



UNIVERSIDAD SIMÓN BOLÍVAR

CI5852

Informe Proyecto AR

Estudiante:

Pietro Iaia 15-10718

Diego Peña 15-11095

Emmanuel Bandres 14-10071

Sartenejas, Julio 2021

1. Título, descripción, plataformas del proyecto

Galactic Colony AR es un juego móvil de recolección de recursos local single-player, donde el jugador tomará el control de dos planetas casi deshabitados y, por medio de la recolección de los recursos que cada planeta genere, tratará de mejorar las instalaciones de cada uno, con el fin de aumentar la vida y producción en ellos.

Durante esta misión, cada planeta podrá sufrir desastres naturales o accidentales, de categoría regular y aniquiladores, los cuales retrocederán el progreso y harán más difícil llegar a la meta de poblar los planetas con 7 billones de personas.

El proyecto fue desarrollado para teléfonos, con cámara inteligente, que corran el sistema operativo Android.

2. Objetivos de la aplicación

Parte del objetivo del juego fue explicado en la sección anterior, pero el objetivo principal de la aplicación es entretener al jugador.

3. Mecánicas del proyecto (muy breve manual de uso)

Al iniciar la aplicación, el jugador podrá empezar su juego desde el principio o continuar el anterior si ya había empezado a jugar antes. Luego de esto, deberá apuntar su smartphone a las dos marcas AR para hacer que aparezcan los planetas. Una vez que tenga los planetas en vista, podrá tocarlos y hacer que aparezca el menú con la información de este, donde aparece la cantidad de recursos que ha generado, pero no recolectado, y la cantidad de personas que viven en él. También podrá tocar el botón de “recolectar” para recolectar los recursos generados o tocar el botón de “mejoras” para ir al módulo de mejoras de dicho planeta.

En el módulo de mejoras de cada planeta, el jugador podrá mejorar sus generadores de recursos, en el primer tab, y sus instalaciones, en el segundo tab, haciendo uso de los recursos recolectados de estos dos planetas. A medida que el juego avance, el planeta podría sufrir desastres que disminuyan la producción de recursos y eliminen parte de la población, por lo que el jugador debe estar preparado para afrontarlos y no malgastar recursos.

4. Ambiente donde se desarrolló y documentación del desarrollo

El proyecto fue desarrollado en el motor de videojuegos Unity, version 2019.4.26f1, haciendo uso de la herramienta del motor Vuforia para Unity, con el cual fue posible integrar las partes de realidad aumentada con el proyecto. El título en la pantalla de inicio fue diseñado en InkSpace, y toda la UI del juego fue diseñada en Photoshop.

Para las naves espaciales que salen de un planeta a otro, fue utilizado un modelo 3D descargado de internet llamado "Space Frigate", el cual contiene todas sus texturas y meshes integrados en un prefab listo para ser arrastrado en escena. El inspector es dividido en 3 partes, el canvas donde están todos los elementos UI del juego, La camara AR especial de Vuforia del cual todos los elementos AR parten y se encuentran, y la parte del manejador de juego donde todos los ajustes, generadores de planetas, y contadores globales se encuentran.

En el Game Handler tenemos los contadores de recursos que el jugador haya recolectado de los planetas, y las mecánicas de cargado y guardado del juego. En los GameObjects generadores de planetas se encuentran los scripts necesarios para generar el mapa de regiones del planeta, el cual solo se llama al iniciar el juego y cuando el planeta aumenta de nivel, que hace que cambie la vegetación y mares del planeta. Estos niveles del planeta se desbloquean a medida que aumenta la cantidad de habitantes:

- Para completar el primer nivel se necesitan 10.000 habitantes en el planeta.
- Para completar el segundo nivel se necesitan 100.000 habitantes en el planeta.

- Para completar el tercer nivel se necesitan 250.000 habitantes en el planeta.
- Para completar el cuarto nivel se necesitan 500.000 habitantes en el planeta.
- Para completar el quinto nivel se necesitan 750.000 habitantes en el planeta.
- Para completar el sexto nivel se necesitan 1.000.000 habitantes en el planeta.
- Para completar el séptimo, y último, nivel se necesitan 1.500.000 habitantes en el planeta.

Para el estado actual del proyecto, completar los niveles solo cambia el mapa del planeta, pero se podrían agregar mejoras desbloqueables que vengan como recompensa, los cuales por factores de tiempo no nos dio tiempo de agregar.

En la parte de canvas, existen 5 menús UI, dos que representan el menú principal, junto con el de mejoras del planeta, uno que representa el UI global del juego donde se encuentran los contadores de recursos recolectados, y dos últimos que representan los menús pop-up de desastres para cada planeta. Todos estos están completamente integrados con los planetas para que puedan mostrar su información. Para el caso de los menús principales de los planetas se logró que, al mostrarlos, estos estén “atados” al planeta y se muevan con él, por lo que si movemos el smartphone, veremos como el menú se queda por encima de la posición donde se encuentra el planeta. No se hizo lo mismo para los menús de desastres ya que pensamos que es mejor si solo se muestran como una alerta en el medio de la pantalla, la cual los jugadores pueden cerrar.

En cuanto a los planetas, estos contienen la mayor parte de la lógica del juego, ya que en ellos se ejecutan los generadores de recursos y desastres, manejadores de humanidad y mejoras, y otros scripts que permiten que el planeta se vea perfectamente en pantalla.

Empezamos hablando sobre el generador de recursos. En este, cada cierto tiempo se hará un intento de extraer recursos del planeta, con una probabilidad de extracción, y cantidad extraída, por cada recurso que dependerá del nivel en el que se encuentre dicho generador. Es decir, si el generador de hierro se encuentra en nivel tres, este

tendrá una probabilidad alta de extraer hierro más seguido, y cada vez que extraiga podrá extraer de entre 1 y 5 lingotes de hierro.

El generador de desastres funciona de forma similar al generador de recursos. Este, cada cierto tiempo, tendrá la posibilidad de generar un desastre en el planeta, donde los desastres regulares son más propensos a suceder que los desastres aniquiladores. El generador de desastres no es mejorable, por lo que los porcentajes de que sucedan no se cambiarán durante el juego. Un desastre regular es capaz de disminuir la población en un 26% máximo, y de disminuir en 1 el nivel de alguno de los generadores. Para los desastres aniquiladores, son capaces de disminuir la población en un 50% máximo y de disminuir en 2, o 3, los niveles de dos generadores.

Por último, el manejador de mejoras contiene todos los cálculos necesarios para aumentar de nivel los generadores e instalaciones del planeta. Estos cálculos se encuentran en funciones, las cuales son llamadas por los botones de UI que hacen subir de nivel a los elementos mencionados. En este, también se encuentran los diccionarios que contienen los recursos requeridos para aumentar un elemento a cierto nivel. Estos diccionarios son utilizados por los botones de UI para verificar si el jugador cuenta con los recursos necesarios para aumentar de nivel cierta estructura.

El manejador de humanidad lo único que hace es aumentar la humanidad cada cierto tiempo hasta llegar a un límite, el cual se puede aumentar al mejorar las instalaciones del planeta (no los generadores).

5. Marcas utilizadas, su descripción y justificación.



Marca del Planeta 1



Marca del Planeta 2

Estas marcas fueron generadas en [random.org](https://www.random.org) a través de un bitmap al azar, y luego fueron agregadas a la base de datos de Vuforia que utilizamos para el proyecto.

6. Posibles problemas que se encontraron durante el desarrollo

Durante el desarrollo se encontraron varios problemas de implementación. El principal problema de implementación fue la UI del juego, ya que en un principio esta era la que se encargaba del manejador de mejoras y de humanidad del planeta. Luego, nos percatamos que era bastante confuso e innecesario que toda esta lógica estuviera en un elemento de UI, que hacía varias llamadas a componentes del GameObject del planeta, y no en dicho GameObject que contenía toda la información que estos manejadores necesitan.

Otro problema que nos encontramos casi finalizando el desarrollo, fue que el generador de nubes del planeta 1 consumía mucha más RAM de la que pensábamos, lo cual hacía que el dispositivo móvil cerrara el juego a los 3 minutos por falta de memoria RAM. Este problema no ocurría en el entorno Game de Unity, por lo que nos dimos cuenta muy

tarde que existía, precisamente al momento de generar la apk del juego. Para solventarlo, tomamos la decisión de eliminarlas del juego por falta de tiempo para poder modificar su lógica.

7. Posibles cambios en el planeamiento inicial

No ocurrieron cambios muy drásticos del planteamiento inicial, lo único que no se logró implementar fue un tercer planeta por falta de tiempo, con el cual el jugador podría obtener aún más recursos pero con más probabilidades de desastres ya que podrían ocurrir en cualquiera de los tres planetas, aumentando el nivel de dificultad del juego.

8. Conclusión y Recomendaciones a futuro

Durante el desarrollo de este juego AR pudimos aprender sobre cómo funciona el sistema de posición en Unity cuando se acopla con la cámara AR de Vuforia, donde esta traduce la posición del mundo real a la posición en la escena del proyecto, que se denota como un 3-tupla la cual guarda la posición en el axis X, Y y Z respectivamente.

El proceso de desarrollo de una aplicación de AR fue bastante más llevadero gracias al uso del ambiente elegido para el desarrollo, ya que estas herramientas se encargaban de la mayoría de los cálculos que traducían el mundo real al mundo de la escena. Hacer que el desarrollo de aplicaciones AR sea más fácil permite que aún más desarrolladores puedan intentar crear su propia aplicación en esta área, lo cual amplía en parte el conocimiento colectivo de esta.

Una recomendación a futuro sería investigar sobre reconocimiento de planos y captura de movimiento, y ver cómo podría ser posible implementarlo en Unity, con el fin de colocar los planetas en el mundo real sin necesidad de marcadores. Otra recomendación, sería monitorear más de cerca el consumo de los recursos del dispositivo durante el desarrollo, ya que vimos que al estar en constante uso la cámara y

su funcionalidades de reconocimiento de marcas, el teléfono podría quedarse sin recursos más fácilmente.