

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Física móvil: Aplicación móvil de choques elásticos e inelásticos enfocada al aprendizaje usando Google Cardboard

Trabajo de graduación presentado por Freddie Adriano Batlle Valdez para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2019

Física móvil: Aplicación móvil de choques elásticos e inelásticos enfocada al aprendizaje usando Google Cardboard

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



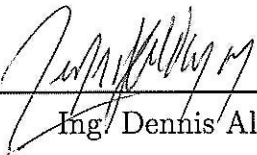
Física móvil: Aplicación móvil de choques elásticos e inelásticos enfocada al aprendizaje usando Google Cardboard

Trabajo de graduación presentado por Freddie Adriano Batlle Valdez para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

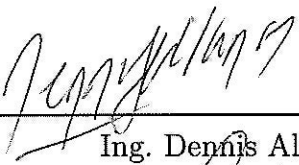
Guatemala,

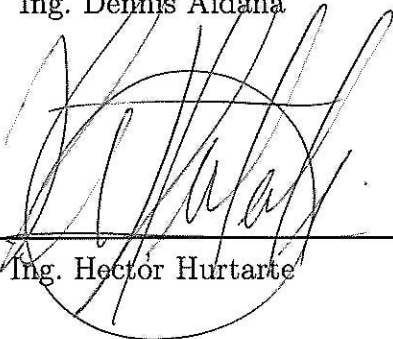
2019

Vo.Bo.:

(f) 
Ing. Dennis Aldana

Tribunal Examinador:

(f) 
Ing. Dennis Aldana

(f) 
Ing. Hector Hurtarte

(f) 
MSc. Douglas Barrios

Fecha de aprobación: Guatemala, 8 de Noviembre de 2019.

Lista de figuras	VIII
Lista de cuadros	IX
Resumen	XI
Abstract	XIII
I. Introducción	1
II. Antecedentes	3
III. Justificación	5
IV. Objetivos	7
A. Objetivo general	7
B. Objetivos específicos	7
V. Marco teórico	9
A. Aprendizaje	9
1. La diversión impacta el aprendizaje	9
2. La curiosidad mejora el aprendizaje	10
3. Actitud favorable del estudiante hacia la realidad virtual en la educación	10
4. El contexto en el que se aprende impacta la memoria	10
5. La visualización en el aprendizaje	11
6. “Retrieval-induced forgetting”	11
7. Aprendizaje en personas con necesidades especiales	11
VI. Física Móvil	13
A. Game engine	13
1. Input	13
2. Raycasting	14
3. Modelos	14
4. Variables	16

5.	Problemas con velocidad	18
6.	Cálculo de fps	18
7.	Menú e interfaces	18
8.	Canvas	20
9.	Colores de carros	22
10.	Retícula de selección	23
11.	Distancia	24
12.	Movimiento	24
13.	Visibilidad	25
14.	Vision aérea	25
15.	Carros fantasma y guardar eventos	26
B.	Investigación sobre aprendizaje	27
VII.Resultados		29
A.	Física Móvil	29
1.	Aplicación	29
2.	Pruebas con usuarios	33
B.	Aprendizaje	34
1.	Investigación	34
2.	Pruebas con usuarios	35
VIII.Conclusiones		47
IX. Recomendaciones		49
X. Bibliografía		51
XI. Anexos		55
A.	Datos extras encuestas	55
1.	Oportunidades de mejora	55
2.	Problemas usando la aplicación	56
3.	Distribución de carreras	56
4.	Uso previo de vr	57
5.	Sexo	57
6.	Cantidad de información en la app	58
7.	Dificultad para usar aplicación	58
8.	Lo mejor de la aplicación	59
B.	Encuesta	59
XII. Glosario		63

Lista de figuras

1.	Carro con pocos poligonos	15
2.	Teclado original	15
3.	Teclado modificado	16
4.	Material fisico de los carros	17
5.	Posicionamiento de menú	19
6.	Elementos “non-diegetic”	19
7.	Clasificación de interfaces	20
8.	“Aliasing”	21
9.	“Dynamic pixels per unit”	21
10.	Marcador con variables	22
11.	Colores de carros	22
12.	Retícula de selección	23
13.	Retícula de selección sobre elemento	23
14.	Distancia entre carros	24
15.	Imagen de visibilidad	25
16.	Imagen de vista aérea	26
17.	Imagen opción para guardar datos	26
18.	Imagen de carros fantasma	27
19.	Imagen objetos en Unity	29
20.	Imagen de la escena	30
21.	Imagen de la aplicación desde el celular	30
22.	Imagen del menú inicial	31
23.	Imagen del menú de carros	31
24.	Imagen de todo el menú	32
25.	Imagen del reseteo de menú	32
26.	Imagen de los resultados al momento del choque	33
27.	Gráfica de potencial usuarios	34
28.	Gráfica sobre diseño gráfico	35
29.	Gráfica sobre facilidad de visualizar problemas físicos	36
30.	Gráfica sobre facilidad de visualizar problemas de choque	37
31.	Gráfica sobre la facilidad de entender la relación entre variables	38
32.	Gráfica tipo pie sobre el conocimiento previo	38

33.	Segunda gráfica tipo pie sobre el conocimiento previo	39
34.	Gráfica sobre facilidad de visualizar problemas físicos post vr	39
35.	Gráfica sobre facilidad de visualizar problemas de choque post vr	40
36.	Gráfica sobre facilidad de la relación entre variables post vr	41
37.	Gráfica sobre la ayuda de la app para visualizar y entender	41
38.	Gráfica tipo pie sobre el conocimiento post uso vr	42
39.	Gráfica sobre el conocimiento previo	42
40.	Gráfica sobre el conocimiento post uso vr	43
41.	Segunda gráfica tipo pie sobre el conocimiento post vr	44
42.	Segunda gráfica sobre el conocimiento previo	44
43.	Segunda gráfica sobre el conocimiento post uso vr	45
44.	Gráfica sobre la ayuda de la app para responder los problemas	45
45.	Gráfica sobre la ayuda al resolver problemas de choque	46
46.	Gráfica de oportunidades de mejora	55
47.	Gráfica de problemas en app	56
48.	Gráfica distribución de carreras	56
49.	Gráfica de uso previo de vr	57
50.	Gráfica de sexo de encuestados	57
51.	Gráfica sobre cantidad de información en la app	58
52.	Gráfica de la dificultad de uso de la aplicación	58
53.	Gráfica de lo mejor de la aplicación	59

Lista de cuadros

1. Cuadro de variables posibles de ingresar 32
2. Cuadro de variables en el marcador 33

Es de interés para cualquier casa de estudio que sus estudiantes terminen las clases y se gradúen. Ciertamente las clases más complicadas son las de matemática y física. Proveer una herramienta inicial para apoyar a los estudiantes en temas complicados es imperativo. Se creó una aplicación móvil de realidad virtual para el tema físico de choques usando el Google Cardboard y se hizo una investigación en el tema de aprendizaje para encontrar como esta puede influenciar al momento de aprender. Haciendo una serie de preguntas a una población representativa se encontró que el 33% de los encuestados mejoraron su desempeño en la primera mitad de las preguntas después de usar la aplicación, mientras que 20% de las personas mejoraron en la segunda mitad. 83% de las personas reportaron haber encontrado de ayuda usar la aplicación para responder las preguntas. El movimiento de la cabeza para seleccionar valores en la aplicación fue una de las oportunidades de mejora reportado por los encuestados, es recomendable encontrar otra forma de manejar el input de los usuarios.

Any college has interest in getting students to finish their careers and graduate. Surely math and physics are some of the hardest classes. Providing an initial tool to help the students in complicated topics is imperative. A virtual reality mobile app was created for the physics topic elastic and inelastic collisions using Google Cardboard, an investigation on learning was also made, this helped in finding how the app could influence learning skills. With a series of questions to a representative population it was found that, after having used the app, 33% of participants improved their performance regarding the first half of the questions while 20% improved on the second half. 83% of people reported having found the app useful to answer the questions. Head movement to select values was one of the improvement opportunities reported by the participants. It's recommended to find another way to manage user input.

La realidad virtual no es un concepto nuevo, ha existido desde la década de los 90. No es sorprendente que dos décadas después esté incorporándose al mundo moderno. Desde un crecimiento muy alto en las proyecciones de ventas y ganancias usando esta tecnología hasta el uso terapéutico la realidad virtual se puede encontrar en muchos lados. Con este trabajo se quiere descubrir el impacto positivo que puede tener la realidad virtual en la educación.

Principalmente el efecto que la realidad virtual tiene en el proceso del aprendizaje. Se decidió hacer un estudio amplio de las investigaciones existentes relacionando realidad virtual y educación. Así como estudios en los que no estuvieran vinculadas estas dos variables, pero vincularlas tendría sentido. Además, se provee una aplicación para que el estudiante pueda interactuar en un mundo virtual con el concepto físico de choques. Esto servirá como un primer paso hacia la implementación de la realidad virtual en la educación en la universidad del Valle de Guatemala.

Se encontró que no solo hay impactos positivos, sino que hay una gran cantidad de estos, lo que nos deja con una gran cantidad de pros y una baja cantidad de contras en el uso de esta tecnología. La realidad virtual es una herramienta que facilita la visualización y esta es clave en el desarrollo del pensamiento crítico y a la vez esta es un determinante en la calidad del aprendizaje de una persona.

Es importante resaltar que no se busca sustituir el método tradicional de enseñanza, solamente se busca proveer una herramienta más para mejorar este método.

Jong-Heon Kim y sus colegas de la universidad Nacional de Kongju realizaron un trabajo de investigación en el año 2001 titulado *Virtual reality simulations in physics education*. En este llevaron a cabo una simulación en realidad virtual que ayuda a los estudiantes a aprender conceptos físicos como velocidad relativa y máquinas eléctricas a un nivel de bachillerato. Se concluyó que hay una necesidad de realizar más programas de realidad virtual para la educación a distancia, así como para el salón de clase. Los estudiantes muestran un mayor nivel de aprendizaje en el grupo que uso la simulación en realidad virtual, además el grupo que uso la simulación indicaron que estaban más satisfechos con su aprendizaje, entendieron mejor los temas y estaban más interesados en el tema. Esto es importante porque es un trabajo similar.

Hannes Kaufmann en el 2009 realizó un trabajo titulado *Physics Education in Virtual Reality: An Example*. En este presentan una aplicación en realidad virtual para la educación en el tema físico. Los estudiantes tienen la capacidad de crear sus ejemplos y simularlos. La conclusión es que los casos de uso educacionales muestran que tan útil pueden ser al ser integradas en las clases de física.

Bruce Ma y Peter Mak realizaron un trabajo en Hong-Kong llamado *Using virtual reality in teaching secondary school physics*. Se concluye que más estudios se pueden realizar para observar las reacciones y compararlo con la forma tradicional de enseñanza del pizarrón. Esperan que más maestros en el tema de física usen la aplicación.

Justificación

No ha habido intentos previos en la Universidad del Valle de Guatemala de una aplicación de VR/AR que sirva de ayuda al estudiante en el área de física. Esta aplicación se aprovecha de esta tecnología nueva para dar un primer paso hacia solucionar el problema del estudiante en el aprendizaje de temas relacionados con el departamento de física.

Esto es importante para la universidad ya que puede disminuir el número de estudiantes que pierden estas clases, esto sirve para tener más profesionales de alta calidad en Guatemala. Además de que la Universidad podrá usar esto para seguir mejorando y usando proyectos de innovación en el campo del estudio y las clases.

Como estudiante, saber que el material de apoyo que se está usando fue creado por otro estudiante, conecta esto con sentimientos de orgullo y con ideas de que lo que podrían hacer o están haciendo puede ser usado por muchas personas y da una sensación de querer hacer más con lo que se tiene. De esta manera se impulsa la creatividad y el deseo de hacer cosas en el estudiante.

Se escogió el tema físico de choques elásticos e inelásticos debido a la facilidad con la que se puede representar en una escena de 3D como lo son las de Unity, además de que muchas veces este evento físico se mira con carros y este tipo de modelo es muy común en la comunidad y por lo tanto no hay dificultad con conseguir el mismo para la aplicación. La dificultad de simular el mismo no es extremadamente difícil ni es pesada para una aplicación móvil. Por sobre todo esto es un tema interesante y causante de curiosidad por lo común que es de observar a diario.

A. Objetivo general

Construir una herramienta tecnológica de ayuda a los estudiantes en el tema físico: choques elásticos e inelásticos.

B. Objetivos específicos

- Programar una aplicación móvil, haciendo uso de Google Cardboard, con la que el usuario pueda interactuar y modificar para poder ejemplificar el problema de choques.
- Analizar la forma en la que la herramienta será de ayuda para el estudiante, investigando y describiendo cómo afecta en el aprendizaje, para mejorar el mismo.

A. Aprendizaje

1. La diversión impacta el aprendizaje

Si se hace una investigación y se busca la relación entre la diversión y el aprendizaje se encuentra que se habla mucho de esto, especialmente en niños. Lucardie (2014) menciona que esto no solo es en niños, también impacta adultos. La felicidad y la diversión son factores que impactan el hecho de atender a clase, así como el deseo de aprender el conocimiento y las habilidades. Además de que estos factores son considerados mecanismos que ayudan a la concentración y el aprendizaje. Hay claros indicadores de que emociones positivas tales como la diversión y la felicidad se vinculan con un aprendizaje exitoso.

Lightfoot y Brady (2005) encontraron que la gente menciona las cosas nuevas y emocionantes que aprendieron y la felicidad que esta les brindaba cuando hablaban de aprender. Bowman y Kearns (2007) mencionan que usar varios métodos de aprendizaje ayuda a tener un aprendizaje interesante y divertido. Davis (2001), también recomienda que enfocarse en la diversión al momento de aprender es bastante importante.

Podemos ver entonces que hay mucha literatura que relaciona la importancia del aprendizaje con la diversión y con distintos métodos de aprendizaje. Esto es importante dado que la aplicación de VR que se construyó puede categorizarse como una forma distinta de aprendizaje y debido a la forma de interacción con la misma podemos por lo menos decir que es más interesante y divertida que una clase magistral, siempre teniendo en cuenta que no es un sustituto sino una herramienta que ayuda en el salón para traer estos elementos de diversión y felicidad al estudiante, de esta forma mejorando el aprendizaje del tema dado las emociones que causa.

2. La curiosidad mejora el aprendizaje

Según Matthias, Bernard y Charan (2014) hay una mejoría al momento de recordar algo cuando se tiene una curiosidad por la información y de todo el material que se aprenda mientras está en este estado de curiosidad. Esto indica que un estímulo en la curiosidad al momento de aprender mejora el aprendizaje de la persona.

La curiosidad es un factor de motivación y este afecta la memoria. Si se incorpora la curiosidad en el proceso de aprendizaje se puede mejorar la memoria y a su vez el aprendizaje.

Una predicción, hecha por PwC Global, menciona que para el 2022 el crecimiento anual de ingresos en cuanto a VR habrá crecido un 40%. Esto es importante porque nos muestra cómo va a crecer el interés en esta tecnología y la curiosidad por la misma.

La curiosidad también provoca una mejora en la memoria cuando se hacen preguntas y el individuo se equivoca en la respuesta, mostrado por Kang (2008). Esto quiere decir que si el individuo responde preguntas a las que no tiene conocimiento con algún tipo de bonificación por no equivocarse mejorará en su memoria y retendrá más información sobre estas preguntas erróneas incluso 10 días después.

Esto es importante ya que el concepto de VR es algo que provoca curiosidad no solo por la implicación de usarlo sino porque es relativamente nuevo en el campo del aprendizaje (en cualquier campo de hecho). Si el individuo usa la aplicación, previo a algún conocimiento, y con un sistema de recompensa se puede mejorar el proceso de aprendizaje.

3. Actitud favorable del estudiante hacia la realidad virtual en la educación

Una investigación realizada por Mikropoulos, Chalkidis, Katsikis, y Emvalotis señala que los estudiantes están altamente motivados y demuestran una actitud favorable hacia la realidad virtual en la educación.

Esto se relaciona muy bien con el hecho de que la diversión, la curiosidad (puede definirse como una motivación a aprender) y felicidad impactan en el aprendizaje. Ya que los estudiantes encontraban retos y se emocionaban cuando interactuaban con el mundo virtual, esto sumado al hecho de que los distractores en un salón de clase se disminuyen inmensamente en esta virtualización ayudan a que el estudiante tenga un proceso de aprendizaje más completo y positivo.

4. El contexto en el que se aprende impacta la memoria

Un ambiente informal propicia que haya pocos miedos (miedo a preguntar, por ejemplo), minimización de la ansiedad y un aprendizaje más profundo. (Bowman 2006; Clemens 2003).

Este ambiente informal, dentro de un salón de la universidad puede propiciar el aprendizaje para el propósito universitario. Dado que, como lo menciona Nicole (2017), el ambiente en el que se aprende influencia al momento de recordar, es decir si se aprende en un ambiente

de universidad o de salón de clase con todo lo que esto implica, al momento de recordarlo en el mismo ambiente será más fácil recordarlo, pero si al momento de recordarlo se está en un ambiente de trabajo, por ejemplo, será más difícil recordarlo.

Lo que indica que si el aprendizaje con la aplicación se realiza en un ambiente de clases (dado que se desea que el aprendizaje sea en este ambiente) mejorara la memoria y el aprendizaje en este ámbito.

5. La visualización en el aprendizaje

Un estudio realizado por Kyvete y Kastriot (2017) concluye que el uso de la visualización en el proceso del aprendizaje tiene una influencia positiva en el desarrollo del pensamiento crítico. La visualización es un componente importante al momento de aprender y el pensamiento crítico determina la calidad del aprendizaje. Un aprendizaje basado en la interacción ayuda a estudiantes con diferentes estilos de aprendizaje a comprender teoría y lograr aplicarla.

Esto es algo positivo para el concepto de realidad virtual. Aquí podemos representar muchos conceptos y de esta manera ayudar al estudiante a lograr visualizarlos y comprenderlos.

6. “Retrieval-induced forgetting”

Karl y Christof (2007) mencionan que recordar cosas puede causar pérdida de memoria de las cosas que no se recordaron. Pero esto depende de si se está tratando de recordar algo específico o si se está haciendo un recordatorio por medio de una relación. Es decir, si trato de recordar algo específico en un grupo de cosas el cerebro va a ir eliminando las cosas que no estoy tratando de recordar en ese grupo, por ejemplo, una fórmula física específica entre todas las conocidas. Mientras que, si recuerdo algo a través de una relación se disminuye el riesgo de olvidar las demás cosas.

Esto es particularmente importante porque si se logra hacer una relación entre una experiencia nueva y un concepto y sus implicaciones, se puede disminuir el riesgo de olvidar otros conceptos o ejemplos. Lo cual es un beneficio al momento de experimentar con la aplicación de VR.

Esto viene sumado al hecho de que la experiencia que se tuvo es más vivida y por lo tanto es más fácil de recordar que un ejemplo leído en un libro.

7. Aprendizaje en personas con necesidades especiales

Desde hace ya muchos años se han usado los entornos virtuales para ayudar en el aprendizaje en personas con algún tipo de necesidad especial, ya sea alguna discapacidad física, en estas se usan entornos virtuales para ayudar a las personas a manejar una silla de ruedas y evitar muchos de los problemas y el estrés que causa aprenderlo de la manera tradicional.

Otro caso en el que la virtualización de un problema puede ayudar es en personas con

problemas en el aprendizaje. Durante la década pasada estos sistemas se enfocaron en simular situaciones de socialización sin embargo con el mayor énfasis que hay en la educación esto ha cambiado el enfoque de estos sistemas de un enfoque en la socialización a un enfoque en lo académico. Tara (2009) menciona que estos sistemas ayudan a experimentar eventos y esto puede ser efectivo en el aprendizaje de física, conceptos como las leyes de movimiento de Newton, energía cinética, entre otros. Este también es el caso para personas con déficit de atención y personas con traumas cerebrales.

A. Game engine

Se decidió usar Unity3d como “game engine” para el desarrollo de la aplicación con esto se aprovecharía el conocimiento previo de la herramienta, así como la gran cantidad de librerías de vr que son compatibles en caso de necesitar alguna. El proyecto inicialmente no usaba librerías, se inició un proyecto nuevo y se probó en una escena como compilar para Unity, el movimiento, las cámaras y cómo detectar el input del Google Cardboard.

Mientras se investigaba y buscaba información sobre las buenas prácticas al momento de usar vr en Unity, se encontró la librería NewtonVR. Esta brinda objetos de Unity con componentes de cámara, controles y “raycasting” para usarlos con vr, esto simplifica problemas, como la diferencia de controles entre plataformas (Google Cardboard solo tiene un input, pero Oculus Rift tiene varios). Esta librería sirvió para terminar de entender cómo funcionaban estos elementos en vr, pero un problema que no se tenía previsto fue que no era compatible con Google Cardboard. Por esta razón se terminó usando la librería default de Google VR.

1. Input

Una dificultad con hacer aplicaciones en vr es cómo manejar los inputs. En este caso particular hay un solo input, un botón en la caja del dispositivo. Lo curioso de este botón es que al presionarlo se activa un mecanismo el cual mueve una pequeña palanca que toca la pantalla del celular, por lo que el verdadero input es el touch del celular. El otro input posible es la cámara de la persona. Este es un objeto en la escena del cual podemos obtener su información: posición, dirección, altura, entre otros.

Para el input de la cámara se debe usar una función de raycasting. Esta enviará un rayo invisible desde la posición de la cámara hacia la dirección a la cual la cámara está viendo, en su paso detectará las colisiones con otros objetos y hará un listado de todos los objetos con los que colisiono. Esto quiere decir que en todo momento en la aplicación hay un rayo siendo emitido desde el jugador hasta el final de la escena y una revisión de si estos objetos son con los que interactúa el usuario, por ejemplo, los menús.

2. Raycasting

Unity trabaja con una función llamada update, esta es una función que se ejecuta una vez por frame (si un juego tiene 60 frames por segundo, comúnmente la cantidad de frame rate en juegos de consolas recientes, esta función se ejecutaría 60 veces por segundo). Por lo que en esta función es una buena idea manejar el raycasting, se llamará una vez por frame (“en todo momento”) y en cada frame que el jugador mueva la cámara se podrá captar los nuevos objetos con los que esta colisionando el raycasting.

Esto podría manejarse de otras maneras: en diferentes threads pero uno de los problemas de esto es que el API de Unity no es threadsafe, por lo que no nos garantiza que las variables que usemos entre los múltiples threads retendrán su valor al ser modificada en otro y se corre el riesgo de modificar datos en la memoria. Otra opción es usar coroutines, y dejar que la coroutine maneje el raycasting pero el propósito principal de esta función es poder ejecutar código durante varios frames, por lo que en el código de un frame se ejecuta hasta donde debe y le devuelve el control del flujo a Unity pero en el siguiente frame (o dependiendo de que condición se le dé, otra podría ser después de una cantidad de segundos) seguirá donde se quedó en la ejecución.

La otra manera de usar raycasting podría ser con el nuevo sistema de trabajos sistema de trabajos de C# de Unity. Este sistema permite usar procesadores multicore sin mucho problema, esto permite escribir código paralelo en Unity con su API. Algunos desarrolladores que lo han probado mencionan que pueden construir meshes y usar raycasts en paralelo. El problema aquí es que este sistema aún no está para el público y se debe obtener una invitación para usarlo. Una ventaja aquí es que permitiría tener más frame rate y mejor duración de batería en los celulares.

3. Modelos

Algo importante en esta plataforma fue encontrar los modelos correctos para los carros, no necesariamente tenían que ser carros sin embargo es uno de los ejemplos más comunes en estos problemas. Fue importante durante todo el desarrollo estar pendiente de la plataforma objetivo, una plataforma móvil no puede soportar lo mismo que una consola de videojuegos o una computadora moderna y teniendo en cuenta que aún se usan celulares de hace 5 años, hubo que pensar en estos y no cargar demasiado la escena. Debido a esto el realismo de la aplicación sería limitado, por lo que toda la escena debería tener un ambiente caricaturesco, de juguete o de un videojuego viejo. Se probó con múltiples modelos de carros, algunos eran demasiado caricaturescos y otros no servían o no tenían animaciones. Se terminó usando “Low Poly Destructible 2 Cars no. 8”. Como el nombre muestra tiene pocos polígonos lo

que no nos afecta mucho en el desempeño de la aplicación. El conteo de polígonos para este carro es de 3700 polígonos. El videojuego, para celulares, Real Racing 3 tiene cerca de 10,000 polígonos por carro. Este modelo viene en formato FBX, esto es bueno porque este formato permite contener jerarquía y animaciones.



Figura 1: Imagen del asset (Low Poly Destructible 2 Cars no. 8) usado para los carros

Luego de esto se necesitó implementar un teclado virtual. Se opto por usar la librería de “Curved VR Keyboard”. Este teclado mapeaba todo el abecedario, así como números y caracteres especiales, pero en las variables de los problemas de física solo se usan números y el punto. Se tuvo que modificar los scripts y assets del teclado para que solo tuviera números, punto y la tecla para borrar.



Figura 2: Imagen del asset (Curved VR Keyboard) original usado para el teclado

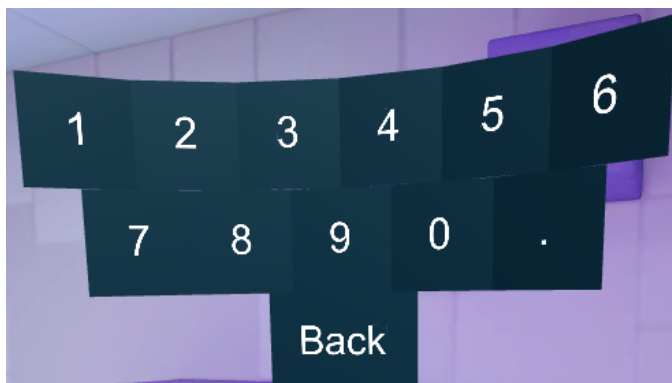


Figura 3: Imagen del asset (Curved VR Keyboard) modificado usado para el teclado

4. Variables

Para el control de la interacción de las variables físicas se usó el engine físico que Unity tiene. Para la selección de variable y la navegación en el menú se usaron botones de unity y raycasting de la cámara para ver si el usuario está viendo un botón o un input específico. Esto también se puede realizar con el botón del Google Cardboard, pero durante las pruebas este fallaba considerablemente debido a que los usuarios no lo apachaban lo suficientemente duro.

Las variables de física que el usuario puede modificar, en ambos carros, son las siguientes: velocidad, masa y coeficiente de elasticidad.

Se usó una función de impulso en donde se usa la masa del objeto para aplicar una fuerza de impulso instantáneo. La fuerza se aplica usando la formula 1. Esta formula de impulso-momentum es logicamente equivalente a la segunda ley de Newton.

$$I = m * v \tag{1}$$

Esta dependencia de la masa implica que entre más masa hay en un objeto más fuerza se necesitara para moverlo la misma cantidad que uno con menos masa. Dado que el experimento se basa en tener dos objetos colisionando a cierta velocidad constante, en vez de verlos acelerar hasta llegar a la velocidad y luego colisionar, esta función permite aplicar una fuerza instantánea para llegar, rápidamente, a la velocidad que el objeto debería tener.

El problema con aplicar una fuerza de impulso a un objeto es que debido a la fricción con la superficie y con el aire, el objeto eventualmente regresara a una posición de reposo, es decir tendrá una desaceleración durante el evento. Esto se puede solucionar de 2 formas en Unity. La primera solución es aplicar una fuerza de impulso en la función de Unity "FixedUpdate", se usa esta en vez de update porque esta se ejecuta exactamente en sincronía con el motor físico de Unity, mientras que update puede ejecutarse antes o después del motor físico.

El problema con aplicar la fuerza de esta manera es que la velocidad se acumula con la velocidad actual del objeto, debido a esto se debe hacer un cálculo antes de aplicar la fuerza o se puede revisar la velocidad del objeto y mantener la velocidad deseada haciendo uso de esta propiedad.

Sin embargo, debido a que estos procesos se hacen en la función de FixedUpdate puede tener repercusiones en dispositivos con un “gpu” malo, debido a que la cantidad de frames puede variar entre dispositivos. Este podría tener un efecto que es posible percibir dependiendo de la velocidad del carro, se notaría una desaceleración y luego una nueva aceleración para ajustar la velocidad perdida. Se debe tener en cuenta la fricción que se usa en el objeto, llamada drag. El drag del objeto puede ser 0 lo que permitiría ignorar la fricción cuando no hay contacto, pero también se debe colocar una fricción de 0 en la superficie.

Algo particular de la fórmula 1 es que se está aplicando la fuerza una vez por frame, el frame rate de la aplicación en un celular Motorola G5 Plus es de 60fps, esto quiere decir que se aplicaría la fuerza 60 veces por segundo. Debido a esto se tomaron las medidas necesarias para que solo se ejecute una vez en total, en el primer frame. También se pudo haber usado una función de fuerza que aplica la misma fuerza durante todos los frames de 1 segundo, pero surge el problema de tener que ejecutar código más veces y además cuando haya un cambio de frame rate no se aplicaría la misma cantidad de fuerza.

Unity provee la función de VelocityChange que le da un cambio en la velocidad de un objeto de forma repentina sin considerar la masa. Esta parecía una buena forma de usar la velocidad ingresada por el usuario, pero hay un problema. La documentación menciona que puede haber comportamientos no deseados. Por lo que se optó por no usarla.

Una vez el objeto entre en colisión con otro, los valores de fricción del material físico de Unity entran en acción. Específicamente la fricción dinámica. Dado que los dos objetos tienen un material físico con una fricción dinámica y una forma de combinarlos, se sigue un sistema de prioridad. Las formas de combinarlos, en orden de prioridad de mayor a menor, son el valor: promedio, mínimo, máximo y el resultante de su multiplicación.

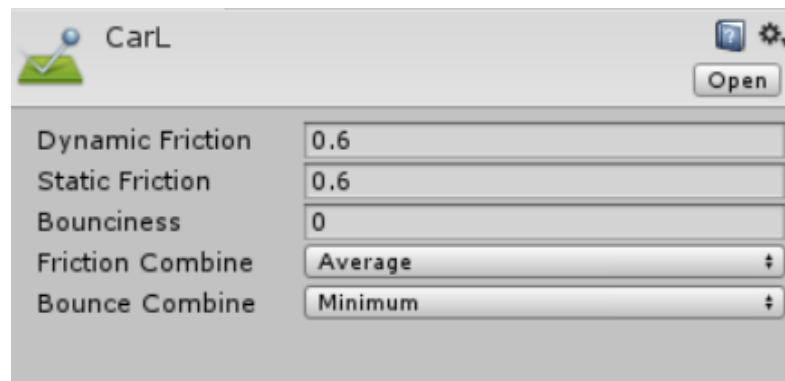


Figura 4: Imagen del material físico usado en los carros, en este caso del carro de la izquierda del usuario

La Figura 4 muestra los valores de un material físico, se puede observar que además de la fricción también tiene valores de elasticidad y como combinarla. La forma de combinarla es igual a la de la fricción, dado que el material de ambos carros tiene el mismo valor “0.6” no importa si el valor de combinación era el mínimo, el promedio o el máximo. El valor final de la fricción es 0 para que no afecte en el evento.

La elasticidad modifica el valor de rebote del carro. En donde el valor 0 hace que después del choque elásticos e inelásticos se queden juntos los carros, a este se le conoce como choque plástico, esto indica que la totalidad de la energía se perdió en el evento y es el valor que interactúa con el coeficiente de elasticidad.

5. Problemas con velocidad

Con la velocidad es posible ingresar valores tal que los carros choquen con el borde del mapa y reboten debido a su propio valor de elasticidad. Debido a esto se implementó un objeto en las paredes con un collider que detiene al carro al momento del impacto, de esta manera no se observa un comportamiento poco realista. Se usaron tags en las paredes para detener los carros únicamente cuando se colisionaba con el tag de las paredes.

6. Cálculo de fps

Unity no provee una forma de ver el frame rate por lo que se implementó una función para calcularlo. Esta usa la función de Unity “unscaledDeltaTime” la cual provee el intervalo en segundos desde el ultimo frame hasta el actual. Se diferencia de la función deltaTime en el sentido que no se ve afectada por la escala de tiempo (una escala de uno quiere decir que el tiempo en la aplicación pasa tan rápido como en la vida real, una escala de dos significa que pasa dos veces más rápido, etc).

7. Menú e interfaces

Para el menú se usaron varios canvas con botones, el canvas está en modo de renderizado de worldSpace. En este modo el canvas se comportará como cualquier objeto y los elementos (en este caso los botones) se renderizarán en frente o atrás de otros objetos basados en el posicionamiento en 3D. Como se puede observar en la Figura 5, los botones (esquina inferior izquierda donde están las tres flechas) están en frente de los carros desde el punto de vista de la cámara (esquina inferior izquierda, icono de cámara blanco).

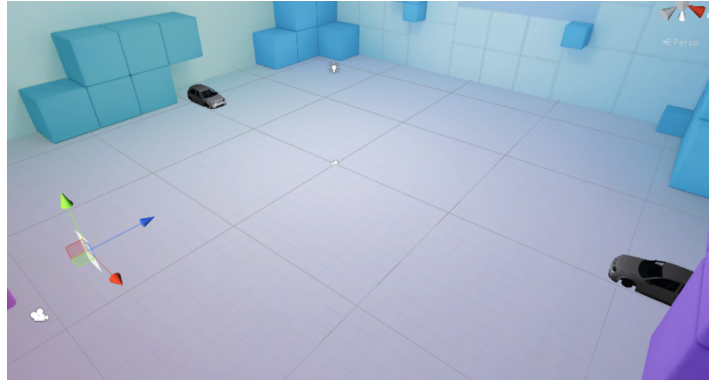


Figura 5: Imagen del posicionamiento del menu en la escena

Este tipo de renderizado también toma el nombre de “diegetic interface” debido a que la interfaz de usuario es parte del mundo, en vez de estar sobrepuesta en una pantalla, “non-diegetic interface”. El tipo de interfaz “non-diegetic” por lo general no funciona con vr porque nuestros ojos son incapaces de enfocarse en algo tan cerca.



Figura 6: Imagen del videojuego World of Warcraft con elementos “non-diegetic”

En la Figura 6, hay un ejemplo de una interfaz de usuario con elementos de “spatial-ui non-diegetic” y “non-diegetic”. Todos los cuadros rojos a excepción del que tiene la letra M, cerca del medio de la pantalla, son elementos “non-diegetic” están pegados a la pantalla y no importa cómo se mueva el jugador o que ocurra en el mundo estos no se moverá en relación a esto. El elemento M representa la vida de un personaje, esta barra roja se encuentra encima del personaje, lo que significa que conforme se acerque uno a este, rote y deje de verlo, todo esto tendrá efecto en el elemento.

La Figura 7 describe como clasificar un elemento de una interfaz. A pesar de que los

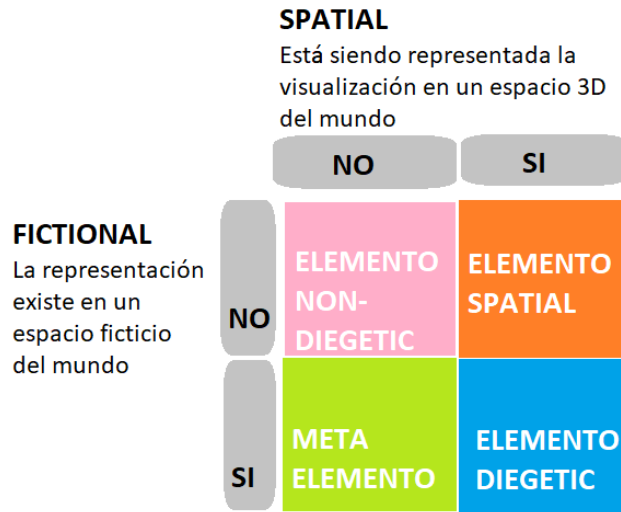


Figura 7: Imagen de la clasificación de interfaces

menús no forman parte de del mundo ficticio en los videojuegos, el personal de Unity lo considera como un elemento “diegetic” no tradicional.

8. Canvas

En los canvas también se usó un canvas scaler, este es usado para controlar la escala general del canvas y sus elementos, así como la densidad de pixeles. Este elemento en un canvas de renderizado de worldSpace provee una forma de obtener anti-aliasing sin mucho trabajo (desde un punto de vista del costo de renderizado en el rendimiento y desde otro punto: fácil implementación). Esto se obtiene usando un valor de uno en el campo de “Reference pixels per unit” y alterando el valor de “Dynamic Pixels per Unit” hasta obtener los bordes lisos deseados. Esto evita que el texto del menú se vea como la última palabra de la Figura 8.

En la Figura 9 se puede observar algunos pixeles con bordes cuadrados sobre todo en la letra O, en la primera palabra. El valor final en el campo “Dynamic Pixels Per Unit” en los canvas scaler quedó en 1.7.

Se agregó un canvas sobrevolando los carros como especie de marcador para presentar valores durante y al momento del choque. Este cambia de posición dependiendo si esta sucediendo el evento o si se esta viendo dese la visión de águila. Entre las variables que se presentan se encuentra el tiempo que dura el suceso. Para calcular el tiempo se guarda el tiempo en el momento de inicio del choque, en cada update (una vez cada frame) se obtiene



Figura 8: Ejemplo de aliasing en texto

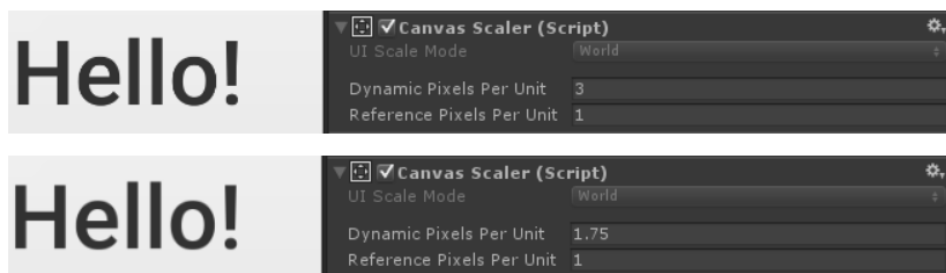


Figura 9: Ejemplo comparativo con diferentes valores de "Dynamic pixels per unit"

el tiempo y se suma, al momento del choque se guarda el tiempo final. Esto implica que si el fps (frames per second) de un celular es diferente al de otro o no es consistente entre cada corrida el tiempo no se podrá calcular exactamente.

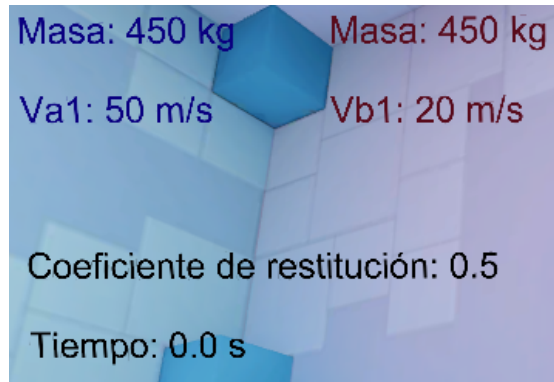


Figura 10: Imagen del canvas sobrevolando con variables por defecto

9. Colores de carros

En una versión inicial el menú incluía una opción para elegir el carro del que se quería cambiar las variables. Este paso se simplificó asociando colores a los carros y al menú, de tal manera que al elegir algo del botón azul se entiende que corresponde al carro azul. Esto también es de ayuda visual al usuario al momento de elegir las variables de los carros para saber a qué carro corresponde la variable que se está cambiando actualmente.

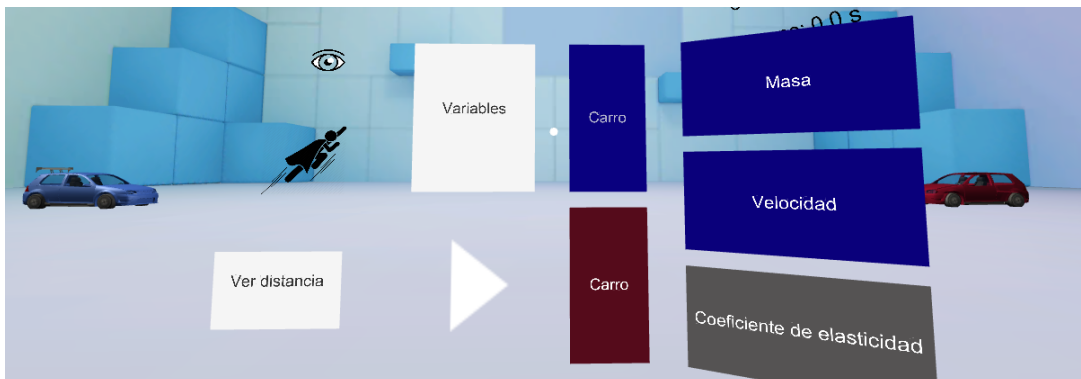


Figura 11: Imagen de los colores aplicados en carros y el menú

10. Retícula de selección

Al momento de ver los inputs con los usuarios se notó que no había forma de retroalimentar al usuario sobre la elección que hacía cuando miraba un elemento de los menús. Se implementó un sistema que basado en el tiempo que un usuario observa un elemento del menú, agranda la retícula de selección del usuario. Esto le permite al usuario sentirse en control y aunque este tipo de implementación limita que tan rápido un usuario puede escoger o seleccionar algo, se garantiza de que nadie elija algo por equivocación. Esto se puede observar en las figuras 12 y 13.



Figura 12: Imagen de la retícula de selección

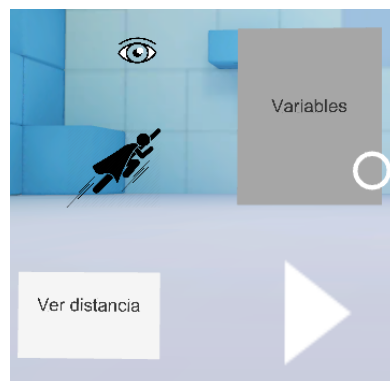


Figura 13: Imagen de la retícula de selección agrandada por causa de un elemento

11. Distancia

Es necesario presentar al usuario con una forma de determinar la distancia a la que se encuentran los objetos para poder entender el entorno mejor. Para lograr esto se utilizaron componentes de tipo “line renderer” para dibujar dos líneas que muestran la distancia. Se obtienen las posiciones de los dos carros, en el momento en el que se utiliza la función de cálculo de distancia, se calcula el tiempo que se desea demorar para dibujar y se utiliza la interpolación lineal para calcular el siguiente punto. Se utiliza la normalización de vector del vector final – vector inicial para obtener la dirección del vector y multiplicarlo con el siguiente punto sumándole luego el vector inicial.

$$size = lerp(0, distCarros, tiempo) \quad (2)$$

$$siguienteP = (normalizar(pFinal - pInicial) * size) + pInicial \quad (3)$$

El cálculo de la distancia entre los dos carros se apoya de la magnitud de vectores.

Sin embargo, un problema surgió de este método. Los puntos que se calculaban no estaban siendo calculados con referencia al padre, que no estaba sobre el centro del plano (0,0,0), sino en base al espacio de la escena por lo que la línea no era una línea recta sino tenía una desviación en el plano. Se hizo uso de la propiedad de local position para obtener los puntos de la línea.

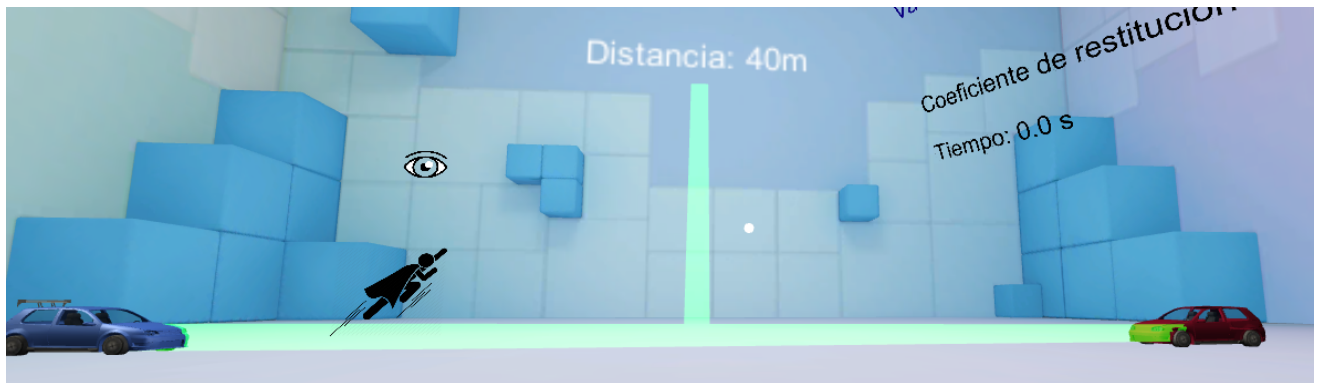


Figura 14: Imagen del cálculo de distancia entre carros

12. Movimiento

El usuario tiene la libertad de moverse en el mapa, con presionar el botón del Google Cardboard por 2 segundos y mantenerlo apachado. Esto en efecto mueve al personaje, así como al menú hasta la posición donde se detenga. Es posible moverse verticalmente. Esto implica quitar la visibilidad del menú cuando se está en movimiento y revertirla cuando se

detiene. Para mover al personaje se usó la propiedad de posición para determinar el lugar donde se debe estar luego de haber apachado el botón. Para mover el menú se anclaron los objetos correspondientes al jugador de tal manera que mover al jugador mueve a todos los objetos manteniendo la posición relativa de entre todos. Debido a configuraciones de la versión de Unity en uso y a la versión de Unity del asset del teclado, este no se podía anclar al jugador y moverlo. Por lo que se guarda la posición relativa del teclado con el jugador y al terminar de caminar se posiciona usando la propiedad de posición y estos valores.

Agregar movimiento al personaje implica que este puede toparse con los carros o ponerse entre los dos e interrumpir el choque. Esto no es un comportamiento deseado por lo que se tomó la medida preventiva de colocar paredes invisibles para restringir el área en el que se puede mover el personaje. Pero para permitir el movimiento vertical y dejar que el jugador tenga mas libertad de moverse se decidió partir el mapa en 2: una sección a nivel de los carros y una sección aérea llamada visión de águila. De esta manera el usuario puede rápidamente moverse entre una visión y otra y apreciar el choque de manera aérea.

13. Visibilidad

Para que el jugador pudiera explorar la escena sin la interrupción del menú se agregó un botón para esconderlo. Se esconde usando la propiedad de visibilidad.



Figura 15: Imagen usada en el boton de visibilidad

14. Vision aérea

Este cambio de funcionalidad impactaba la formaba en la que el usuario miraba los resultados finales por lo que se debió modificar la posición de este canvas para que el usuario lo pudiera ver fácilmente. Tanto al momento de configurar las variables como durante el evento del choque. La Figura 16 representa la imagen usada en el botón que activa esta funcionalidad. Se decidió usar una persona volando para que el usuario comprenda rápidamente que esto implica ver el choque hacia abajo y que también tiene movimiento vertical.



Figura 16: Imagen usada en el boton de vista aérea

15. Carros fantasma y guardar eventos

Para que el usuario pueda ver rápidamente varios eventos distintos sin tener que cambiar las variables de nuevo se agregó la funcionalidad de guardar hasta tres eventos extra (contando el ingresado se pueden tener cuatro eventos en cache y tres guardados en el celular). Una vez se guarde el evento y se de click sobre el boton respectivo de play carros “fantasma” se crearán en tiempo de corrida y tendrán las mismas propiedades y variables que los carros normales, con la excepción de que no tendrán un color sino son semi transparentes, dando un efecto fatasmal. Se puede ver la opción para guardar datos girando en sentido contrario de las agujas del reloj, Figura 17. En la Figura 18, se pueden ver los carros fantasma comparado con los carros normales.



Figura 17: Imagen del menú para guardar hasta tres datos



Figura 18: Imagen de carros fantasma y carros normales

Para guardar las variables se usaron las “sharedpreferences”, lugar donde se guardan los datos en Android. Se guarda en el celular en “/data/data/pkg-name/sharedprefs/pkg-name.xml” y se puede acceder a esto a través de código. Se usó un script hecho por la comunidad de Unity para guardar información en este lugar. Este se puede encontrar en la wiki de Unity buscando “PlayerPrefsX”.

Para lograr el efecto fantasmal de los carros se usó un shader propio aplicado a un material que se colocó sobre los carros. Además de poner tags en los carros e ignorar las colisiones con aquellos que no eran del mismo tag (fantasmas ignoran a normales y viceversa).

B. Investigación sobre aprendizaje

Uno de los objetivos del proyecto es investigar y describir como esta herramienta puede ser de ayuda para los estudiantes. Debido a esto la sección de aprendizaje tiene tantas citas bibliográficas. Se empezó con una búsqueda de las palabras “VR y learning” para ver artículos relacionados, sin embargo, muchas de las paginas encontradas no tenían citas y otras se contradecían. Por lo que se enfocó la investigación en el aprendizaje primero y la forma en la que se relaciona con vr luego.

El procedimiento se basaba en encontrar información interesante sobre el aprendizaje que pudiera relacionarse con la realidad virtual, buscar artículos científicos sobre esto y escoger los artículos más citados o más recientes. Luego se analizaba el artículo para ver sus resultados y como relacionarlo con realidad virtual.

A. Física Móvil

1. Aplicación

Se usaron cerca de 30 objetos padre en Unity, el conteo total de objetos es mucho más alto dado que el teclado virtual tiene un objeto por cada letra o número y los carros tienen múltiples objetos para representar las diferentes partes.

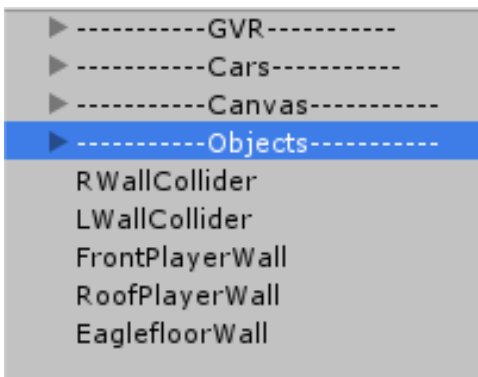


Figura 19: Imagen de objetos contenedor en Unity



Figura 20: Imagen de la escena en Unity

A continuación, se presentan una imagen de la aplicación física móvil tomada desde un celular Motorola G5 plus. Solo se muestra una debido a que se trata de realidad virtual el sistema divide la pantalla en dos para que el lente las pueda unir posteriormente.

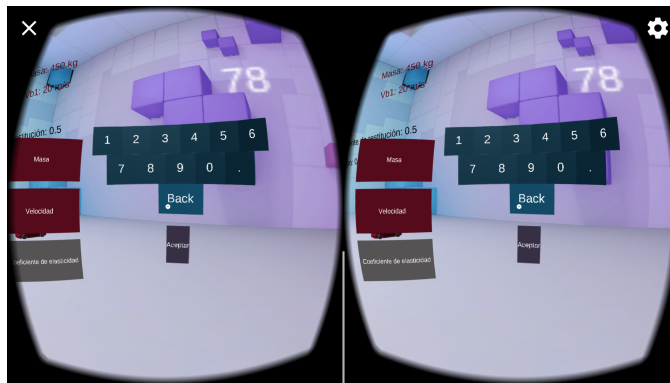


Figura 21: Imagen de la aplicación desde el celular

El usuario mira la Figura 22 nomás se abra la aplicación, estos botones funcionan como el menú. El usuario puede apuntar la retícula blanca circular en el centro de la pantalla hacia el botón que desee presionar.

El botón de variables nos lleva a la Figura 23, aquí se escoge que variables de que carro se quiere modificar. Al escoger el carro se muestran las variables como se aprecia en la figura 24, luego de escoger la variable a modificar el usuario puede observar el teclado virtual.

Una vez hayan apachado el botón de play se esconde el menú inicial y se muestra un botón para resetear el experimento, Figura 25

Una vez el usuario ingrese los valores deseados puede empezar el choque con el otro botón de la figura.

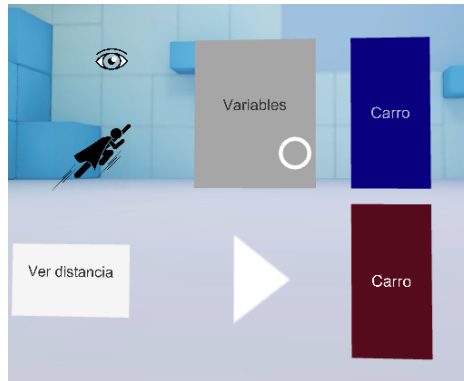


Figura 22: Imagen del menú inicial

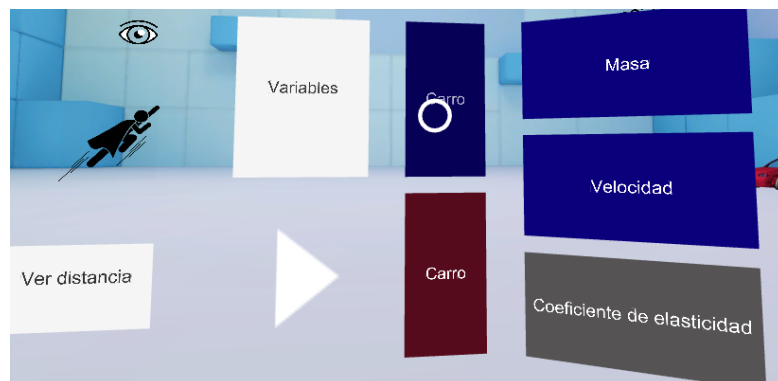


Figura 23: Imagen del menú de carros



Figura 24: Imagen de todo el menú



Figura 25: Imagen del reseteo de menú

Para cada carro se tienen las siguientes variables que se pueden modificar:

Variables	Como afecta los problemas
Masa	Este afecta que tan rápido se puede acelerar dado una fuerza. Y es parte de las ecuaciones de conservación de la energía
Velocidad	Sirve para poder aplicar una fuerza sobre el carro usando la fórmula 1. Y es parte de las ecuaciones de conservación de la energía
Coeficiente de elasticidad	Este define que tan elástico o inelástico será el choque. Permite valores entre 0 y 1.

Cuadro 1: Tabla de variables. Todas las variables posibles de ingresar en el evento y que lo afectan de una manera u otra

Estas variables son las que comúnmente se pueden observar y modificar en los ejemplos y problemas de choques. Puede haber otras como la resistencia del suelo, pero este necesita conceptos diferentes para entenderlo y usarlo.

Como se puede observar en las figuras el usuario tiene toda la disponibilidad de modificar y elegir las variables que desee en su experimento. De esta manera se cumple el objetivo: Programar una aplicación móvil, haciendo uso de Google Cardboard, con la que el usuario pueda interactuar y modificar para poder ejemplificar el problema de choques.

Al momento de iniciar el evento el marcador cambia de posición y muestra las variables ingresadas así como las variables al momento del choque. Adicional a las variables del Cuadro 2 se puede consultar la distancia cuya unidad esta en metros.

VARIABLES OBSERVABLES	UNIDAD
Masas	kilogramos
Velocidades iniciales, V_{a1}	Metros por segundo
Velocidades al momento del choque, V_{a2}	Metros por segundo
Coefficiente de elasticidad	Sin unidad
Tiempo del evento	Segundos

Cuadro 2: Tabla de variables en el marcador

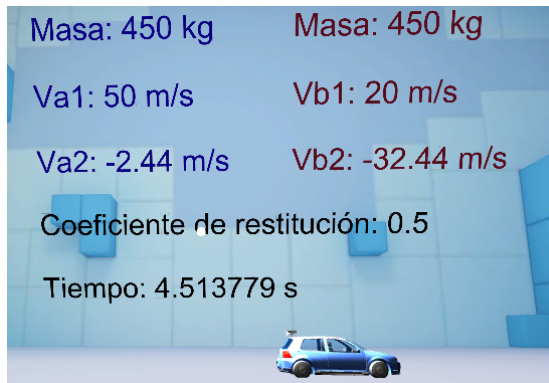


Figura 26: Imagen de los resultados al momento del choque

2. Pruebas con usuarios

Se realizaron pruebas y encuestas con personas de la UVG para saber que pensaban sobre la aplicación, su uso y potencial. Las siguientes gráficas mostrarán los resultados de estas.

Se puede observar de la Figura 27 que solo cuatro personas dijeron no tener un deseo de usar la aplicación. Esto muestra el gran interés de la población de tener una herramienta que facilite visualizar un experimento físico difícil de presenciar en vida real. Uno de los

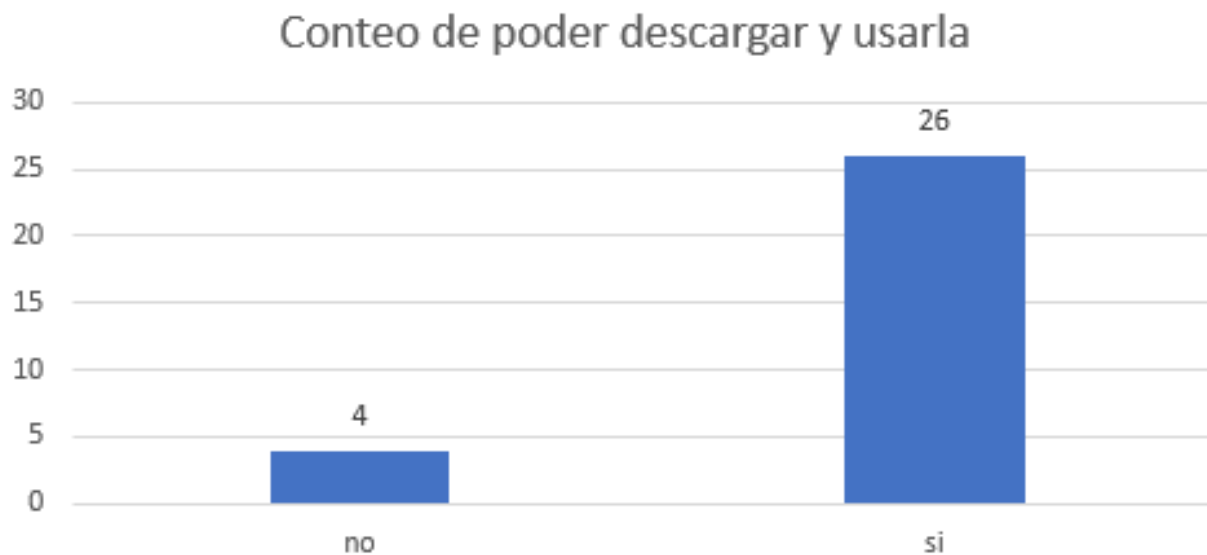


Figura 27: Gráfica que muestra el deseo de los usuarios de poder descargar y usar la aplicación

encuestados que respondió negativamete menciona “me gustó la solución al problema pero preferiría tenerlo en la computadora”.

Con la Figura 28 se puede ver que los usuarios encontraron el diseño gráfico de la aplicación “bueno” o mejor. Esto también se notaba al momento de entregar el equipo necesario para usar la aplicación. La mayoría de personas se mostraban sorprendidas, interesadas y curiosos de saber más sobre la aplicación, su uso y la realidad virtual. Durante el experimentó, en los que se podía observar, se pudo notar las sonrisas reflejando la buena experiencia que estaban teniendo con la aplicación.

B. Aprendizaje

1. Investigación

Listado de beneficios encontrados con el uso de VR en la educación y el aprendizaje.

1. Tener curiosidad, diversión y felicidad son de ayuda en el proceso de aprendizaje y los alumnos muestran curiosidad en cuanto a la realidad virtual en la educación. Los estudiantes responden positivamente hacia el uso de realidad virtual en la educación de esta manera relacionándolo con diversión y felicidad.

2. Cuando se está aprendiendo es importante poner atención en qué ambiente o contexto se está haciendo. Si se usa la realidad virtual en el ambiente de clase se puede mejorar el aprendizaje en este contexto.

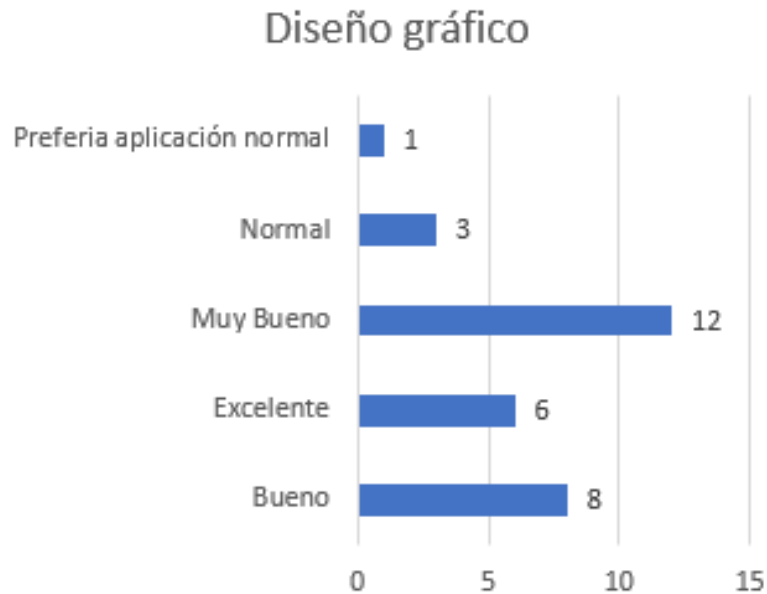


Figura 28: Gráfica sobre el diseño gráfico de la aplicación

3. La visualización es importante en el desarrollo del pensamiento crítico y este determina la calidad del aprendizaje por lo que una herramienta para visualizar, como lo es la realidad virtual, impacta la calidad del aprendizaje.

4. Crear una relación entre una experiencia y un concepto y sus relaciones disminuye el riesgo de olvidar otros conceptos al momento de recordar y la realidad virtual nos ayuda a crear relaciones con los conceptos.

5. La realidad virtual puede ayudar a las personas con necesidades especiales a aprender conceptos físicos.

2. Pruebas con usuarios

Se hicieron pruebas y encuestas con personas de la UVG para saber que pensaban sobre la el uso de la aplicación en la clase, su capacidad de enseñar y su opinión de la aplicación como ayuda en el salón. Las siguientes gráficas mostraran los resultados de estas.

Visualización previo al uso de la app

Se preguntó sobre la facilidad (escala de 1 a 10) de visualizar problemas físicos, visualizar problemas de choques y de entender la relación entre las variables. Los resultados se muestran en las figuras: 29, 30 y 31.

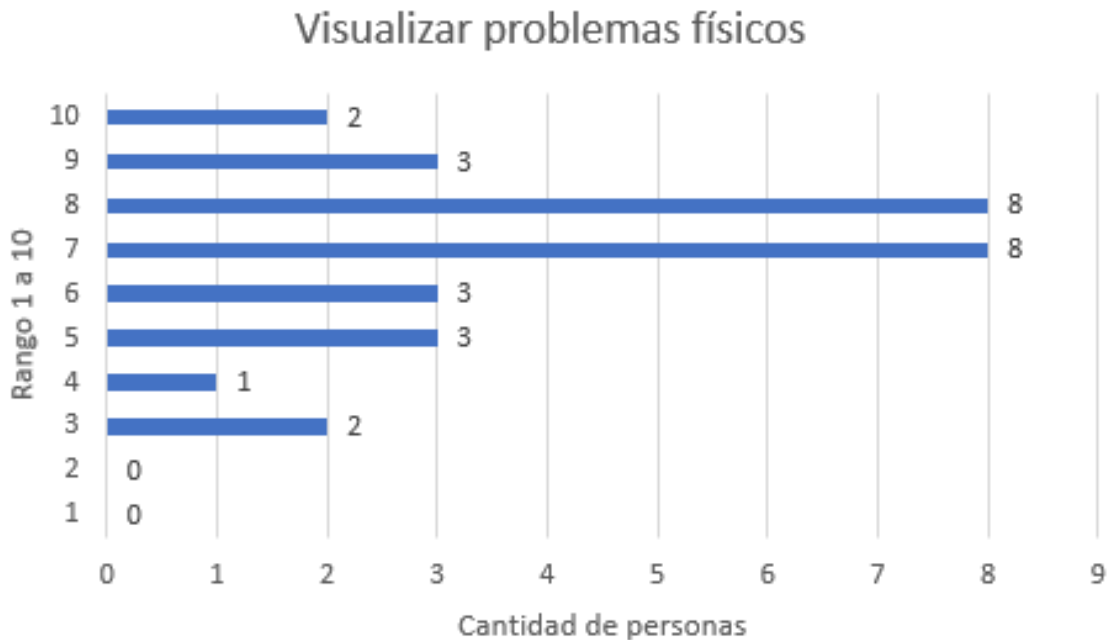


Figura 29: Gráfica sobre la facilidad de los usuarios de visualizar problemas físicos en una escala de 1 a 10

Es interesante ver en la Figura 29 como hay una cantidad de usuarios que esan confiados de sus habilidades para visualizar experimentos físicos (promedio de 7.53) pero en cuanto se pregunta sobre choques el promedio baja a 6.86. Inmediatamente se puede observar que es un tema que no todos entienden y que no lo entienden de la misma manera que otros temas. Mientras que el promedio de la facilidad de ver la relación entre variables es de 6.3, más bajo que la visualización de choques. Se puede entender entonces que aunque pueden imaginar mejor dos carros chocando no se entiende de la misma manera las variables que interactúan en el evento.

Preconocimientos

Se hicieron preguntas puntuales sobre el tema físico, no era de interés que pudieran resolver con valores, el interés era ver el conocimiento de los conceptos previo a usar la aplicación. Se evaluaron las respuestas con un puntéo de: 0, 25, 50, 75 o 100. Además de esto no se proporcionó ninguna ayuda ni se permitió que alguien o algo externo ayudara.

Un puntéo de 0 implica que no respondió o se equivocó por completo, obtener 25 implica que entendía los conceptos de manera inversa, obtener 50 implica que entendía alguno de los conceptos, obtener 75 implica que entendía la mayoría de los conceptos o que le faltó describir mejor la respuesta, obtener 100 implica haber respondido correctamente.

La primera pregunta consistía en que describiera que pasaría en un choque con los mismos

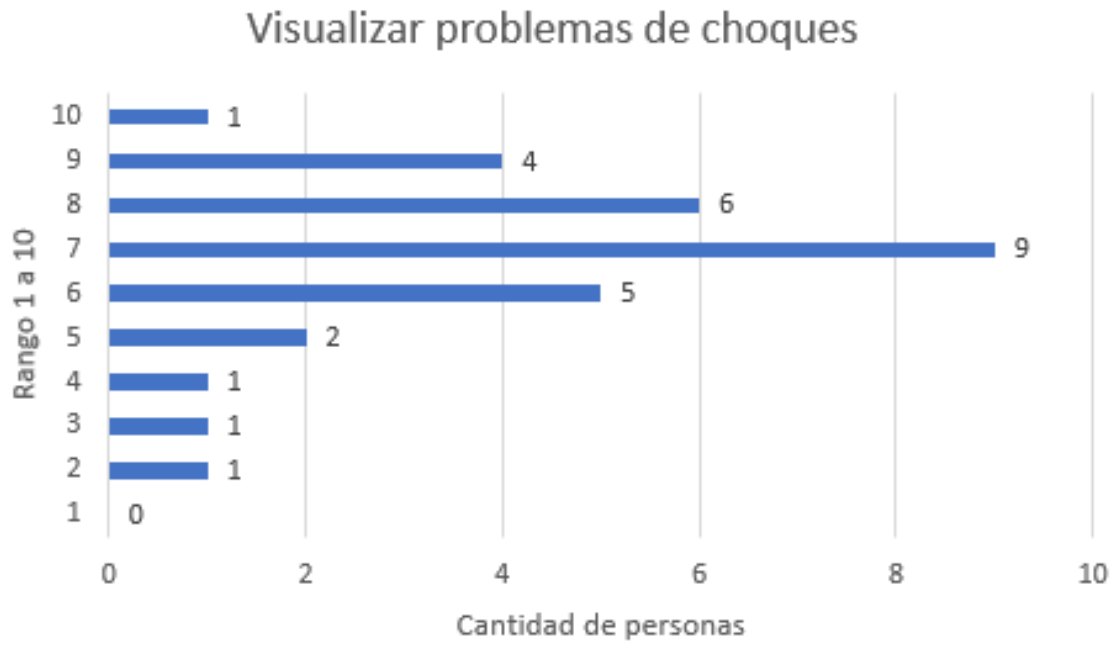


Figura 30: Gráfica sobre la facilidad de los usuarios de visualizar problemas de choque en una escala de 1 a 10

valores de los carros (masa y velocidad) cuando el evento es inelástico.

De la Figura 32 se puede ver que nueve personas no entendían para nada los conceptos y en total más del 40% de los encuestados perderían un examen de basado en esta pregunta.

La siguiente pregunta se basaba en responder la misma pregunta pero ahora la masa de uno de los carros es diferente.

Es interesante ver de la Figura 41 como a pesar de que hubieron menos personas con una nota de 100 se mantuvo la proporción de personas que perderían un examen. Se esperaba que también aumentara la cantidad de personas que sacaran 50 o menos, puede ser que hayan respondido al azar o que esta pregunta refrescara un poco su memoria. Sin embargo sigue siendo peligrosa la proporción.

Respuestas después de usar la app

Se volvieron a hacer las mismas preguntas que antes de haber usado la aplicación solamente agregando si consideraban que la aplicación les había ayudado de alguna manera. Se usaron los mismos rangos y criterios.

De las figuras 34, 35 y 36 se puede ver como rápidamente la aplicación ayuda en la visualización según los usuarios. El promedio en las tres preguntas aumentó casi un punto

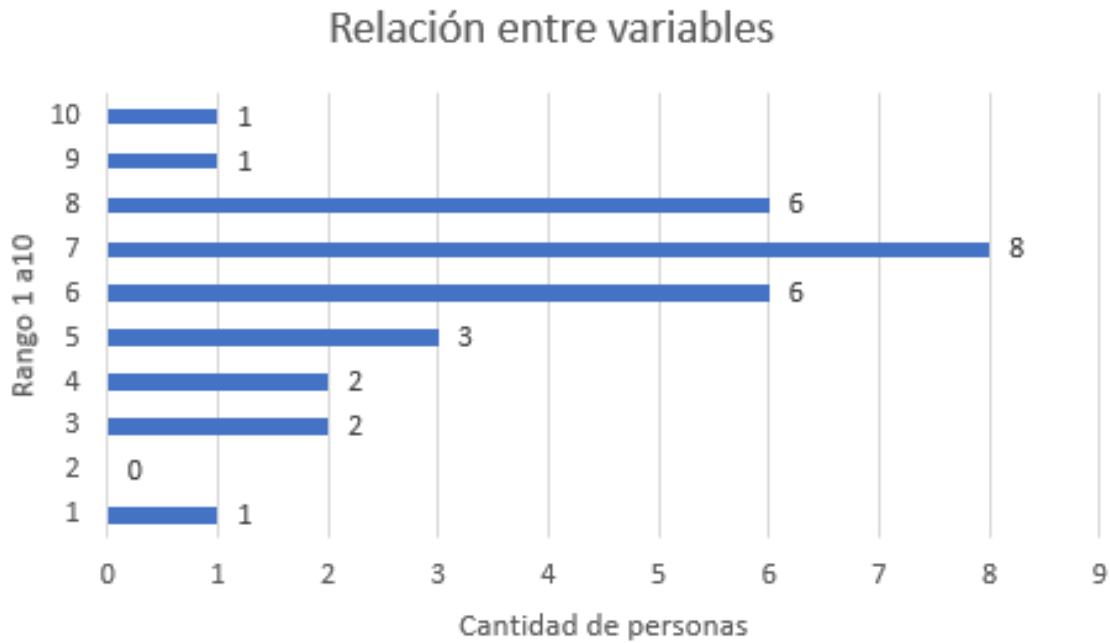


Figura 31: Gráfica sobre la facilidad de entender la relación entre variables de los choques en una escala de 1 a 10

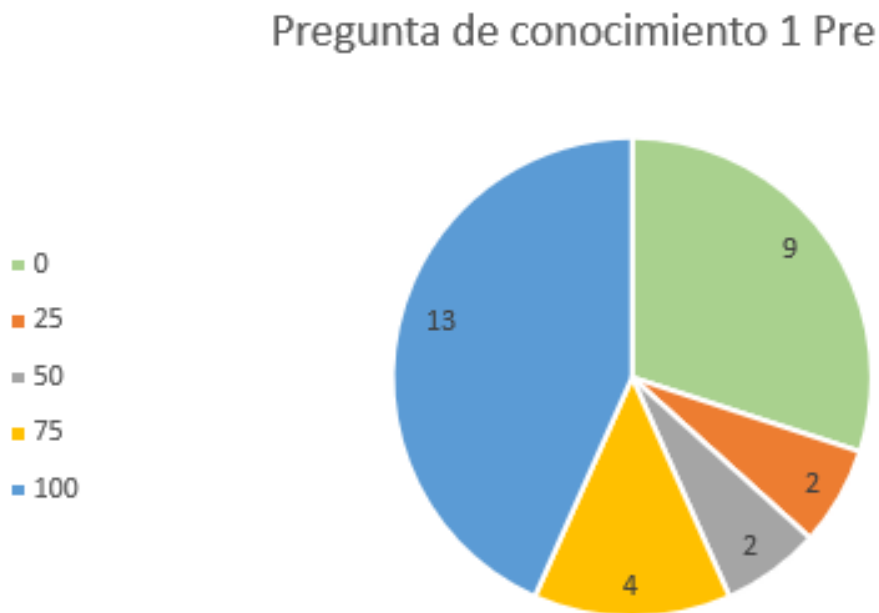


Figura 32: Gráfica sobre el conocimiento de los conceptos previo a usar la app

Pregunta de conocimiento 2 Pre

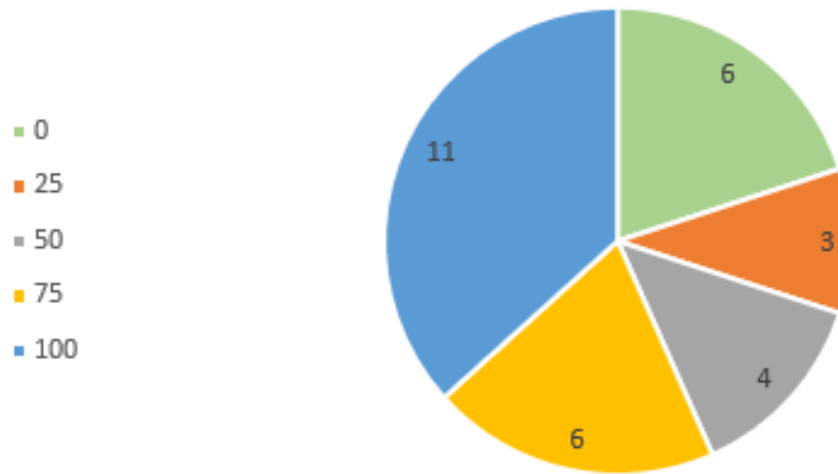


Figura 33: Segunda gráfica sobre el conocimiento de los conceptos previo a usar la app

Visualizar problemas físicos

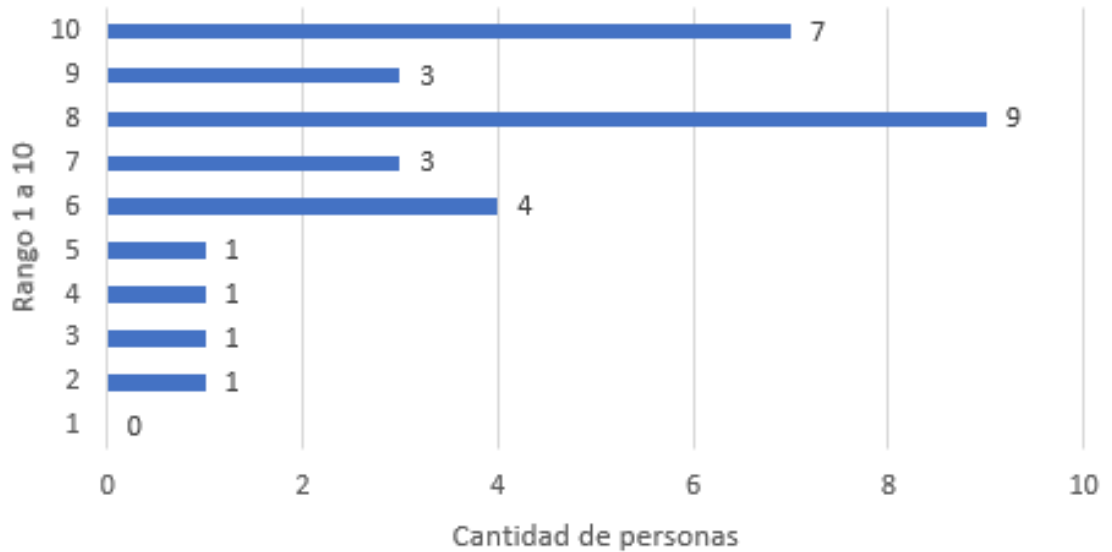


Figura 34: Gráfica sobre la facilidad de los usuarios de visualizar problemas físicos después de usar la app en una escala de 1 a 10

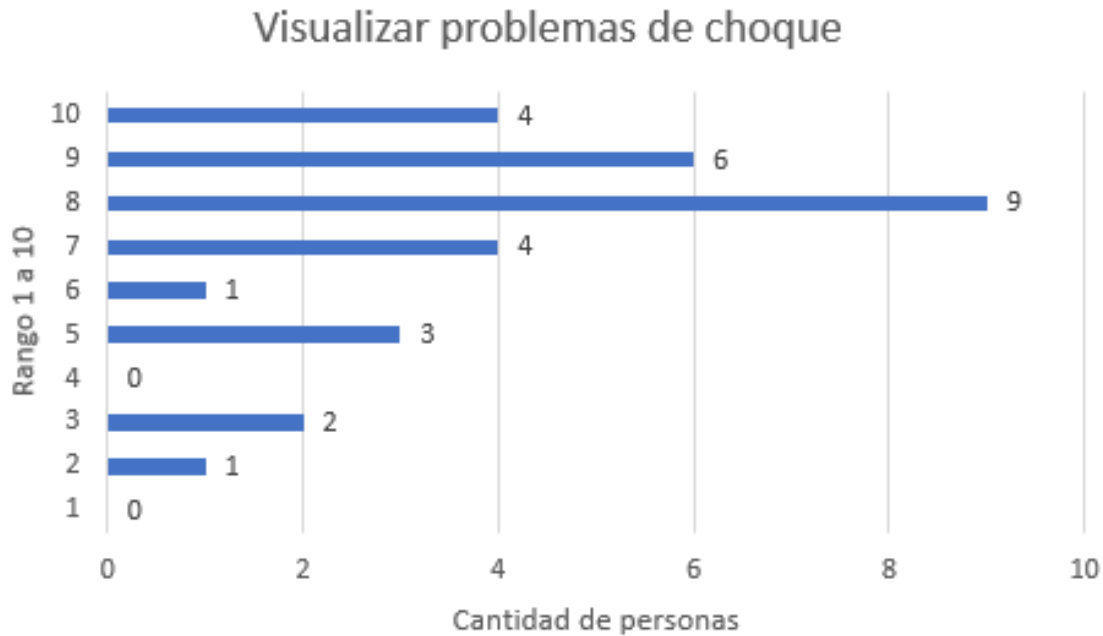


Figura 35: Gráfica sobre la facilidad de los usuarios de visualizar problemas de choque después de usar la app en una escala de 1 a 10

en la relación entre variables (6.3 a 7.13) y la visualización de problemas de choque (6.86 a 7.53) aunque solamente aumentó de 7.53 a 7.60 en la visualización de problemas físicos. Pero esto acerca mucho más el promedio de la visualización de problemas físicos y de choque (7.60 y 7.53).

Además se pidió responder si les había sido de ayuda la aplicación para estos casos.

De esta gráfica se entiende que el 80% de los encuestados encontraron de ayuda la aplicación para visualizar problemas y entenderlos. Lo cual también lo respaldan todas las gráficas post uso de la aplicación.

Respuestas a preguntas de física post uso app

De igual manera se volvieron a hacer las mismas preguntas sobre problemas físicos que antes de haber usado la aplicación solamente agregando si consideraban que la aplicación les había ayudado de alguna manera. Se usaron los mismos rangos y criterios.

Es fácil notar la enorme diferencia de la gráfica antes y después. En este caso 77% de los encuestados ganarían el examen.

Se puede comparar con las siguientes dos figuras.

Se observa muchas menos ondulaciones y es más constante después de usar la aplicación.

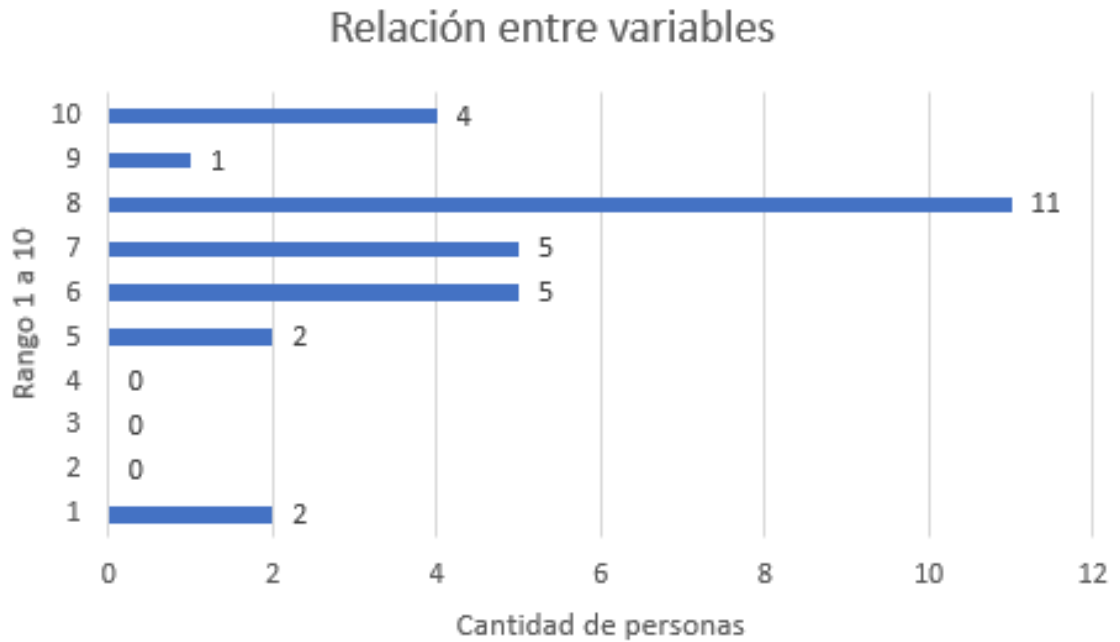


Figura 36: Gráfica sobre la facilidad de los usuarios de la relación entre variables de los choques después de usar la app en una escala de 1 a 10

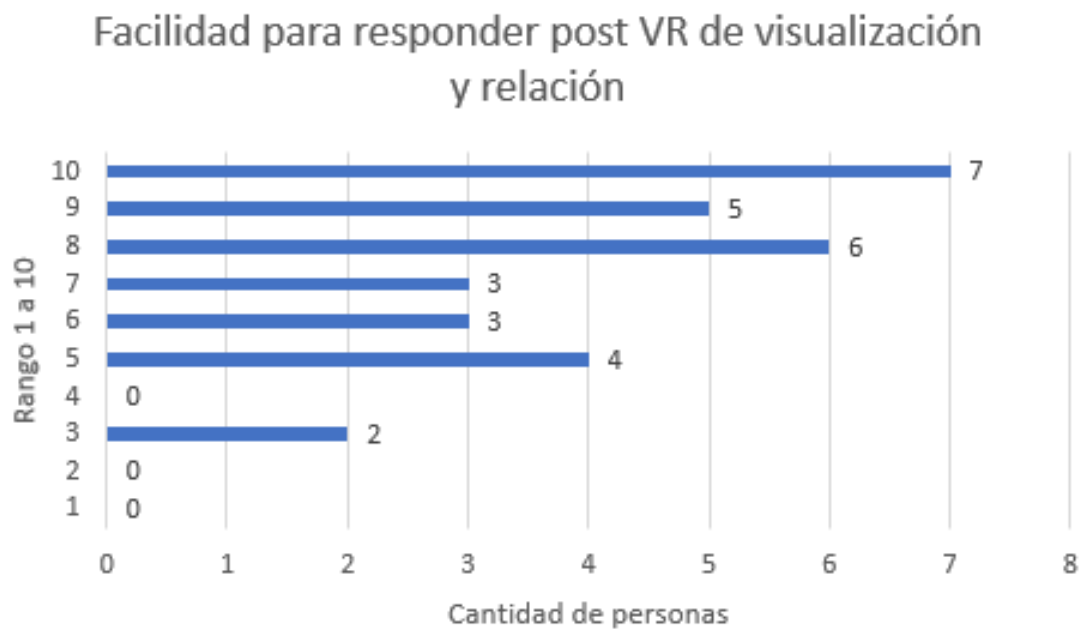


Figura 37: Gráfica sobre la ayuda de la app para visualizar y entender choques después de usarla en una escala de 1 a 10

Pregunta de conocimiento 1 Post

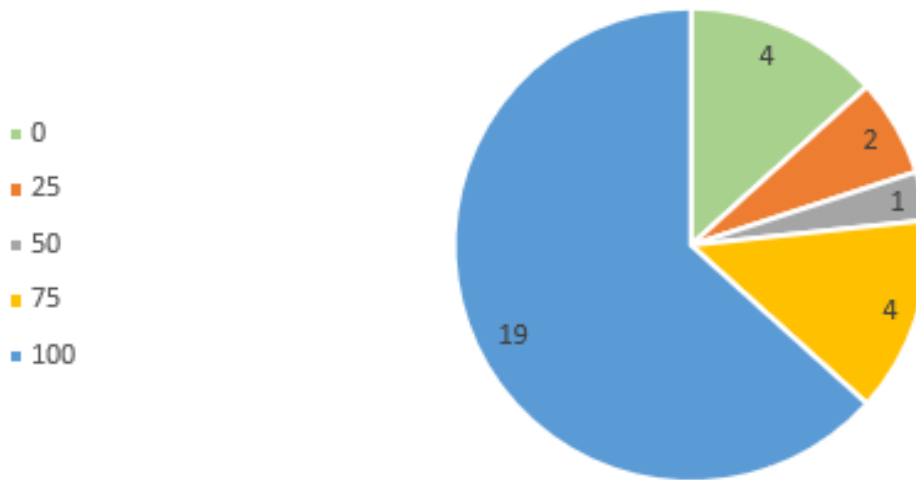


Figura 38: Gráfica sobre el conocimiento de los conceptos después de usar la app

Pregunta de conocimiento 1 Pre

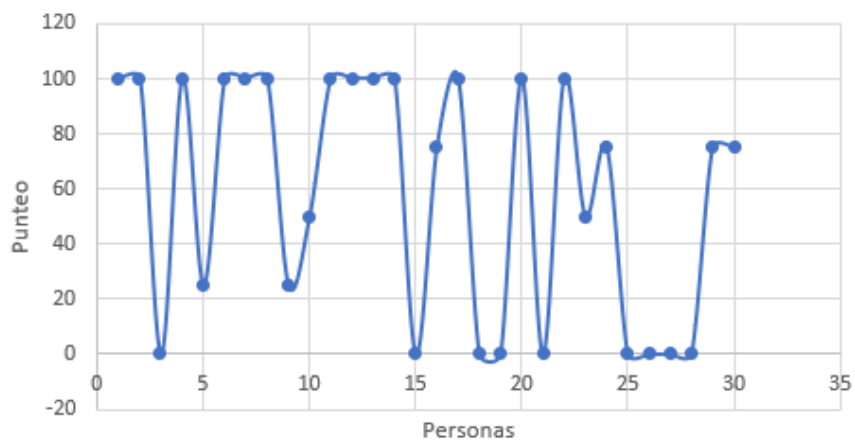


Figura 39: Gráfica sobre el conocimiento de los conceptos previo a usar la app

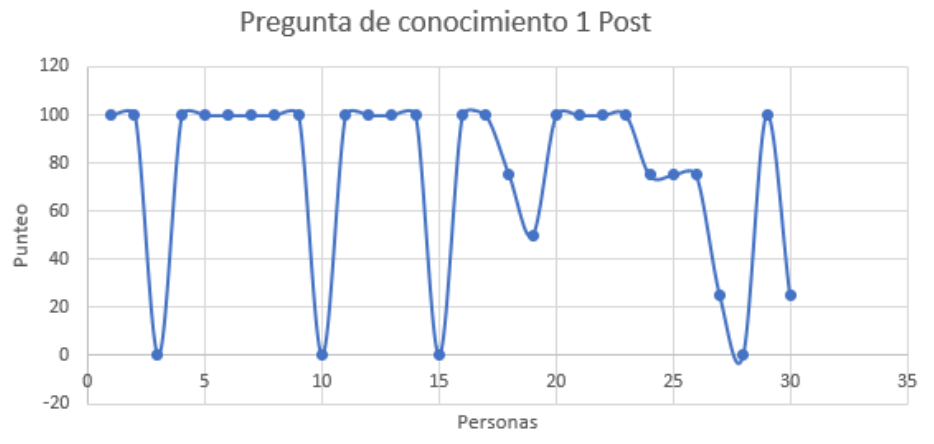


Figura 40: Gráfica sobre el conocimiento de los conceptos después de usar la app

Además de haberse reducido sustancialmente la cantidad de 0 de 9 a 4. Se puede ver también como los individuos mejoraron su desempeño, pero algunos se mantuvieron casi igual por ejemplo los participantes 3, 15, 24 y 27 se mantuvieron igual.

Para la segunda pregunta se observó un comportamiento similar.

Disminuyó la cantidad de personas que perderían el examen a solo 20% para comparar se puede observar las figuras 42 y 43.

Se nota la diferencia de resultados entre ambas gráficas al igual que en la primera pregunta. Y de igual manera se pueden observar individuos que se mantuvieron igual entre ambos casos (pre y post uso). Por ejemplo los individuos 11, 16, 24, 29 y 30 se mantuvieron con el mismo puntaje. Esto puede implicar que no todos aprenden tanto de manera visual o no estaban poniendo atención al experimento.

Cuando se preguntó si la aplicación había sido de ayuda solo el 17% respondió que no, (Figura 44) con 10 personas respondiendo “Mucho más fácil”

Sobre el uso en las clases

Se les preguntó también si considerarían útil la aplicación a la hora de resolver problemas de choques.

Solo una persona no lo consideraría de ayuda al momento de resolver sus problemas de choque.

Adicionalmente el 100% de los encuestados encuentra el beneficio de usar la aplicación en el salón de clase.

Pregunta de conocimiento 2 Post

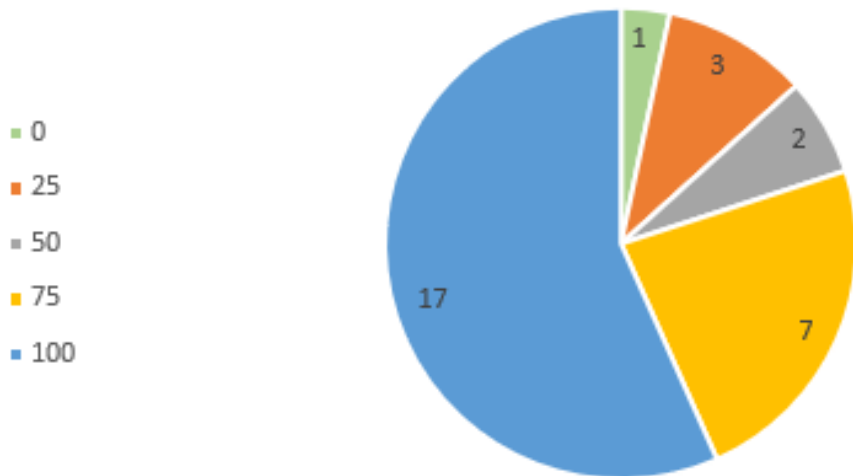


Figura 41: Segunda gráfica sobre el conocimiento de los conceptos después de usar la app

Pregunta de conocimiento 2 Pre

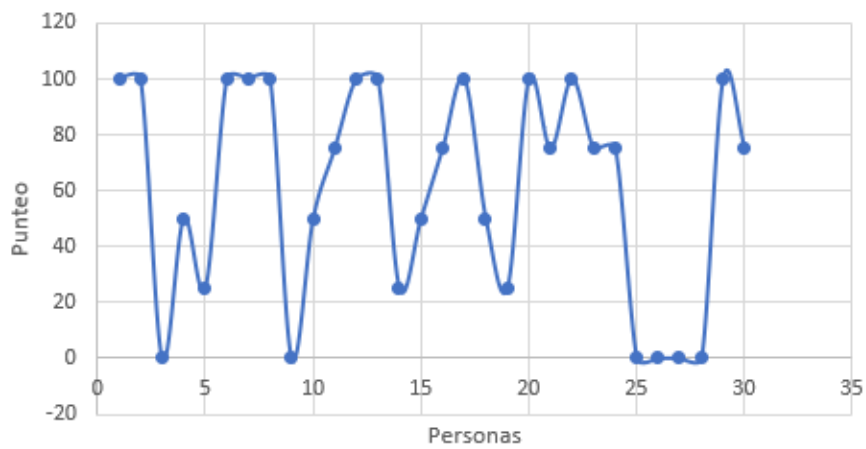


Figura 42: Segunda gráfica sobre el conocimiento de los conceptos previo a usar la app

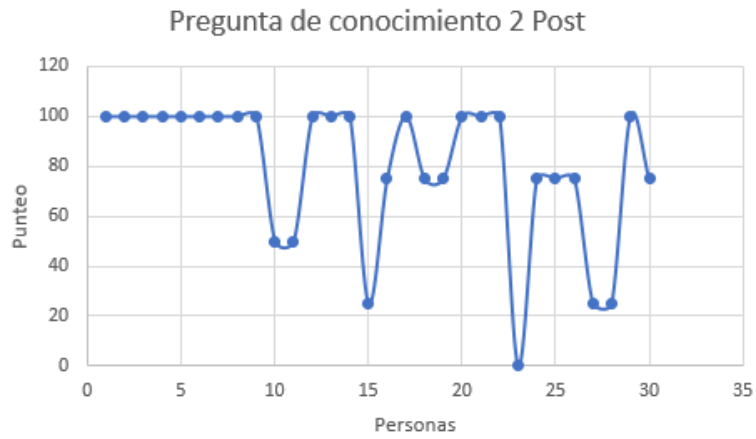


Figura 43: Segunda gráfica sobre el conocimiento de los conceptos después de usar la app

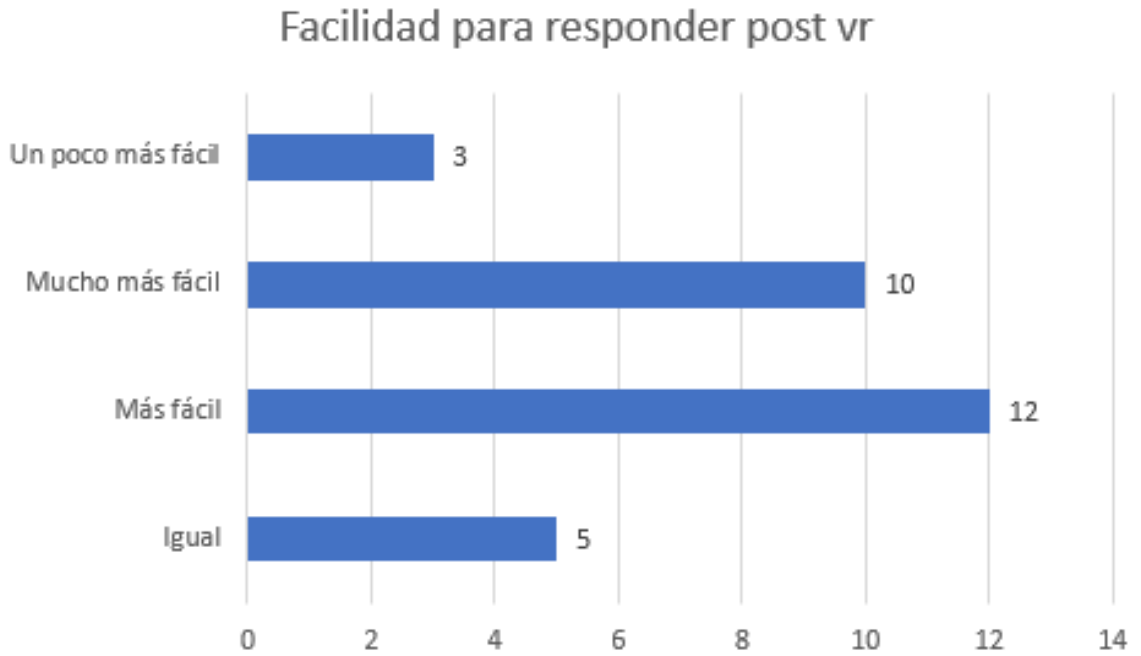


Figura 44: Gráfica sobre la ayuda de la app para responder problemas de choques después de usarla en una escala de 1 a 10

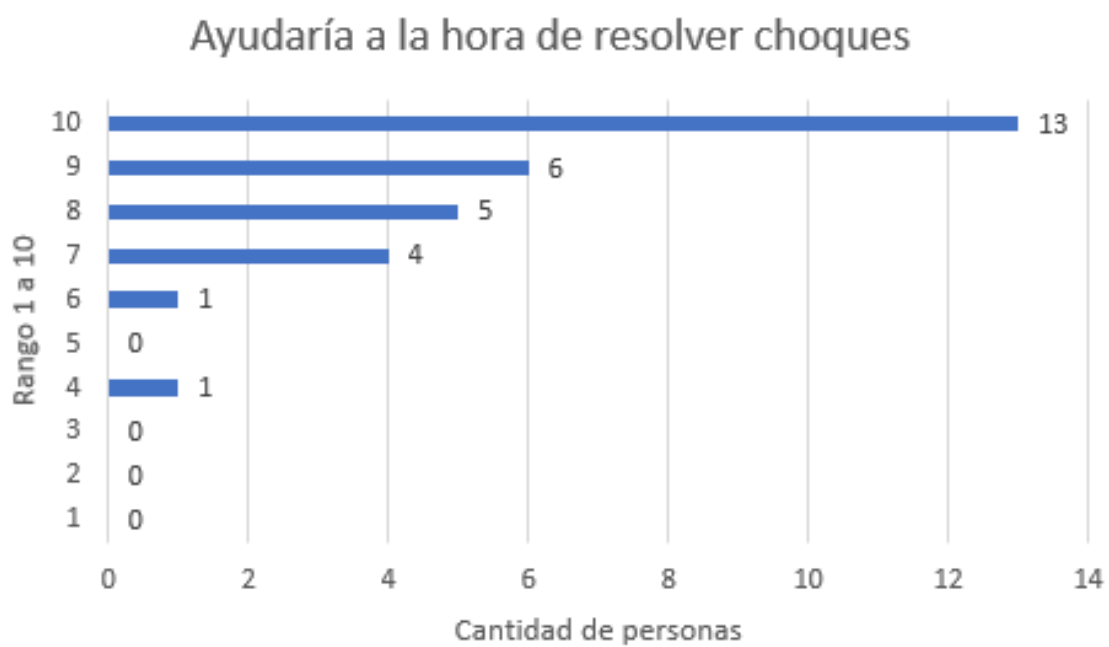


Figura 45: Gráfica sobre la ayuda al resolver problemas de choque en una escala de 1 a 10

- Tener curiosidad, diversión y felicidad son de ayuda en el proceso de aprendizaje.
- Los estudiantes responden positivamente hacia el uso de realidad virtual en la educación.
- Cuando se está aprendiendo es importante poner atención en qué ambiente o contexto se está haciendo.
- La visualización es importante en el desarrollo del pensamiento crítico y este determina la calidad del aprendizaje.
- Crear una relación entre una experiencia y un concepto y sus relaciones disminuye el riesgo de olvidar otros conceptos al momento de recordar.
- La realidad virtual puede ayudar a las personas con necesidades especiales a aprender conceptos físicos.
- El uso de una realidad virtual en el salón de clase tiene muchos beneficios en el proceso de aprendizaje.
- Los estudiantes desean poder descargar y usar la aplicación creada.
- A los usuarios les gustó el diseño gráfico de la aplicación.
- El problema de choques fue más difícil de visualizar que otros problemas físicos por 0.67 puntos.
- 40 % de los encuestados no tenían conocimientos suficientes del tema de choques.
- La distribución de notas sobre las preguntas de conocimiento previo se mantuvo.
- La aplicación ayudó en la visualización de problemas.
- Después de usar la aplicación la diferencia de visualizar problemas de choque y otros problemas físicos disminuyó a solamente 0.07 puntos.

- Al 80% de encuestados les ayudó la aplicación para visualizar problemas de física.
- Hubo una diferencia de 33% de personas que mejoraron su desempeño en la primera pregunta de los problemas después de usar la app.
- Algunos usuarios obtuvieron la misma nota antes y después de usar la aplicación.
- En la segunda pregunta hubo una mejoría de 20% al responder.
- Al 83% de los encuestados les sirvió la aplicación para responder preguntas del problema físico.
- A más del 90% de usuarios les ayudaría tener la aplicación a para resolver problemas de choques.
- Todos los encuestados creen que les sería de ayuda tener la aplicación.

Recomendaciones

Debido al hecho que es más fácil recordar algo cuando hay un tipo de bonificación en las preguntas se recomienda ampliar sobre el tema de “gamification” en la aplicación. Punteos y recompensas basado en el desempeño del usuario lo ayudará a recordar y mejorar el aprendizaje de los temas.

Se recomienda hacer más pruebas con el movimiento de la aplicación ya que al menos un usuario reportó un ligero mareo solo moviendose para ver el menú.

Se recomienda hacer pruebas de HCI para mejorar la interacción del usuario con la aplicación y que sea más amigable para el mismo. Las pruebas deberían realizarse con una población muestra del usuario final. Específicamente para el menú.

Se recomienda hacer pruebas con sonido dentro del salon de clase para ver su efectividad y si es muy notable con varios usuarios.

Dado que el contexto en el que se aprende impacta la memoria se recomienda relacionar el salón de clase con el mundo virtual.

Si el nuevo sistema de trabajos de Unity está implementado, se recomienda probar usarlo para los raycasts, esto podría mejorar la duración de la batería de un celular y en celulares viejos podría mejorar el frame rate.

A partir de las encuestas realizadas (Figura 46, anexos) se pueden ver algunas oportunidades de mejora mencionadas por los usuarios. Es necesario resaltar lo siguiente: al menos 7 personas mencionaron el ingreso de variables como una oportunidad (movimiento de cabeza, mareo, cambiar variables). Los resultados de borroso se descubrió posteriormente que al momento del uso, la pantalla del Google Cardboard se había movido ligeramente por lo que el lente no enfocaba bien, dando el efecto de “borroso”.

Otros resultados interesantes fueron sobre los problemas al usar la aplicación (Figura 47, anexos). 1 persona tuvo problemas con la orientación de la escena (hacia donde ver) esto se debe a que parece aleatorio el posicionamiento respecto a la persona de la escena. Valdría la

pena investigar más sobre esto. Otra persona tuvo problemas viendo los fantasmas, podría probarse con otros colores o quizás con un tamaño mas grande de los carros.

Durante las encuestas también se encontró el deseo de toda la población de encuestados de incluir otros problemas en la aplicación.

-
- [1] A. M. Hoover, S. Burden, X.-Y. Fu, S. S. Sastry y R. S. Fearing, “Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot”, en *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, IEEE, 2010, págs. 869-876.
 - [2] M. C. Anderson, R. A. Bjork y E. L. Bjork, “Remembering can cause forgetting: retrieval dynamics in long-term memory”, *Experimental Psychology: Learning, Memory, and Cognition*, 1994.
 - [3] K.-H. Bäuml y C. Kuhbandner, “Remembering Can Cause Forgetting—but Not in Negative Moods”, *Psychological Science*, vol. 18, págs. 111-115, 2 2007.
 - [4] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill y D. R. Krathwohl, *Taxonomy of educational objectives, handbook I: The cognitive domain*. New York, United States of America: David McKay Co Inc, 1956, vol. 19, pág. 56.
 - [5] K. Bowman, *The value of ACE: A guide to the evidence base*. Canberra, Australia: Adult Learning, 2006.
 - [6] K. Bowman y P. Kearns, *E learning for the mature age worker: Case studies*. Canberra, Australia: Australian Flexible Learning Framework, DEST, 2007.
 - [7] N. Burgess, W. E. Hockley y K. L. Hourihan, “The effects of context in item-based directed forgetting: Evidence for” one-shot” context storage”, *Memory & cognition*, vol. 45, págs. 745-754, 2017.
 - [8] A. Clemens, R. Hartley y H. Macrae, *ACE outcomes*. Adelaide, Australia: NCVER, 2003.
 - [9] A. Davis, “The impact of aging on education, Adult Learning and Development Continuing Education Division”, 2001.
 - [10] EcuRed, *Unity3D*. dirección: <https://www.ecured.cu/Unity3D>.
 - [11] Fisicalab.com, *Fisicalab / Web de Física*, 2018. dirección: <https://www.fisicalab.com>.
 - [12] Gamecareerguide.com, *What is a Game Engine?*, 2018. dirección: https://www.gamecareerguide.com/features/529/what_is_a_game_.php.

- [13] M. J. Gruber, B. D. Gelman y C. Ranganath, “States of Curiosity Modulate Hippocampus-Dependent Learning via the Dopaminergic Circuit. Neuron”, *Neuron*, vol. 84, págs. 486-496, 2 2014. DOI: <https://doi.org/10.1016/j.neuron.2014.08.060>.
- [14] Helpx.adobe.com, *Aliasing & Anti-aliasing*, 2016. dirección: <https://helpx.adobe.com/photoshop-elements/key-concepts/aliasing-anti-aliasing.html>.
- [15] J. Hurtado de Barrera, R. Rojas M y H. Blanco, *Cómo formular objetivos de investigación*, 2.^a ed. Caracas, Venezuela: Sypal, 2008.
- [16] K. H. James, S. Vinci-Booher y F. Munoz-Rubke, “The impact of multimodal-multisensory learning on human performance and brain activation patterns”, en *Handbook of Multimodal-Multisensor Interfaces*. Association for Computing Machinery y Morgan & Claypool, 2017, págs. 51-94.
- [17] T. L. Jeffs, “Virtual Reality and Special Needs”, *Themes in science and technology education*, 2009.
- [18] M. J. Kang y et al, “The Wick in the Candle of Learning: Epistemic Curiosity Activates Reward Circuitry and Enhances Memory”, *Psychological Science*, 2008.
- [19] D. R. Krathwohl, B. S. Bloom y B. B. Masia, *Handbook II: affective domain*. New York, United States of America: David McKay, 1964.
- [20] R. E. Landrum y R. A. Gurung, “The memorability of introductory psychology revisited”, *Teaching of Psychology*, vol. 40, págs. 222-227, 3 2013. DOI: 10.1177/0098628313487417top.sagepub.com.
- [21] K. Lightfoot y E. M. Brady, “Transformations Through Teaching and Learning”, *Transformative Education*, vol. 3, págs. 221-235, 3 2005.
- [22] J. Linowes, *Unity virtual reality projects*. Birmingham, England: Packt Publishing, 2015.
- [23] D. Lucardie, “The Impact of Fun and Enjoyment on Adult’s Learning”, *Procedia - Social and Behavioral Sciences*, vol. 142, págs. 439-446, 2014.
- [24] T. Mikropoulos y et al, “Students’ attitudes towards educational virtual environments”, *Education and Information Technologies*, vol. 3, págs. 137-148, 2 1998.
- [25] Newtonvr.com, *Physics-based interactions in VR made possible and easy*, 2015. dirección: <http://www.newtonvr.com>.
- [26] Oculus.com, <http://www.newtonvr.com>, 2018. dirección: <https://www.oculus.com>.
- [27] V. S. Pantelidis, “Reasons to Use Virtual Reality in Education and Training Courses and a Model to Determine When to Use Virtual Reality”, *Themes in science and technology education*, 2009.
- [28] J. Penman y B. Ellis, “Regional academics’ perceptions of the love of learning and its importance for their students”, *Australian Journal of Adult Learning*, 2009.
- [29] *Física: Rotación de un Cuerpo Rígido*, 2018. dirección: http://www.astro.puc.cl/~avalcarc/FIS109A/15_Rotacion_Solido_Rigido.pdf.
- [30] Pwc.com, *Global Entertainment & Media Outlook 2018-2022*, 2018. dirección: <https://www.pwc.com/gx/en/industries/tmt/media/outlook.html>.
- [31] M. G. Rhodes, A. E. Witherby, A. D. Castel y K. Murayama, “Explaining the forgetting bias effect on value judgments: The influence of memory for a past test”, *Memory & cognition*, 2017.

- [32] A. Stonehouse, *User interface design in video games*, 2011. dirección: <http://yuting-jiang.blogspot.com/2011/11/user-interface-design-in-video-games.html>.
- [33] U. Technologies, *Unity - Manual: Asset Workflow*, 2018. dirección: <https://docs.unity3d.com/Manual/AssetWorkflow.html>.
- [34] Unity, *Interaction in VR - Unity*, 2015. dirección: <https://unity3d.com/learn/tutorials/topics/virtual-reality/interaction-vr>.
- [35] —, *Unity*. dirección: <https://unity3d.com/>.
- [36] —, *User Interfaces for VR - Unity*, 2015. dirección: <https://unity3d.com/learn/tutorials/topics/virtual-reality/user-interfaces-vr>.
- [37] S. Urbina, *Essentials of psychological testing*, 2.^a ed. Hoboken, New Jersey: Wiley, 2004.
- [38] Google, *Google Cardboard - Google VR*. dirección: <https://vr.google.com/cardboard/>.
- [39] —, *Google VR*. dirección: <https://vr.google.com>.
- [40] S. Zerbst y O. Duevel, *3D Game Engine Programming*. Boston, Massachusetts: Course Technology Crisp, 2004.

A. Datos extras encuestas

1. Oportunidades de mejora



Figura 46: Gráfica de las oportunidades de mejora de la aplicación

2. Problemas usando la aplicación



Figura 47: Gráfica de problemas encontrados en la aplicación

3. Distribución de carreras

Distribución de carreras

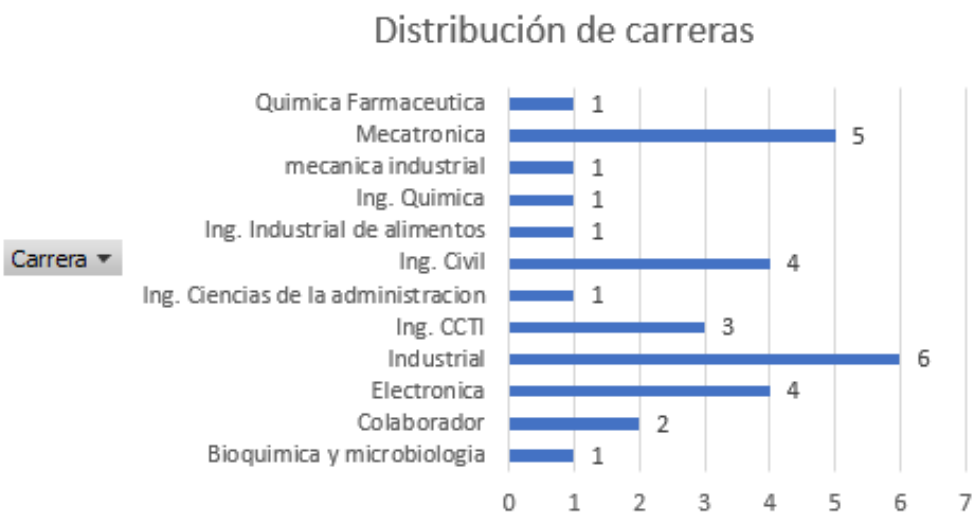


Figura 48: Gráfica de la distribución de carreras de encuestados

4. Uso previo de vr

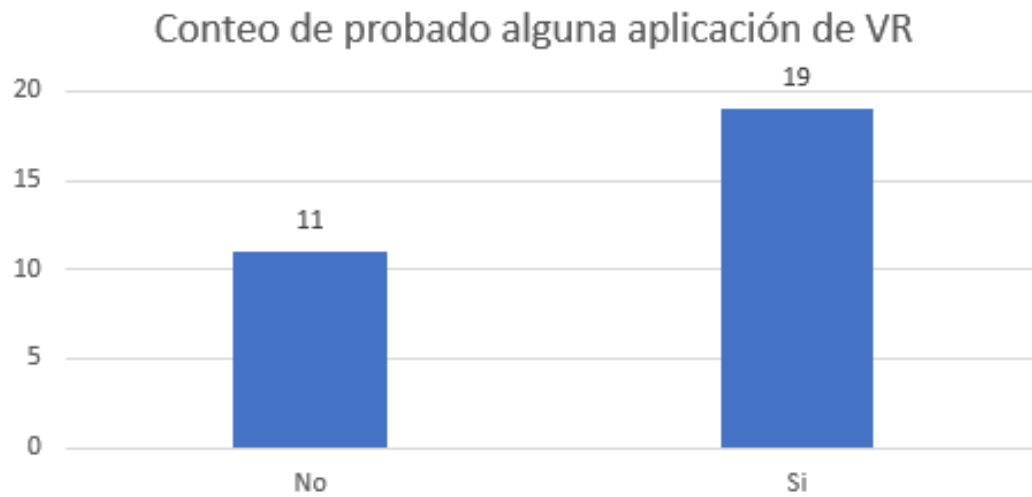


Figura 49: Gráfica del uso previo de aplicaciones de vr de encuestados

5. Sexo

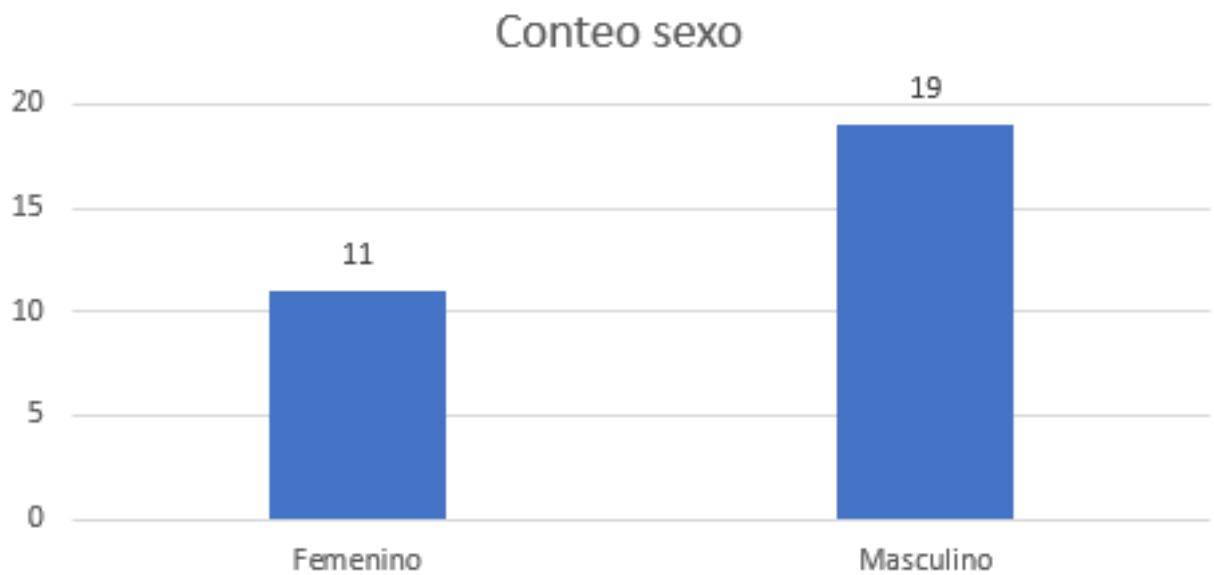


Figura 50: Gráfica del sexo de encuestados

6. Cantidad de información en la app

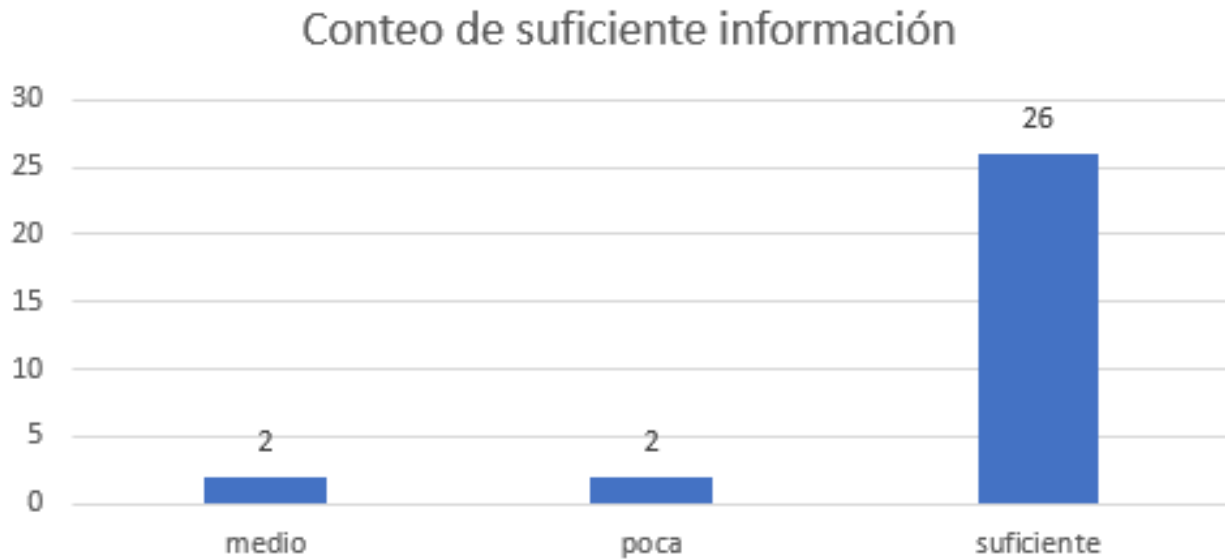


Figura 51: Gráfica sobre la cantidad de información en la app

7. Dificultad para usar aplicación



Figura 52: Gráfica de cuánta ayuda necesitaron los encuestados durante el uso de la aplicación

8. Lo mejor de la aplicación

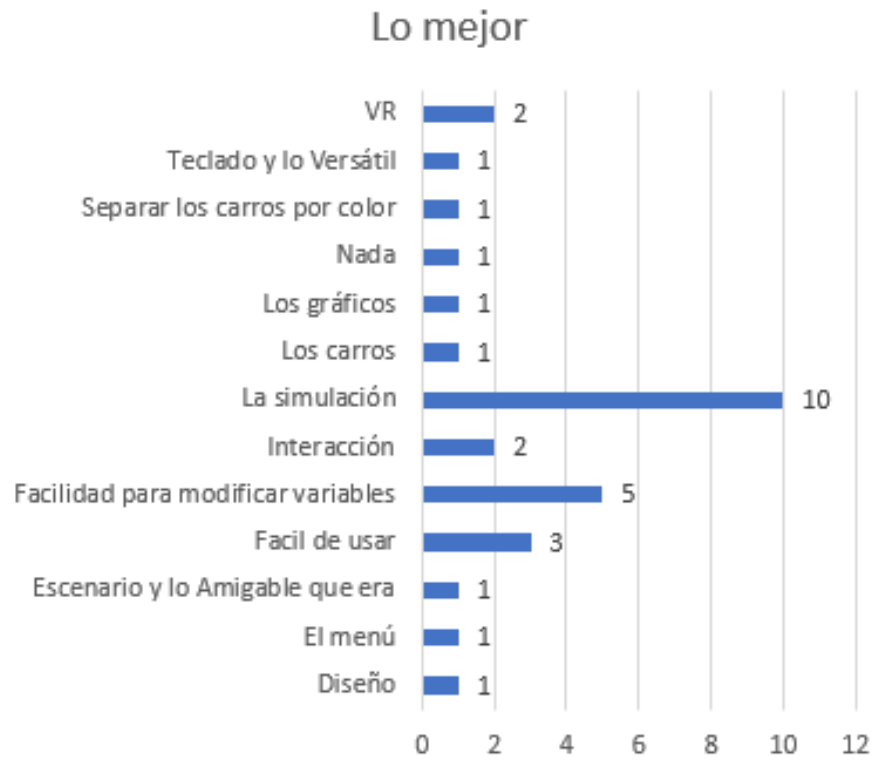


Figura 53: Gráfica de lo que más les gustó a los encuestados

B. Encuesta

A continuación se muestra la encuesta usada para los usuarios.

Física Móvil
Freddie Batlle

Edad:

Carrera:

Ha probado alguna vez una aplicación de realidad virtual?

Sexo:

Cuestionario Pre-test

Preguntas (En escala de 1 a 10 donde 1 es muy poco y 10 es mucho)	Respuesta
Qué tan fácil te es visualizar (imaginar) problemas físicos?	
Qué tan fácil te es visualizar problemas (de física) de choques?	
Qué tan fácil te es ver la relación entre las variables y un problema de choque y su resultado?	
Qué cree que pasará en un choque de carros iguales (misma masa y misma velocidad) y totalmente inelástico (coeficiente de restitución 0)?	
Qué cree que pasará si ahora uno de los carros tiene más masa?	

Cuestionario Post-test

Pregunta	Respuesta
Que cree que pasará en un choque de carros iguales (misma masa y misma velocidad) y totalmente inelástico (coeficiente de restitución 0)?	
Que cree que pasará si ahora uno de los carros tiene más masa?	
Cuanto más fácil o más difícil te fue responder las preguntas anteriores después de usar la aplicación?	

(En escala de 1 a 10 donde 1 es muy poco y 10 es mucho)	
Qué tan fácil te es visualizar (imaginar) problemas físicos?	
Qué tan fácil te es visualizar problemas (de física) de choques?	
Qué tan fácil te es ver la relación entre las variables y un problema de choque y su resultado?	
Cuanto más fácil o más difícil te fue responder las preguntas anteriores después de usar la aplicación?	

Qué te ha parecido el diseño gráfico de la aplicación?	
Necesitaste ayuda con algo respecto a la aplicación?	
Sientes que hay poca,mucha o suficiente información dentro de la aplicación?	
Qué fue lo que más apreciaste de la aplicación?	

Qué fue lo que menos apreciaste de la aplicación?	
Tuviste problemas para usar la aplicación?	
Qué tanto te ayudaría tener esta aplicación a la hora de resolver problemas de choque? (Donde 1 es muy poco y 10 es mucho)	
Quisieras poder descargar la aplicación y usarla?	
Crees que sería beneficioso para el salón de clase el uso de la aplicación?	
Quisieras poder representar otros problemas físicos para visualizar mejor los problemas de tus tareas?	

Acuerdo de privacidad

Mi nombre es Freddie Batlle, soy miembro de la Universidad del Valle de Guatemala. Me identifico con número de carnet 14074.

En esta prueba a realizar se empezará por preguntarle cierta información personal la cual se incluirá dentro del reporte final de esta investigación exceptuando el nombre de la persona, todos los demás datos (edad, carrera, etc) sí serán incluidos en el reporte final de esta investigación con el único uso de estadística de los usuarios.

Se le tomará una foto únicamente con el objetivo de dejar constancia de la veracidad de la prueba realizada.

El usuario tiene permitido hacer cualquier pregunta antes, durante y después del test y tiene toda la libertad de retirarse de la prueba en cualquier momento de esta.

He leído la información proporcionada o me ha sido leída. He tenido la oportunidad de preguntar sobre ella y se me ha contestado satisfactoriamente las preguntas que he realizado. Consiento voluntariamente participar en esta investigación como participante y entiendo que tengo el derecho de retirarme de la investigación en cualquier momento sin que me afecte de ninguna manera.

Fecha _____
Día/mes/año

- Aliasing:** Es cuando en una imagen de resolución muy baja se pueden observar en los contornos “gradas de pixeles”. Anti-aliasing se encarga de alisar estas gradas calculando y poniendo un promedio del color que deberían tener los pixeles alrededor de las mismas. 20
- Asset:** Es una representación de cualquier item que puede ser usado en un proyecto de Unity. Puede provenir de archivos externos a Unity, audio, un modelo 3D o cualquier otro tipo de archivo que soporta Unity. También puede venir de archivos creados por Unity como controlador de animación. 15
- Canvas:** Es el componente raíz para renderizar todos los objetos de la interfaz de usuario en Unity. 18
- Choque elásticos e inelásticos:** El tema físico de choques elásticos e inelásticos es el evento en que dos cuerpos interactúan entre sí físicamente en donde al menos uno de estos está en movimiento. Esta colisión puede ser elástica o inelástica. 18
- Clamp:** Esta función proporcionada por Unity permite restringir un valor a un rango que está definido por el valor mínimo y máximo. Devuelve el valor ingresado si está dentro del rango, devuelve el mínimo si es menor o devuelve el máximo si es mayor. 64
- Collider:** Un elemento invisible que define la forma de un objeto para el propósito de una colisión física. 18
- Coroutines:** Es como una función que tiene la habilidad de pausar su ejecución y regresar el control del flujo a Unity, pero luego continuar la ejecución donde se quedó en el siguiente frame. 14
- Drag:** Es la tendencia de un objeto a disminuir su velocidad debido a la fricción con el air o el agua que lo rodea. 17
- Formato FBX:** Es un formato de intercambio de objetos en 3D, este puede tener animaciones y jerarquía de sus partes. 15

Frame: Son imágenes individuales en una secuencia de imágenes. Frame-rate es el número de imágenes o frames que son mostradas por segundo. 14

Game Engine: Es un framework hecho para la creación y desarrollo de un videojuego. Se usa para crear software para consolas, móviles y computadoras. Proveen un motor renderer para gráficas 2D o 3D, un motor físico (principalmente detector de colisiones), sonido, scripting, animación, inteligencia artificial, opciones multijugador, streaming, manejo de memoria y threads y a veces soporte para videos o cinemáticas. Sin embargo, se pueden usar para usos diferentes al de hacer videojuegos. 13

Google Cardboard: Es una plataforma de realidad virtual (VR) desarrollada por Google sobre la base de cartón plegable que funciona a partir de montar un teléfono móvil inteligente con Android o IOS. Los lentes están para dar la sensación de profundidad. Los campos de visión para el ojo izquierdo y derecho están delimitados por una franja de cartón separatoria en el centro de las gafas. Los cristales crean un efecto lupa. 13, 16

Google VR: Es un grupo de productos, desarrollados por Google, con enfoque en la realidad virtual, estos incluyen lentes de realidad virtual como Google Cardboard y Daydream, cámaras 180 grados en vr, programas para pintar en 3D para lentes de realidad virtual, entre otros. 13

Input: Conjunto de datos que se introducen en un sistema o en un programa informático. 13

Interpolación lineal: Es un procedimiento muy usado para estimar los valores que toma una función en un intervalo del cual conocemos sus valores en los extremos. Para esto se utiliza la aproximación a la función $f(x)$ por medio de una recta $r(x)$. La expresión se obtiene del Polinomio interpolador de Newton de grado uno. Este método (interpolación de Newton) es muy útil cuando es un número pequeño de puntos a interpolar ya que a medida que crece el número de puntos también lo hace el polinomio. 24

Lerp: Linealmente interpola entre dos puntos a, b por un tiempo t. Al parámetro t se le aplica la función “clamp” en el rango [0, 1]. Estas operaciones están implementadas en el hardware de todos los procesadores gráficos modernos. 24

Line renderer: Este es un componente que toma un array de dos o más puntos en un espacio en 3D y dibuja una línea recta entre cada uno. Un solo componente de Line Renderer puede ser usado para dibujar cualquier cosa desde una simple línea hasta un complejo espiral. La línea es siempre continua; si se deseara dibujar 2 o más líneas completamente separadas, se deberá usar múltiples GameObjects, cada uno con su propio Line Renderer. Este componente no renderiza líneas de 1 píxel de anchura. Renderiza líneas “billboard” (polígonos que siempre miran a la cámara) que tienen una anchura en unidades del mundo y pueden ser texturizadas. 24

Local position: En Unity es la posición relativa al objeto padre. 24

Low Poly: Es una colección de vértices, aristas y caras que definen la forma de un objeto poliédrico, en graficas de computación y modelado. Este particularmente tiene una

baja cantidad de polígonos. Los polígonos comúnmente se dividen en triángulos. Entre más triángulos hay en un polígono el proceso computacional aumenta, pero el objeto es más detallado. 14

Magnitud de vector: Distancia entre el punto origen del vector y su punto final. 24

Material físico: Es una serie de variables que se aplican a un objeto para modificar su fricción y la elasticidad con los objetos con los que colisiona. 17

Mesh: Es una colección de vértices, bordes y caras que describen la forma de un objeto en 3D. 14

Modelo: Es una representación matemática de un objeto en tres dimensiones. 14

NewtonVR: Es un sistema, desarrollado por Tomorrow Today Labs enfocado en las aplicaciones en VR. Le permiten al usuario agarrar, dejar caer, tirar y usar objetos en la mano. Los objetos no traspasan otros objetos ni el entorno. Los items que se agarran interactúan con otros de forma natural, tomando en consideración la masa. Los items tienen opciones de configuración que permiten ser agarrados o que se les aplique alguna rotación cuando se agarran. También provee algunos menús y elementos para la configuración del sistema. 13

Normalización de vector: El proceso de hacer que un vector tenga una magnitud de 1. Esto implica que aún mantiene su dirección, pero tiene una longitud de 1. 24

Oculus Rift: Es un dispositivo de visualización, desarrollado por Oculus Vr, similar a un casco, que permite reproducir imágenes creadas por ordenador sobre una pantalla muy cercana a los ojos o proyectando la imagen directamente sobre la retina de los ojos. Debido a su proximidad con los ojos, el casco de realidad virtual consigue que las imágenes visualizadas resulten mucho mayores que las percibidas por pantallas normales, y permiten incluso englobar todo el campo de visión del usuario. Gracias a que el casco se encuentra sujeto a la cabeza, éste puede seguir los movimientos del usuario, consiguiendo así que éste se sienta integrado en los ambientes creados por ordenador. Este necesita de una computadora para poder usarse y no hay forma de usarlo en dispositivos móviles. 13

Polinomio interpolador de Newton de grado uno: $f_1(x) = p_0 + p_1(x - x_0)$ Donde:
 $p_0 = f(x_0)$ $p_1 = f(x_1, x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$. 64

Raycasting: Es el proceso de lanzar un rayo invisible desde un punto hacia una dirección específica para detectar con que objetos colisiona en el camino. 13

Script: Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guion. El uso habitual de los guiones es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. 15

Sistema de trabajos: Este maneja código multithread creando trabajos en vez de threads. Maneja un grupo de threads trabajadores a través de múltiples cores del CPU. Por lo general maneja un thread por core lógico del CPU. Este sistema pone trabajos en una cola de trabajos a ejecutar. Los threads trabajadores agarran los trabajos de esta cola y los ejecutan. El sistema de trabajamos maneja dependencias y se encarga de revisar que las tareas se ejecuten en el orden establecido en la cola. 14

Tag: Es una palabra de referencia que se le puede asignar a uno o más objetos para identificarlos en los scripts. 18

Thread: Es un mecanismo que permite a una aplicación realizar varias tareas a la vez de manera concurrente. Los Threads son una ampliación del concepto de multitarea, si bien multitarea se refiere a la capacidad de un sistema para ejecutar varios procesos a la vez, en un comienzo esto hacía referencia a que más de una aplicación se estuviera ejecutando de manera concurrente, sin embargo, pronto se hizo notoria la necesidad de que una misma aplicación hiciera varias cosas a la vez. 14

Threadsafe: Es un concepto de programación aplicable en el contexto de los programas multihilos (multithread). Una pieza de código es segura en cuanto a los hilos si funciona correctamente durante la ejecución simultánea de múltiples hilos. En particular, debe satisfacer la necesidad de que múltiples hilos accedan a los mismos datos compartidos, y la necesidad de que una pieza compartida de datos sea accedida por solo un hilo en un momento dado. 14

Unity3d: Es un motor gráfico 3D para PC y Mac que viene empaquetado como una herramienta para crear juegos, aplicaciones interactivas, visualizaciones y animaciones en 3D y tiempo real. El editor de Unity es el centro de la línea de producción, ofreciendo un completo editor visual para crear juegos. El contenido del juego es construido desde el editor y el gameplay se programa usando un lenguaje de scripts. Los juegos creados en Unity son estructurados en escenas. Una escena puede ser cualquier parte del juego, desde el menú de inicio como un nivel o área del juego. El motor también incluye un editor de terrenos, desde donde se puede crear un terreno (como una hoja en blanco), sobre la que los artistas podrán esculpir la geometría del terreno usando herramientas visuales, pintar o texturizar, cubrir de hierba o colocar árboles y otros elementos de terreno importados desde aplicaciones 3D como Blender, 3DS Max o Maya. Unity es un sistema de desarrollo único. Es enfocado en los assets y no en el código, el foco en los assets es similar al de una aplicación de modelado 3D. Un juego construido en Unity se compondrá de un proyecto. Contiene todos los elementos de su proyecto, tales como los modelos, guiones, planos, menús, entre otros. Unity hace el proceso de producción de juego simple dándole un set de pasos lógicos para construir cualquier panorama concebible de juego. Establece el concepto Game Object, donde se puede estudiar los componentes del juego en objetos dóciles, que está hecho de muchos componentes individuales. Haciendo objetos individuales dentro del juego e iniciando funcionalidad en ellos con cada componente que se suma, se puede expandir el juego en una manera progresiva lógica. Los componentes a su vez tienen variables, esencialmente por los cuales serán controlados. 13

Vr: La realidad virtual (VR) es una emocionante y nueva tecnología que promete transformar de manera fundamental la forma en la que interactuamos con la información,

nuestros amigos y el mundo en general. La realidad virtual para el consumidor se basa en usar una especie de pantalla en la cabeza (como el Google Cardboard), se pueden ver escenas en 3D estereoscópicas. Se puede ver alrededor por medio del mismo movimiento de la cabeza y muchas veces se puede caminar y moverse a través de las manos usando los controles. Se puede estar en una experiencia verdaderamente inmersiva. Simulando estar en un mundo virtual o una virtualización del nuestro. 13