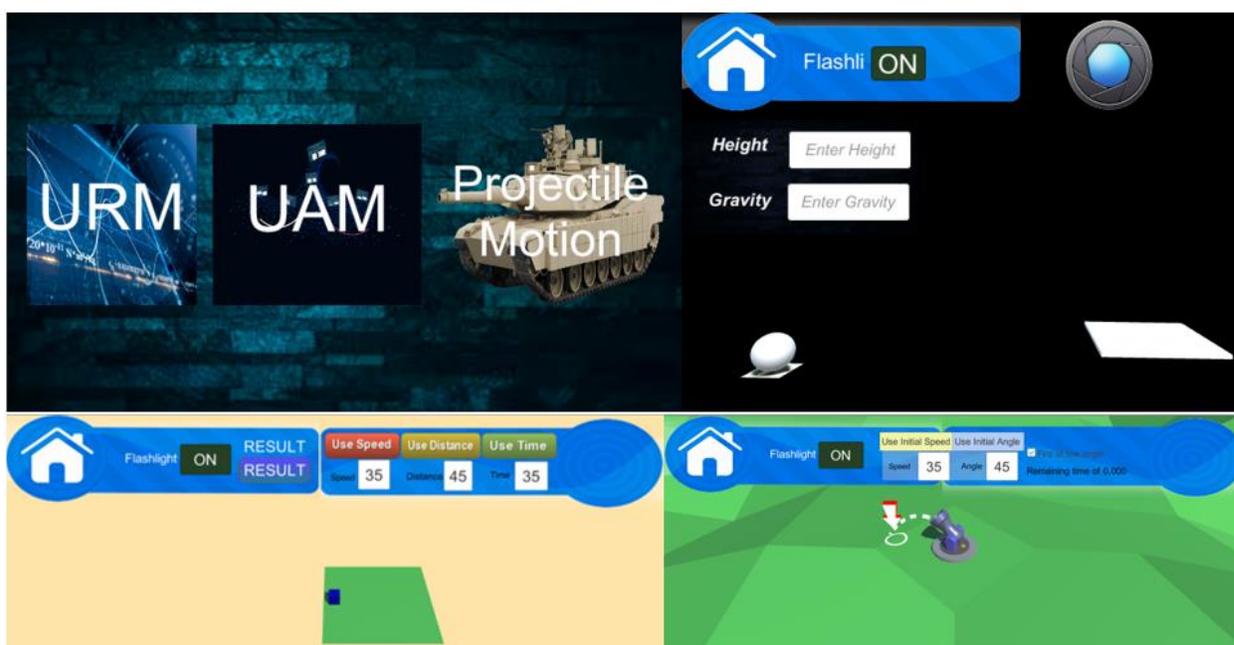


Figura 25. Interfaces y módulos Versión 1 KinematicsApp



Figura 26. Interfaces y módulos Versión 2 KinematicsApp

Anexo C

Ground Plane para la detección de superficies planas

Vuforia Ground Plane como parte de Smart Terrain permite que el contenido digital se coloque en superficies horizontales en su entorno, como pisos y mesas. Admite la detección y el seguimiento de superficies horizontales, y también le permite colocar contenido en el aire utilizando puntos de anclaje (Vuforia, 2020).



Figura 27. Representación de Ground Plane (Vuforia, 2020)