



# VROOM - INFORME FINAL

DOMINGO 04 DE DICIEMBRE

## **Curso:**

*CS2H01 - Interacción Humano Computador*

## **Docente:**

Teofilo Chambilla - *tchambilla@utec.edu.pe*

## **Integrantes:**

Abril Vento - *daniela.vento@utec.edu.pe*

Diego Paredes - *diego.paredes.r@utec.edu.pe*

Efraín Córdova - *efrain.cordova@utec.edu.pe*

## CONTENTS

|            |  |           |
|------------|--|-----------|
| <b>I</b>   | <b>Antecedentes</b>                              | <b>3</b>  |
| <b>II</b>  | <b>Requisitos de diseño y persona</b>            | <b>3</b>  |
| II-A       | Requerimientos . . . . .                         | 3         |
| II-A1      | Requerimientos funcionales . . . . .             | 3         |
| II-A2      | Requerimientos no funcionales . . . . .          | 3         |
| II-A3      | Explicación de los requerimientos . . . . .      | 4         |
| II-B       | Audiencia . . . . .                              | 4         |
| II-C       | Especificación final de persona . . . . .        | 4         |
| II-D       | Objetivo de la experiencia . . . . .             | 5         |
| II-E       | Narrativa de la aplicación . . . . .             | 5         |
| II-F       | Arco de la historia . . . . .                    | 5         |
| II-G       | Métrica HEART . . . . .                          | 6         |
| <b>III</b> | <b>Framework de diseño</b>                       | <b>6</b>  |
| III-A      | Descripción del framework . . . . .              | 6         |
| III-B      | Plan de diseño . . . . .                         | 6         |
| III-B1     | Controles . . . . .                              | 6         |
| III-B2     | Escenas . . . . .                                | 7         |
| III-B3     | Consideraciones de diseño . . . . .              | 9         |
| <b>IV</b>  | <b>Prototipo</b>                                 | <b>9</b>  |
| IV-A       | Descripción General . . . . .                    | 9         |
| IV-B       | Aspectos internos de la implementación . . . . . | 10        |
| <b>V</b>   | <b>Implementación y Evaluación</b>               | <b>10</b> |
| <b>VI</b>  | <b>Discusión</b>                                 | <b>12</b> |
|            | <b>Referencias</b>                               | <b>13</b> |

## I ANTECEDENTES

Las habilidades de estacionamiento de algunos de los conductores en Lima son muy pobres, lo que se evidencia en casi cualquier estacionamiento al que se vaya en esta ciudad: muy pocos son los casos en los que no se ve al menos un carro que esté parado encima de la línea que marca el límite de su espacio para estacionarse. Esto es comprensible, pues con lo desordenado que es el tráfico en Lima (considérese el hecho de que Lima está rankeada en puesto 19 de las ciudades con el peor tráfico [1]), el proceso para estacionarse puede ser estresante.

Asimismo, se conoce que uno de los errores más comunes al momento de rendir el examen de manejo es no saber estacionar [2], y es que, fuera de lo que puede ofrecer un paquete de una academia de manejo, lugares para practicar en Conchán, el mismo Touring en sí y quizá alguna zona despejada que se haya podido encontrar con mucha suerte, no hay muchos espacios donde puedan practicar sin interrupciones (o posibilidad de recibir una papeleta). Esta falta de ambientes de práctica nos lleva a diseñar un simulador de manejo en el cual el usuario podrá practicar estacionarse.

## II REQUISITOS DE DISEÑO Y PERSONA

### II-A REQUERIMIENTOS

#### 1) *Requerimientos funcionales*

- **RF001 (Must-have):** El usuario puede interactuar con el escenario usando los controles establecidos.
- **RF002 (Must-have):** El usuario puede poner a prueba sus capacidades de manejo en diferentes escenarios.
- **RF003 (Must-have):** El usuario puede poner a prueba sus capacidades para estacionarse en diferentes escenarios.
- **RF004 (Must-have):** El usuario puede recibir retroalimentación sobre su desempeño en los circuitos que completa.
- **RF005 (Must-have):** El usuario puede elegir el modelo del carro y el circuito.

#### 2) *Requerimientos no funcionales*

- **RNF001 (Must-have):** La aplicación es compatible con Oculus Quest 2.
- **RNF002 (Must-have):** El sistema funciona correctamente para un sólo jugador.
- **RNF003 (Must-have):** El sistema incluye una documentación explícita durante su uso que permite al usuario entender las mecánicas de manejo en todo momento.

### *3) Explicación de los requerimientos*

Para la propuesta de la solución, nos centramos particularmente en resolver los siguientes problemas:

- Las prácticas de manejo son actividades supervisadas por el instructor del alumno. Además, dichas prácticas son limitadas y dependen de la matrícula del estudiante. Se desea, entonces, brindar espacios interactivables donde el alumno pueda, libremente, poner a prueba los conceptos teóricos aprendidos (**RF001 y RF002**).
- Muchos conductores, incluso luego de dominar el resto de aspectos de la conducción, encuentran difícil el estacionarse. Es, ciertamente, un proceso estresante que es difícil de emular en un entorno tranquilo debido a la congestionada movilización automovilística de la ciudad de Lima. Se desea, entonces, que el usuario pueda poner a prueba su expertiz en esta actividad, de forma que el sistema entregue retroalimentación valiosa que le permita mejorar su rendimiento (**R003 y RF004**).

Nuestra solución es construir un programa que sirva como plataforma útil para resolver estos conflictos que se le presentan a nuestro público objetivo (ver Audiencia). Identificamos que este programa debe cumplir con lo siguiente:

- Debe ser compatible con el Oculus Quest 2. Por condiciones económicas, este será el único dispositivo compatible. Se debe asegurar, además, que el funcionamiento de todas las mecánicas deben funcionar correctamente para un sólo jugador (**RNF001, RNF002**).
- El sistema es un programa didáctico: debe contener indicaciones claras de su uso a través de una documentación disponible en todo momento de la experiencia (**RNF003**).

## **II-B AUDIENCIA**

La audiencia a la que se dirige el sistema son personas limeñas con conocimientos teóricos de manejo aspirantes a una licencia de conducir -o ya poseedoras de una- que necesiten reforzar sus habilidades para manejar y que entre sus principal déficit como conductor se encuentra a la hora de estacionar el vehículo.

## **II-C ESPECIFICACIÓN FINAL DE PERSONA**

Algunos ejemplos de posibles usos incluyen:

- Un adolescente de 17 años que está preparándose para sacar su licencia ni bien cumpla 18, pero cuyos padres se rehúsan de llevar hasta el Touring porque les queda lejos de casa.

- Un alumno universitario en prácticas preprofesionales que le quedan lejos de casa, motivo por el cual quiere sacar una licencia para transportarse con mayor flexibilidad. Debido a que tiene que balancear su tiempo de estudio con el trabajo, tiene pocos espacios para prepararse para su examen práctico de manejo ni transportarse hasta Conchán.
- Un padre de familia de 45 años que ha estado en cuidado clínico por muchos meses y está por volver a las vías, pero necesita un poco de refuerzo, pero que por sus tareas familiares y proceso de rehabilitación le está dificultando conseguir ese espacio.
- Un conductor que nunca aprendió a estacionarse bien y tiene miedo de que le metan papeleta algún día por dejar su carro mal cuadrado.

En otras palabras, son personas que ya saben manejar pero necesitan practicar o reforzar manejar y están bajo alguna circunstancia o circunstancias que evitan que puedan hacerlo.

## II-D OBJETIVO DE LA EXPERIENCIA

El objetivo de la experiencia es emular la situación de manejar un vehículo, de forma que el usuario puede practicar esta habilidad en un entorno seguro, con el fin de mejorar en ella.

## II-E NARRATIVA DE LA APLICACIÓN

- 1) El usuario ingresa a la pantalla de inicio.
- 2) El usuario elige su carro y su circuito.
- 3) La escena cambia al circuito elegido. El usuario se encuentra dentro del carro que ha escogido previamente.
- 4) El usuario recibe las indicaciones básicas de como controlar el vehículo. Se le solicita que complete el circuito.
- 5) El usuario debe avanzar por la pista, y al pasar por la zona de estacionamiento, deberá estacionar el vehículo en paralelo y en diagonal.
- 6) El usuario debe retirar el vehículo del estacionamiento y completar el circuito.
- 7) Al llegar al final de este, el sistema provee feedback al usuario basado en su rendimiento.

## II-F ARCO DE LA HISTORIA

- 1) **Exposición:** menú en el cual el usuario elige de carro y circuito
- 2) **Acción ascendente:** aparecer en el circuito, estando en el carro.
- 3) **Climax:** conducir y estacionar el carro.

- 4) **Acción descendente:** completar el circuito.
- 5) **Descenlace:** recibir retroalimentación del desempeño en el circuito.

## II-G MÉTRICA HEART

| Criterio            | Objetivos                 | Señales                       | Métricas                                  |
|---------------------|---------------------------|-------------------------------|---|
| <b>Happiness</b>    | Satisfacción del usuario  | Encuestas de satisfacción     | Puntaje en NASA TLX                       |
| <b>Engagement</b>   | Disfrute de la aplicación | Tiempo de uso                 | Duración de la sesión y frecuencia de uso |
| <b>Adoption</b>     | Integración de usuarios   | Descargas                     | Número de nuevos usuarios                 |
| <b>Retention</b>    | Lealtad del usuario       | Datos de usuarios retornantes | Veces que se ha usado la aplicación       |
| <b>Task success</b> | Logros de los usuarios    | Uso que le da el usuario      | Tiempo para completar el circuito         |

## III FRAMEWORK DE DISEÑO

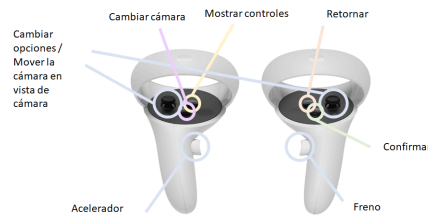
### III-A DESCRIPCIÓN DEL FRAMEWORK

El aplicativo se centra en la implementación de circuitos de manejo mediante los cuales se pueda practicar manejar y estacionarse. Específicamente, nos centramos en la mecánica de estacionamiento. Los escenarios constan de una pista de manejo con algunas zonas de parqueo; la idea es que el usuario recorra un segmento del circuito y termine la experiencia estacionando el vehículo en alguna de las zonas destinadas para ello.

### III-B PLAN DE DISEÑO

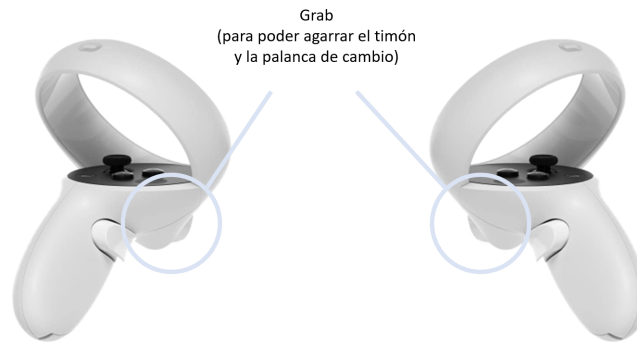
#### 1) Controles

Se consideran los siguientes controles para la implementación:



**Fig. 1:** Vista de frente de los controles

Asimismo, el usuario puede mover su cabeza usando el headset para cambiar el ángulo de la cámara.



**Fig. 2:** Vista lateral de los controles

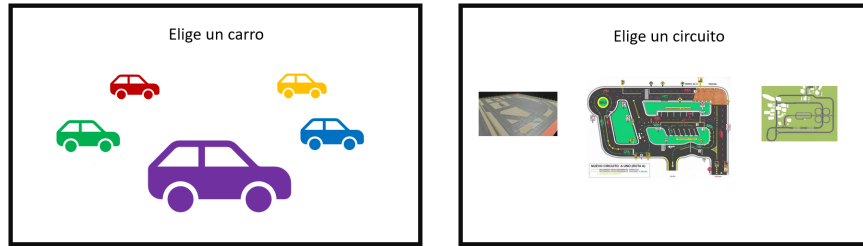
## 2) Escenas

Para la interacción, se parte de una pantalla de inicio que muestra el título, los créditos respectivos del aplicativo y un mensaje en pantalla que indica cómo continuar. Asimismo, se puede presionar "Y" para ver los controles.



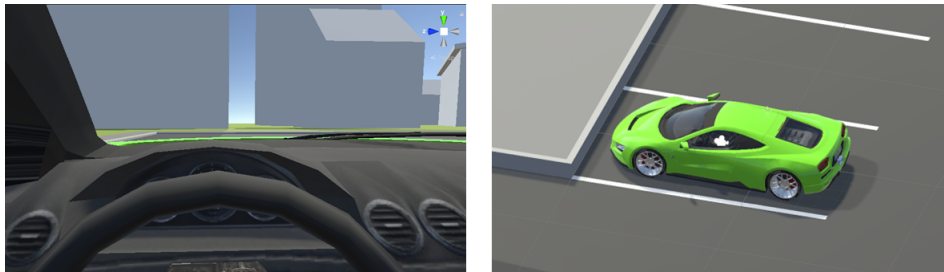
**Fig. 3:** Inicio

Tras ello, se pasa a una escena de configuración. En esta, el usuario usa los joysticks de cualquiera de los controllers para moverse entre los modelos de carro. El modelo que se muestra al centro en un momento dado es el que está por elegirse. Se elige usando el botón "A" en el controller de la mano derecha. Estos mismos pasos aplican para la selección del circuito. Se usa "B" para regresar a la pantalla anterior.



**Fig. 4:** Configuración

Después de elegir ambas cosas, se pasa a la escena de manejo. La cámara, y por tanto la vista del usuario, está dentro del carro. El usuario puede manejar este vehículo por el circuito y estacionarse en la zona de estacionamiento. También tiene una opción que le permite cambiar la vista de la cámara.



**Fig. 5:** Conducción

Tras completar el circuito y manteniendo la escena anterior, aparece un splash screen con un resumen de datos relevantes de la sesión de manejo: tiempo en completar el circuito, distancia recorrida, tiempo empleado (solo estacionarse) y tiempo empleado (solo conducir). Presionar "A" da paso a otro splash screen que le permite hacer el circuito de nuevo, regresar al resumen o volver al menú principal.



**Fig. 6:** Feedback



### *3) Consideraciones de diseño*

Cabe mencionar que para los textos incluidos en el aplicativo, se tomaron en consideración las conclusiones de un estudio [3] respecto a visualización de texto en VR. Por este motivo, los textos están en 22pt font size, estilo normal, en una sola columna y en un fondo claro con letras oscuras.

El sistema satisface algunas de las heurísticas de usabilidad consideradas por un checklist realizado para tecnologías AR y MR [4], pero se consideraron aspectos prudentes a VR con ellas. Se listan a continuación:

- Ayuda y documentación disponibles en todo momento, pues se provee ayuda al usuario al ofrecerle una forma de ver los controles en todo momento.
- Integración de mundo físico y mundo virtual a través de controles del Oculus Quest 2.
- Consistencia y estándares (mecánicas de conducción tradicionales)
- Feedback integrado para cuantificar el rendimiento del usuario.
- User interaction.
- Reconocimiento en lugar de memoria (sistema intuitivo).

Una de las limitaciones que nos podremos encontrar a la hora de desarrollar el proyecto va a ser la fidelidad que podamos alcanzar con los controles proporcionados. Un factor muy importante para herramientas de asistencia en el aprendizaje, es la cercanía de esta herramienta a la realidad. Nosotros no contamos con periféricos de uso especializado como lo serían un volante o pedales de vehículos, es por ello que usaremos solamente los controles del Oculus como interfaz entre el usuario y el vehículo virtual. Es por este motivo también que nos ceñimos a trabajar con vehículos que tengan una palanca de cambios automática, pues la lógica detrás de un carro con palanca mecánica es más complicada de simular.

## **IV PROTOTIPO**

### **IV-A DESCRIPCIÓN GENERAL**

El usuario va a aparecer en el asiento de conductor del vehículo. Aquí el usuario va a poder usar el grip de sus mandos para poder agarrar el timón del vehículo. Asimismo, el usuario cuenta con la libertad de mover su cámara con la cabeza. Finalmente, el usuario puede accionar el acelerador y el freno del vehículo con los gatillos.



**Fig. 7:** Prototipo: usuario dentro del carro

#### **IV-B ASPECTOS INTERNOS DE LA IMPLEMENTACIÓN**

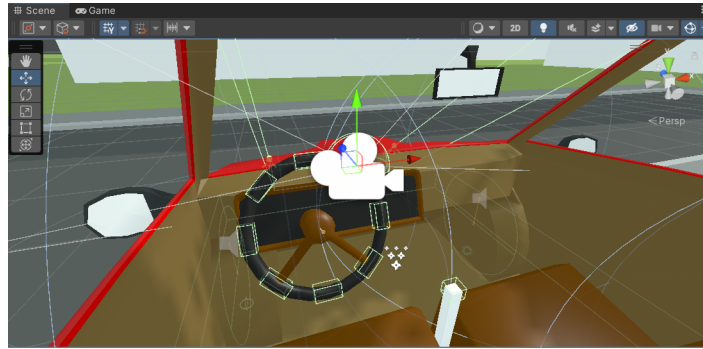
Para el desarrollo de este prototipo hemos usado Unity 3D con el módulo de integración de Oculus. Debido a esto estamos restringidos a usar el lenguaje de programación C# para el scripting de los componentes más personalizados de nuestro software lo cual no es un problema en sí, pero puede llegar a ser algo a tomar en cuenta para el desarrollo de nuevos features. Por otro lado, hemos contado con algunos problemas con el engine Unity 3D debido a que hacemos usos de diferentes versiones de este en nuestros computadores personales. Finalmente para poder trabajar remotamente este proyecto hacemos uso de GitHub.

### **V IMPLEMENTACIÓN Y EVALUACIÓN**

Se logró implementar parte de las funcionales para dos modelos de carros: el agarre del timón y la palanca de cambio. Asimismo, se añadieron sonidos del motor para poder mejorar la inmersión. Para lograr esto, tomamos referencias de un proyecto ya existente que trabajó con carros en realidad virtual [5] y uno que configuró el movimiento de un auto en Unity [6].

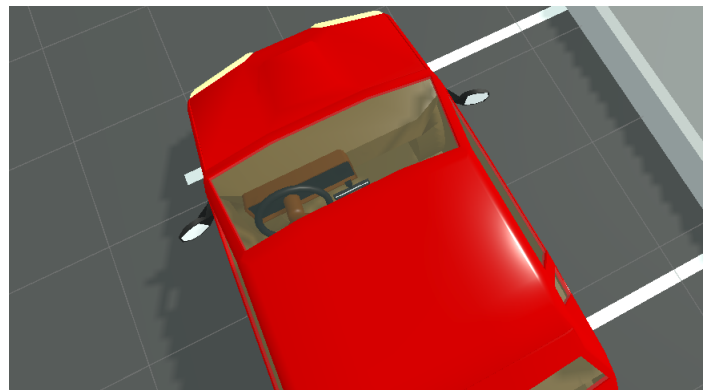


**Fig. 8:** Implementación: carro con timón y palanca



**Fig. 9:** Implementación: carro con timón y palanca puntos de agarre

El vehículo también puede moverse, pero nos costó mucho lograr este funcionamiento debido a problemas con algunos objetos en la escena, cuales se vieron acentuados al momento de trasladarnos de OVR a XR, puestos a que el cambio de toolkit significaba que la configuración hecha en OVR no se podía trasladar a XR de manera directa. Esto nos llevó a tener problemas con la cámara de XR y el collider del vehículo, donde el principal era que la cámara no se quedaba de forma estática dentro del vehículo, sino que esta aparecía por encima haciendo imposible el uso del vehículo por parte del usuario.



**Fig. 10:** Implementación: vista del usuario desde el Oculus

Adicionalmente, faltó ajustar la palanca para que se ajuste al movimiento esperado por su posición si se considera una palanca automática, aunque esto es cuestión de modificar un par de variables en el script de C#, solo que no pudimos probarlo debido a la falta de un *Oculus* a nuestra disposición. También faltó modificar el panel de control en el modelo para que pueda mostrar el tacómetro, velocímetro, señales de giro y otras luces.

En cuanto a las otras escenas, no pudieron implementarse, así que no pudimos probar el flujo de completo la aplicación. Teníamos planeado añadir el mapa por cada uno de los circuitos que utilizemos, pero debido a que no se logró trabajar con más de uno, no se llegó a agregar esta función.

Respecto al feedback, no se pudieron hacer cálculos debido a que no se pudo probar el funcionamiento del carro en su totalidad. Se tenía pensado proporcionar métricas que detallen el tiempo que le tomo al usuario realizar ciertas acciones. Principalmente, se iba a detallar el tiempo que le tomaba al usuario: encontrar un lugar para estacionarse, estacionarse (sea en paralelo o diagonal) y salir del estacionamiento.

## VI DISCUSIÓN

En un inicio se planteó poder realizar un aplicativo con todo el flujo mostrado en la primera sección. Sin embargo, debido a que nos concentramos mucho en hacer funcional la mecánica principal del aplicativo, cual era el *feature* de manejo, limitaciones de acceso a los equipos de Oculus debido a cruces con los horarios del equipo de trabajo, algunos problemas de gestión de tiempo y factores inesperados como dos miembros del equipo enfermándose, no pudimos culminar el proyecto. En su estado actual, como se mencionó en la parte de implementación, no es jugable.

Si pudiéramos repetir el proceso de elaboración de este proyecto, partiríamos por planificar mejor las sesiones de trabajo. El proceso de creación no fue eficiente debido a que no nos trazamos objetivos claros antes de poder utilizar los *Oculus*. Luego, a pesar de que pudimos implementar algunas funcionalidades, determinar cual sería la próxima mecánica a implementar era algo ambiguo para nosotros, principalmente por nuestro poco expertiz en *Unity*. Creemos que invertir más tiempo para aprender esta herramienta y utilizar ese conocimiento para diseñar un mejor plan de implementación nos hubiese acercado más a la meta inicial.

Adicionalmente, una vez cubiertos los requerimientos que planteamos inicialmente, consideraríamos que algunos que podrían añadirse en un futuro incluyen añadir un circuito que simule al que se encuentra en el Touring para poderlo hacer una herramienta de práctica más poderosa para estudiar para el examen práctico de manejo. Además, sería bueno añadir una funcionalidad en multijugador para que se pueda simular el tráfico o incluso la misma experiencia de un examen de manejo a más detalle puesto que al momento de dar este examen no es solo una persona quien lo toma a la vez.

## **REFERENCIAS**

- [1] TomTom, “Lima Traffic”, 2022, <https://www.tomtom.com/traffic-index/lima-traffic/>
- [2] Gob.pe, “Conozca cuáles son los errores más comunes al realizar la prueba de manejo en el circuito del Touring”, 2020, <https://www.gob.pe/es/n/312630>
- [3] VAMR, Kojic T, et al., “Assessing User Experience of Text Readability with Eye Tracking in Virtual Reality”, 2022, [https://doi.org/10.1007/978-3-031-05939-1\\_13](https://doi.org/10.1007/978-3-031-05939-1_13)
- [4] J. Derby, B. Chaparro, 2022, “The Development and Validation of an Augmented and Mixed Reality Usability Heuristic Checklist”, [https://doi.org/10.1007/978-3-031-05939-1\\_11](https://doi.org/10.1007/978-3-031-05939-1_11)
- [5] Valem. “I Made Mario Kart In VR”, *YouTube*, 1 de febrero del 2021 [Archivo de video]. Disponible en: <https://www.youtube.com/watch?v=PRG1PyXPwlg>
- [6] Mix and Jam. “Mario Kart’s Drifting — Mix and Jam”, *YouTube*, 20 de junio del 2019 [Archivo de video]. Disponible en: <https://www.youtube.com/watch?v=Ki-tWT50cEQ>

## **ANEXOS**

[Enlace al repositorio del prototipo](#)