

tf g

memoria

bellas artes

2018-2019



MENCIÓN: Artes visuales y diseño

TÍTULO: Newsgame Palmeral: Desarrollo de un videojuego con Unity.

ESTUDIANTE: Jaén Moreno, Francisco

DIRECTOR/A: Martínez Cano, Francisco Julián

PALABRAS CLAVE: Serious game, newsgame, videojuegos, ludificación, diseño.

RESUMEN: Esta propuesta de TFG propone el desarrollo con el motor de videojuegos Unity de un videojuego dentro del marco de los newsgames, de exploración y recolección de objetos, como herramienta principal para un test de eficiencia sobre los formatos informativos lúdicos.

Índice

pág/s.

1. Propuesta y Objetivos	1 - 1
2. Referentes	2 - 6
3. Justificación de la propuesta	7 - 12
4. Proceso de Producción	13 - 20
5. Resultados	21 - 23
6. Bibliografía	24 - 24

1. PROPUESTA Y OBJETIVOS

Este proyecto versa sobre la realización de un videojuego dentro del contexto de los *seriousgames*, en concreto de un *newsgame* en colaboración con la tesis doctoral de Alba García Ortega, doctoranda del programa de Doctorado en Ciencias Sociales y Jurídicas, del departamento de Ciencias Sociales y Humanas de la Universidad Miguel Hernández de Elche. Se trata una herramienta para llevar a cabo un test de usuarios para comprobar si los formatos informativos basados en pensamiento lúdico son eficaces, o si por el contrario los elementos gamificados distorsionan el mensaje. Con esta finalidad, para este trabajo de fin de grado se ha creado un *newsgame* utilizando el motor de videojuegos Unity, en el que se sitúa al jugador en dos escenarios libremente explorables e interactivables.

El juego consta de un menú básico, un escenario de la plaza de Elche que sirve como introducción y tutorial al jugador, y un primer nivel basado en el palmeral en donde el jugador es puesto a prueba tras haber aprendido los controles básicos.

1.1. Objetivos

A continuación se exponen los objetivos del TFG:

- Aprender a desarrollar un videojuego desde pre-producción hasta post-producción.
- Aprender a manejar la herramienta Unity y su lenguaje de programación (C#).
- Desarrollar un *newsgame* sobre una noticia, en este caso la plaga del picudo rojo en Elche.
- Crear una serie de mecánicas que le permitan al jugador transmitir la información del *newsgame*.
- Diseñar niveles de juego que resulten orgánicos e intuitivos para el usuario.
- Estudiar en profundidad los *seriousgames*, concretamente los *newsgames*.

2. REFERENTES

En este apartado se exponen aquellos referentes temáticos y estéticos que han enriquecido e influenciado la creación del presente trabajo de fin de grado.

2.1. Antonio Gonzalo Frasca (1972)

De origen uruguayo, catedrático de videojuegos en la Universidad ORT de Uruguay y co-fundador de Powerfull Robot Games. Actualmente, trabaja como diseñador jefe en DragonBox, un proyecto de educación para implementar App's de videojuegos en las escuelas para fomentar el aprendizaje en los niños.

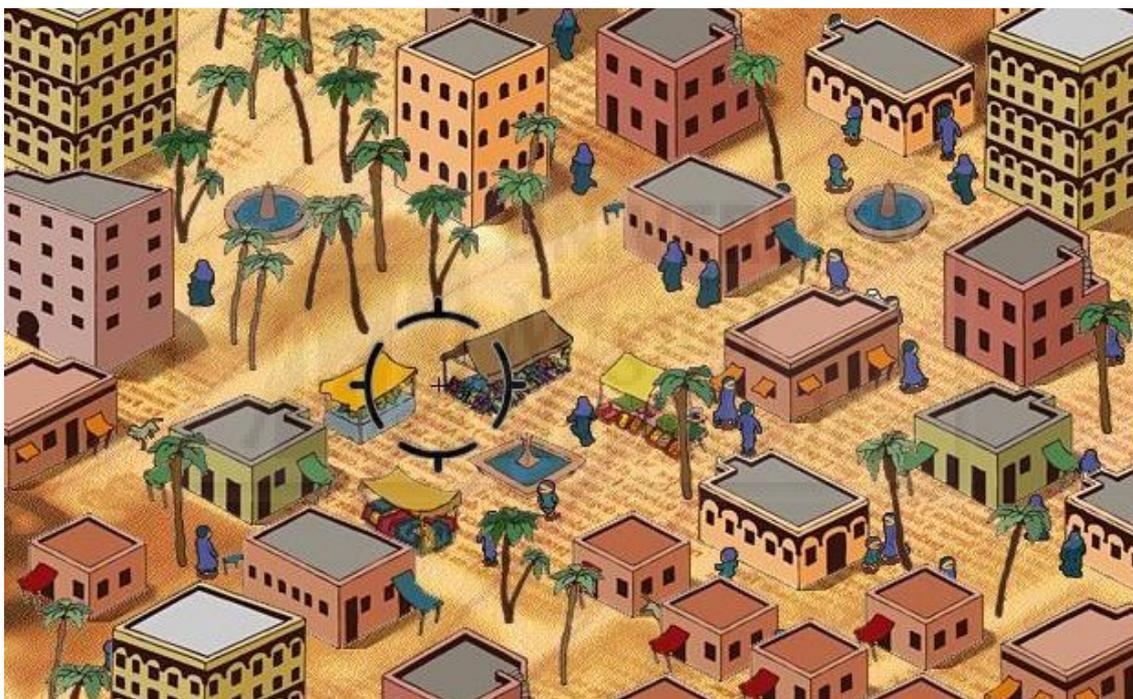


Fig. 1. Imagen del videojuego *September 12th* (Newsgaming, Gonzalo Frasca, 2003).
Fuente: <http://gamesforchange.org>. Url: <<https://bit.ly/2KHzwCo>> [consulta: 10/03/2019]

Se le considera el pionero de los *newsgames* por su juego *September 12th* (2003) (Fig.1), un producto videolúdico con fuertes connotaciones políticas y antibelicistas. Su principal meta siempre fue la de utilizar las narrativas lúdicas con fines educativos, considera que la ludificación, o la enseñanza a través del juego, es algo que existe desde la antigüedad, pero es hoy en día cuando se está empezando a investigar sobre este concepto, según indica Antonio Gonzalo Frasca, en su texto *Jugar en los jardines en miniatura* (2005):

[...] los japoneses pasaron siglos perfeccionando los sistemas para crear mundos alternativos. Y ahora tenemos testimonios en su extraordinaria capacidad de crear los mejores videojuegos y simulaciones del planeta (Gonzalo Frasca, 2015: 30).

Para él, los japoneses ya crearon sistemas virtuales en sus jardines Ginko, donde podían aprender y experimentar libremente. Es en esta idea del pequeño jardín donde se ha querido plasmar el concepto de este *newsgame*, ambos niveles del juego desarrollado son un pequeño campo de exploración donde el jugador deberá superar por su cuenta los niveles, aprendiendo a jugar por medio de su propia experimentación.

2.2. Espen J. Aarseth (1965)

Nacido en Noruega, doctorado en Comparativa Literaria en la Universidad de Bergen en Noruega y, el principal investigador en el Centro de Investigación de Juegos en la Universidad IT de Copenhague en la actualidad. Es conocido por hablar de la llamada literatura ergódica, término acuñado por él mismo, que en palabras del autor se define como “la literatura que requiere un esfuerzo no trivial por parte del lector para atravesar el texto” (Aarseth, 1997: 1).

A pesar de ser un término propio de la literatura, puede ser empleado perfectamente para hablar de videojuegos, pues en ambos casos ponen un reto u obstáculo al espectador para que pueda continuar. En el trabajo de fin de grado hay una narrativa ergódica a dos niveles; el primero serían los objetivos principales del juego, es decir, aquellas fases necesarias para superar el reto, y la segunda sería el acceso al *newsgame*, es decir, a los elementos periodísticos que están repartidos por el nivel del videojuego, siendo estos la verdadera sustancia del juego. No hay una derrota implícita, pero sí un desafío que hace emerger el pilar central de la experiencia lúdica.

2.3. Proteus (2013)

Videojuego creado por Ed Key y David Kanaga, al principio como un juego de rol similar a *The elder scrolls IV: Oblivion* (Bethesda Game Studios, 2006), pero cambiado posteriormente por un concepto, descrito por los autores, “menos violento y tradicional” (Fig. 2).



Fig 2. Videojuego Proteus (Key y Kanaga, 2013). Fuente: <https://www.pcgamer.com>. Url: <<https://bit.ly/2wPlo0z>> [consulta: 6/04/2019]

Proteus es un juego *first person* caracterizado por no tener historia ni personajes, de hecho no tiene objetivo. Los elementos que lo han convertido en un juego tan peculiar son su mundo 2.5D, con un *pixelart* plano y extremadamente simple, casi tan sencillo como la mecánica central: caminar. El único control que tiene el jugador es en su propio movimiento, no obstante esto no es ninguna limitación sino su fuerza, ya que el jugador está obligado a encontrar el sentido de la experiencia. Proteus está plagado de eventos como estaciones, climas, animales en el entorno y lugares de interés. No hay ninguna recompensa o logro por encontrarlos más que la satisfacción personal, esta idea tan poco convencional para un *gameplay* ha sido la que inspiró el diseño de la plaza y el palmeral que se muestran en este proyecto, los documentos, videos o infografías que el jugador encuentra no son necesarios para completarlo, no aportan nada y, sin embargo, son un eje fundamental de la experiencia puesto que de no tenerlos se perdería la faceta de *newsgame*, uno de los principales objetivos de esta propuesta.

2.4. Abzû (2016)

Juego desarrollado por Giant Liquid Studios, empresa fundada por los creadores de *Flower* (Thatgamecompany, 2009) y *Journey* (Thatgamecompany, 2012).

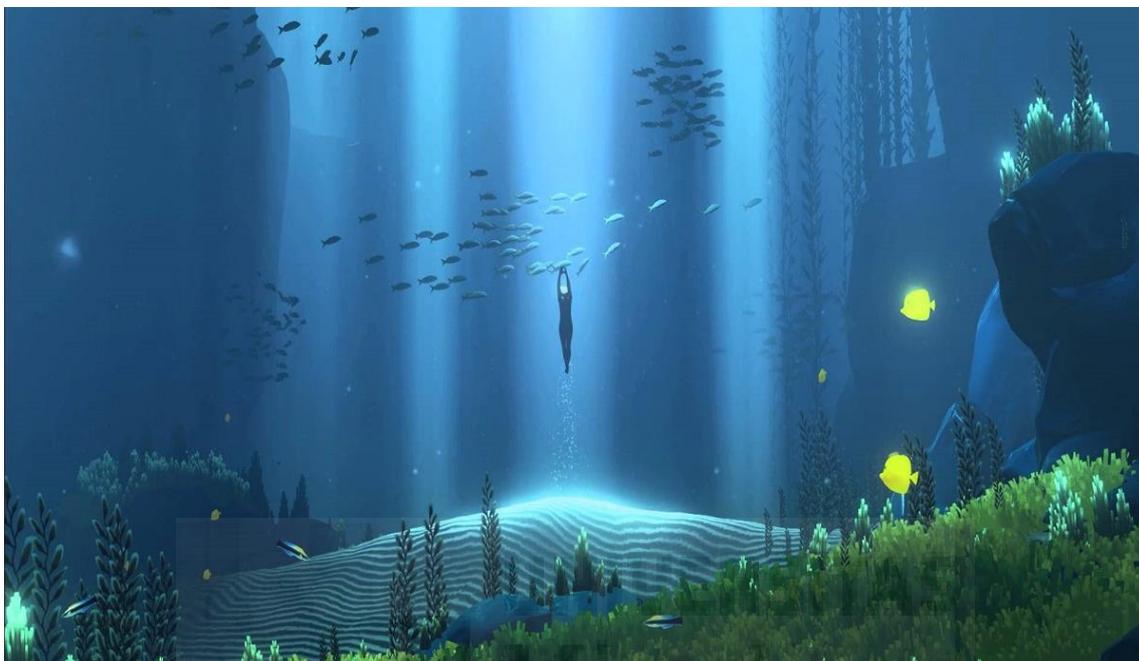


Fig 3. Videojuego *Abzû* (Giant Liquid Studios, 2016). Fuente: <https://www.20minutes.fr>. Url: <<https://bit.ly/2X8qogB>> [consulta: 13/05/2019]

Todos los juegos de estos desarrolladores tienen en común tres características: la soledad, la contemplación y la naturaleza. *Abzû* (Giant Liquid Studios, 2016), es el máximo exponente de estas características; el juego nos sitúa en la piel de un submarinista que debe explorar los distintos biomas marinos e interactuar con la fauna que encuentre. Este producto videolúdico, aun siendo corto y lineal, consigue embaucar gracias a su diseño visual minimalista, y cabe remarcar esta característica, puesto que gráficamente *Abzû* (Giant Liquid Studios, 2016) es extremadamente simple a nivel técnico. Todos los modelos son *low poly* y las texturas suelen tener un tono o dos, y aun así todo el entorno es diferenciable, estos colores han sido elegidos con una luminosidad y una saturación que dota a todo su universo de una vida y un encanto especial.

En el videojuego que hemos desarrollado, se ha tratado de transmitir esa misma simplicidad gráfica con escenarios y personajes poligonales dotados de colores claros y sencillos, que se diferencian claramente en el entorno. Hay muchos juegos en los que la paleta de colores se inclina hacia algunos tonos, normalmente marrones o

grises para dar realismo y seriedad, que acaban confundiendo al jugador por el parecido cromático entre escenarios, con este *newsgame* se quería diseñar niveles distinguibles y fáciles de navegar.



3. JUSTIFICACIÓN DE LA PROPUESTA

En este apartado exponemos las motivaciones de la propuesta y el estado actual de la industria del videojuego. Se hablará también de la situación de los *seriousgames*, centrándonos en los *newsgames*.

3.1. Estado del Arte

Los videojuegos son una forma de entretenimiento, que ha pasado de ser una sub-cultura a un titán económico que supera en ganancias a la industria musical y cinematográfica juntas. Por ejemplo, en 2013 con la salida de Grand Theft Auto V (Fig. 4) se batió el récord del producto de entretenimiento con más ventas en menos de 24 horas, de nuevo superando a cualquier competidor de otro sector de las industrias culturales; título que sigue manteniendo hoy en día.



Fig 4. Videojuego Grand Theft Auto V (Rockstar Games, 2013). Fuente: <https://www.gamespot.com>. Url: <<https://bit.ly/2Re004t>> [consulta: 2/06/2019]

En el Libro Blanco del desarrollo español de videojuegos (2016), España se posicionaba en 2016 como el cuarto mercado de Europa:

España, con una previsión de ingresos totales de 1.810 millones de dólares en 2016, se posiciona un año más como el cuarto mercado europeo – después de Alemania, Reino Unido y Francia – y como el octavo mundial, según datos de Newzoo (DEV, 2016).

Según el estudio elaborado por el portal web Clasf, España es uno de los países en los que más se juega a videojuegos, alrededor del 40% de la población está formada por jugadores y dedican de media 5,8 horas semanales a esta forma de ocio.

Sin embargo, según AEVI (Asociación Española de Videojuegos), estas ventas no reflejan la consideración que se tiene del medio como algo serio. Parece que existe una especie de barrera que impide ver a los videojuegos como una expresión cultural, que puede ser usada como herramienta de aprendizaje, más allá de la concepción habitual que se tiene de estos como simple entretenimiento.

Críticos como Robert Ebert han consolidado una serie de prejuicios a la industria, criticando que son obras meramente comerciales y sin ningún fin más que evadirse, o argumentando que no pueden ser elevados a la categoría de arte debido a que necesitan de una serie de decisiones, que el jugador toma y sobre las que el autor no tiene control, durante la experiencia del usuario.

No obstante, es especialmente en este último punto donde Ebert resulta desacertado a nuestro juicio. Primero, porque debido a la naturaleza de la computación, el juego nunca podrá ofrecer al jugador algo que no fuese programado; puede haber dinámicas emergentes que ni los autores llegasen a prever, pero es elección de los directores y diseñadores el hecho de que existan o no estas dinámicas. Y segundo, porque mucho antes de los videojuegos, ya existían obras ficticias que exploraban la capacidad de interacción con el receptor como la literatura del género *Choose your own adventure*, en la que el lector debe tomar las decisiones según transcurre la estructura narrativa del libro.

La interactividad es un elemento que siempre ha existido, pues se originó con los primeros juegos y juguetes que creó la humanidad. Pero existe un rechazo, o más bien un tabú, al concepto de juguete, señalado socialmente como infantil e improductivo. Es tarea de los jugadores y equipos de desarrollo romper este tabú, como menciona

Gonzalo Frasca en su artículo *Juego, videojuego y creación de sentido. Una introducción*: “El juego es la primera estrategia cognitiva del ser humano y como tal, una herramienta increíble para explicar y entender el mundo” (2009).

Una de las propuestas para consolidar el juego como elemento narrativo son los *seriousgames*, productos lúdicos que tienen más finalidades a parte del ocio y utilizan la gamificación como medio de expresión.

3.1.1. *Seriousgames*

Los juegos serios o formativos son un concepto relativamente nuevo. Se tratan de productos videolúdicos que promueven la construcción del conocimiento y el desarrollo del aprendizaje, trasladando al jugador a una simulación donde se le plantean situaciones, casos o problemas. Aunque el paradigma de un *seriousgame* sea algo reciente, la idea de utilizar los videojuegos como simulación con el objetivo de instruir al usuario parece venir ya desde el inicio de la *Magnabox* según Baer (1972), su creador.

La primera videoconsola de hogar, la Magnavox Odyssey (Ralph Baer, 1972), fue traída tanto con juegos de entretenimiento (Tennis, Haunted House, Roulette...) como con juegos educativos (Analogic, States, Simon Says...) [...] Baer rápidamente imagino aplicaciones serias de esta tecnología. Él probó su tecnología de pistola laser para diseñar juegos de disparo con motivos de entrenamiento bélico (Djaouti, Alvarez, Jessel, Rampanoux, 2011: 8).

Los *seriousgames*, aun teniendo la palabra serio en su nombre, no tienen por qué ser algo aburrido, al contrario, el hecho de que sean disfrutables es lo que hace que sean mucho más efectivos para aprender que los métodos tradicionales. Es un nuevo método de enseñanza basado en el aprendizaje a través del juego (*game-based learning*), algo que poco a poco se está implementando en escuelas, universidades y empresas.

Un ejemplo reciente es *Minecraft Education Edition* (Mojang, 2016) (Fig. 5), una versión de *Minecraft* (Markus Person, 2011), lanzada por Microsoft para impulsar el aprendizaje en materias CTIM (Ciencia, tecnología, ingeniería y matemáticas), que ha sido empleado en una universidad de Dallas, en Texas, según Franchescin (2016), para instruir a los alumnos y posteriormente someterlos a un examen sorpresa; el resultado fue que más de la mitad de los alumnos pudieron responder exitosamente sin haber estudiado previamente.



Fig 5.: Videojuego *Minecraft Education Edition* (Mojang, 2016). Fuente: <https://minecraft.makecode.com>. Url: <<https://bit.ly/2wNfPAY>> [consulta: 6/06/2019]

La principal diferencia entre el *game-based learning* y la enseñanza tradicional está en que permiten que personas avancen, fallen y aprendan a su propio ritmo, el juego forma parte de una motivación que hace que no lo vean como algo forzado, sino como una experiencia orgánica y entretenida. Es por esto que en los últimos años ha habido un crecimiento en la tendencia a la ludificación en actividades académicas o laborales gracias a estas nuevas formas de aprendizaje.

Ahora bien, no todos los *seriousgames* son iguales, existe una clasificación aún en proceso de consenso de los distintos tipos, recogida en el libro *Serious Games* (Abt, 1987):

- Advertgaming, videojuegos que se emplean como método de publicidad.
- Edutainment, programas que se utilizan para la educación.
- Edumarket Games, juegos serios que combinan varios aspectos.
- Newsgames, juegos periodísticos o de carácter editorial.
- Simuladores, juegos que generan diversas habilidades como la conducción o la aviación.

Debido a la naturaleza de este de fin de grado, nos centraremos únicamente en los *newsgames*.

3.1.2. *Newsgames*

También conocidos como juegos periodísticos, son una práctica reciente en la industria del videojuego y de la comunicación, donde la información se junta con el juego, normalmente con algún tema de actualidad. Supone, no sólo una nueva forma de ver los videojuegos, sino una revisión de cómo podemos percibir las noticias e implicarnos en ellas. Bogost los define de la siguiente manera:

El resultado no ofrece ni entretenimiento ni comentarios [...] pero sí saca provecho del interés público, ganando tráfico como resultado en los sitios [...] son tan fáciles de hacer que a menudo son mejorados y reinterpretados por otros diseñadores aficionados (Bogost, 2010:16).

El pionero que acuñó el término fue Antonio Gonzalo Frasca, que en su obra hacía referencia a las producciones lúdicas de las viñetas políticas, creador del primer videojuego con fines puramente periodísticos: *September 12th* (Gonzalo Frasca, 2003), una crítica a las políticas belicistas de EEUU. Pero antes de él ya hubieron antecesores que parecían estar imitando lo que en el futuro se conocería como *newsgames*, juegos como *Nuclear War* (New World Computing, 1898), *Balance of the power* (Chris Crawford, 1985) (Fig. 6) y *Hidden Agenda* (Trans Fiction Systems, 1988), todos con una fuerte implicación en hablar de los grandes miedos de la población del momento como eran la guerra fría o la posible guerra nuclear.

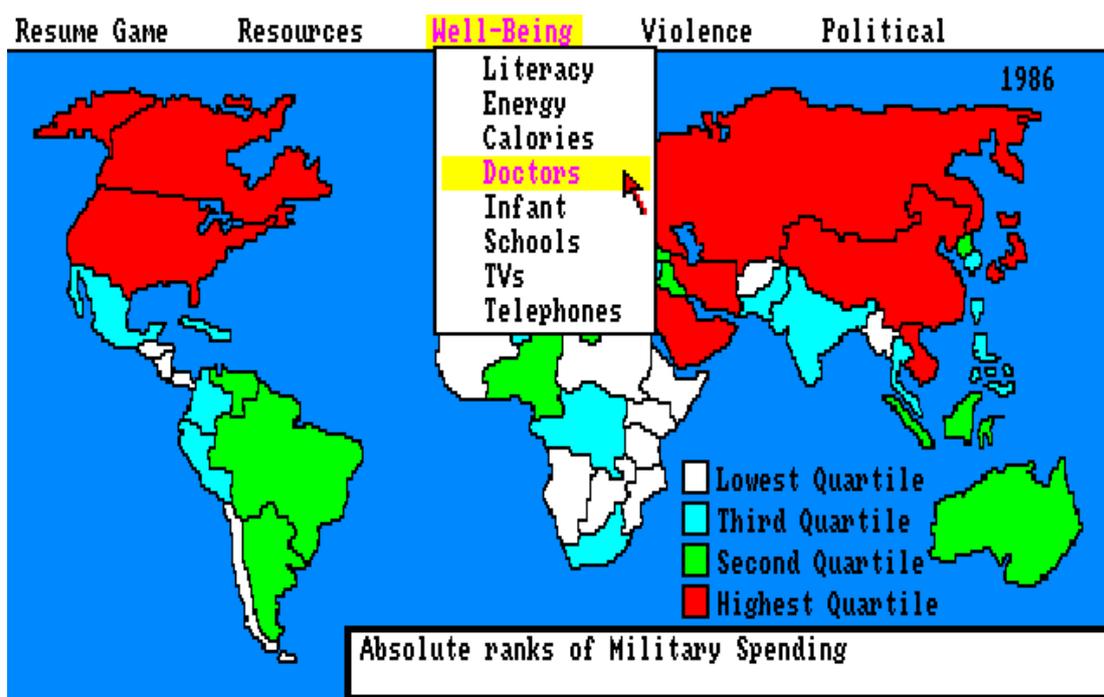


Fig 6. Videojuego *Balance of power* (Crawford, 1985). Fuente: <https://www.myabandonware.com>. Url: <<https://bit.ly/2F3JBFZ>> [consulta: 8/06/2019]

La esencia del *newsgame* es la misma que la del periodismo, hablar de aquello que está ocurriendo en el mundo en este mismo instante. Sin embargo, en el contexto actual globalizado, donde hay un problema de saturación de información y una cada vez más acentuada cultura del espectáculo, los receptores poco a poco han ido dejando de implicarse en aquello que contemplan, según el Libro blanco de la Información, el 56,8 % se informa a través de internet, pero el 57,1 y el 63,7 % de los usuarios sólo lee el titular y el encabezado respectivamente.

En esta situación, el *newsgame* podría ser una opción viable como herramienta de difusión, sin embargo existen varios problemas inherentes a la naturaleza del *newsgame*. Según afirma Cano (2016), los *newsgames* requieren tiempo y dinero, algo complicado de tener si estamos hablando de periodismo, y además de la brecha digital que impide a una parte de la población acceder a la tecnología y conseguir esta información. Pero añade que, como fortaleza, el *newsgame* cuenta con el poder de hacer que el usuario se identifique y forme parte de la obra, algo exclusivo del medio interactivo, haciendo que el *newsgame* sea único en las narrativas periodísticas.

4. PROCESO DE PRODUCCIÓN

En este apartado hablaremos de las distintas fases de la creación del proyecto, así como las herramientas que se han utilizado en el mismo.

4.1. Características del proyecto

4.1.1. Gráficos

Son modelos en 3D *low poly*; es decir, modelos en baja resolución, por lo que serán bastante poligonales y minimalistas, según se aprecia en la figura 7. Las texturas de dichos modelos serán planas, predominará uno o dos colores en un solo modelo. No se le ha aplicado ninguna clase de *shader*¹.



Fig 7. Fotograma del gameplay del videojuego *Newsgame Palmeral* (Jaén-Moreno, 2018).
Fuente: Elaboración propia.

4.1.2. Diseño de nivel

Se estructura en dos niveles: la plaza y el palmeral. El primero es, como su nombre indica, una plaza enorme (Fig. 8) donde el jugador puede explorar a su antojo. El área más grande del juego. Tiene forma radial y sitúa al jugador en el centro para darle varios caminos que puede seguir mientras recolecta los objetos clave. Dichos objetos

¹ Shader: Programa informático que controla la luz, sombra y color de una imagen.

están distribuidos de forma homogénea y acompañados de señales y otros objetos, que le darán información relevante sobre la noticia que trata este videojuego.

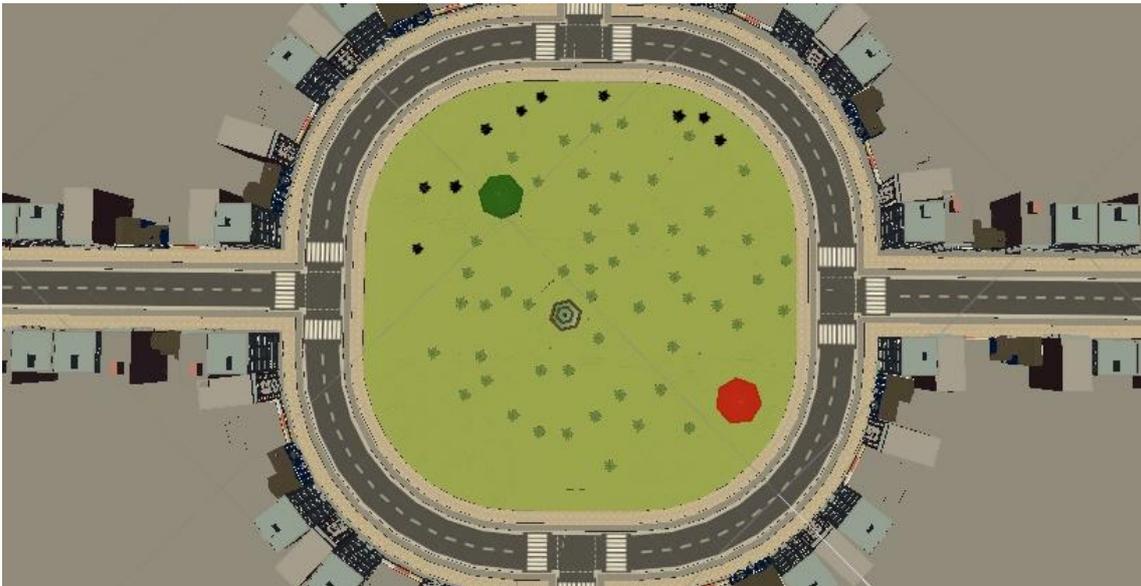


Fig 8. Vista cenital del primer nivel del videojuego *Newsgame Palmeral* (Jaén-Moreno, 2019). Fuente: Elaboración propia.

En comparación con la plaza, el palmeral (Fig. 9) es mucho más pequeño y lineal. Puesto que el jugador debe centrarse solo en los picudos, que estarán escondidos y deberá buscarlos minuciosamente. Para que el jugador no se frustrase en un área enorme se optó por algo más reducido en extensión y cerrado.



Fig 9. Vista cenital del segundo nivel del videojuego *Newsgame Palmeral* (Jaén-Moreno, 2019). Fuente: Elaboración propia.

4.1.3. Mecánicas

Las mecánicas que desarrolladas son las siguientes:

- **Movimiento:** El jugador puede moverse tanto en el eje X como en el Z con el teclado, para desplazarse en el eje Y puede saltar. Además la cámara se mueve junto al ratón.
- **Inventario:** El jugador a lo largo del nivel puede recolectar objetos principales y secundarios. Los primeros sirven para completar el nivel, mientras que los segundos le permiten adquirir información y conocimientos ajenos al juego, se tratan de documentos y vídeos que hablan del estado actual del palmeral de elche y su lucha contra el picudo rojo.
- **Inyección:** Mientras el jugador esté en el palmeral, contará con la jeringuilla, herramienta que le permitirá acabar con los picudos si está en un área cercana a su posición.

4.2. Plan de trabajo

El desarrollo comenzó a principio del segundo cuatrimestre del curso académico 2018-2019. Se hizo un modelo prematuro del picudo que se dejó aplazado con el fin de adelantar el trabajo en Unity todo lo posible. Se dedicaron los primeros meses al aprendizaje del programa, puesto que no se tenían conocimientos previos de su manejo.

El proyecto continuó con la creación del palmeral en un entorno 3D, junto con un personaje temporal como avatar del jugador para poder desplazarse por el nivel. Los *scripts* (código de programación) de movimiento y cámara fueron los propios de los *assets* que venían incorporados como prefabricados para el modelo del personaje (Fig. 10), y más tarde se implementó la plaza, nivel que sirve como tutorial e introducción a las mecánicas para el jugador. El funcionamiento del movimiento se basa en transformaciones del eje X y Z, pero además aplicándole una rotación al

personaje sobre su eje central (de forma que este siempre está mirando hacia la dirección que el jugador le indica.

```
// Get Input for axis
float h = Input.GetAxis("Horizontal");
float v = Input.GetAxis("Vertical");

// Calculate the forward vector
Vector3 camForward_Dir = Vector3.Scale(Camera.main.transform.forward, new Vector3(1, 0, 1)).normalized;
Vector3 move = v * camForward_Dir + h * Camera.main.transform.right;

if (move.magnitude > 1f) move.Normalize();

// Calculate the rotation for the player
move = transform.InverseTransformDirection(move);

// Get Euler angles
float turnAmount = Mathf.Atan2(move.x, move.z);

transform.Rotate(0, turnAmount * RotationSpeed * Time.deltaTime, 0);

if (_characterController.isGrounded)
{
    _moveDirection = transform.forward * move.magnitude;
    _moveDirection *= Speed;

    if (Input.GetButton("Jump"))
    {
        _animator.SetBool("is_in_air", true);
        _moveDirection.y = JumpSpeed;
    }
    else
    {
        _animator.SetBool("is_in_air", false);
        _animator.SetBool("run", move.magnitude > 0);
    }
}

_moveDirection.y -= Gravity * Time.deltaTime;

_characterController.Move(_moveDirection * Time.deltaTime);
```

Fig. 10. Captura de pantalla del código C# para el personaje. Fuente: Elaboración propia

Una vez terminados los escenarios, empezamos con el desarrollo de la parte de programación; se creó un sistema de inventario a partir de un script de terceros que fue modificado para adaptarse a las versiones más actuales de C# (Fig. 11). El inventario también se puede ocultar con la tecla Q, por si el jugador prefiere que no ocupe un lugar en la pantalla. El *script* del inventario funciona por medio de un HUD² en el que la base de datos sitúa los cálculos del ID del objeto y su cantidad al ser recogido. En el caso de los objetos clave, para superar el nivel, hay paralelamente un score que detecta cuántos de estos objetos tenemos (Fig. 11). En caso de tener el mínimo necesario, el juego aplicará un *LoadScene*³ para llevarnos al nivel del palmeral (Fig. 12).

² HUD: Head-Up Display. Es toda información que se muestra en pantalla, ya sea en imagen o en texto.

³ LoadScene: Función que ejecuta un cambio de un nivel a otro.

```

public class InventarioCollider : MonoBehaviour {
    InventarioManager m;
    public int scoreA;

    void Start()
    {
        m = GetComponent<InventarioManager>();
    }

    // Update is called once per frame
    void OnTriggerEnter (Collider col)
    {
        if (col.GetComponent<InventarioObjetoRecogible>() != null)
        {
            InventarioObjetoRecogible i = col.GetComponent<InventarioObjetoRecogible>();
            m.AgregarAlgoAlInventario(i.id, i.cantidad);
            scoreA = scoreA + 1;
            Destroy(col.gameObject);
        }
    }
}

```

Fig. 11. Captura de pantalla del código C# para el inventario. Fuente: Elaboración propia.

```

public class SceneSwitch : MonoBehaviour
{
    public int scoreB = 0;

    // Update is called once per frame
    void Update()
    {
        InventarioCollider score = GetComponent<InventarioCollider>();
        scoreB = score.scoreA;
        if(scoreB >= 3)
        {
            SceneManager.LoadScene("palmeral");
        }
    }
}

```

Fig. 12. Captura de pantalla del código C# para *LoadScene*. Fuente: Elaboración propia.

El segundo *script* añadido fue el sistema de interacción con objetos. A lo largo del nivel, el jugador se encontrará con señales y elementos secundarios que le proporcionarán información sobre el juego o del picudo rojo y la situación del palmeral en Elche. Se trata de una función que al estar dentro del área de colisión y apretar la tecla E, activa un *sprite* que muestra en pantalla la información, si el jugador vuelve a apretar la E, o si se sale del radio de colisión, el *sprite* es desactivado.

Para el nivel del palmeral se usó un *script* modificado a partir del de interacción con objetos. En vez de mostrar información en pantalla, lo que hace es eliminar a los picudos rojos. Sin embargo, el radio de colisión es mucho menor y estarán más escondidos, por lo que el jugador deberá buscarlos minuciosamente. Una vez el jugador los ha encontrado y ha acabado con todos, finaliza el juego (Fig. 13).

```
public class WinPalmeral : MonoBehaviour
{
    public int scoreB = 0;
    public GameObject felicidades;

    // Start is called before the first frame update
    void Start()
    {
        felicidades.SetActive(false);
    }

    // Update is called once per frame
    void Update()
    {
        KillPicudo score = GetComponent<KillPicudo>();
        scoreB = score.scoreA;
        if (scoreB >= 22)
        {
            felicidades.SetActive(true);
        }
    }
}
```

Fig 13. Captura de pantalla del código C# para la finalización del juego. Fuente: Elaboración propia.

Finalmente, se añadió el menú por el que se accede al videojuego. Consta de tres botones: uno para comenzar a jugar, otro para ver los controles, y otro para cerrar el juego. Asimismo pulsando la tecla Esc dentro del juego, se puede volver otra vez al menú.

4.3. Herramientas empleadas

4.3.1 Unity

Es el motor de videojuegos escogido para el desarrollo de nuestro *newsgame*. Unity fue creado por Unity Technologies, se lanzó por primera vez en Apple en el 2005 de forma exclusiva. Han ido adentrándose en otras plataformas y actualizando su software hasta llegar a la versión más reciente (Unity 2017).

Posee una gran polivalencia a la hora de manejar distintos formatos, tales como Blender, 3ds Max, Maya, Softimage, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks y Allegorithmic Substance.

Utiliza un sistema de objetos y *scripts* en C# que se ordenan por medio de un sistema de carpetas y otro de jerarquías, con los que se ordenan los objetos que se incluyan. Además, contiene varios *scripts* ya prefabricados para programar cuestiones generales (físicas, colisiones, controles, gravedad, etc). En caso de que aquello que el usuario busca no sea encontrado, cuenta también con una tienda donde la comunidad puede subir *assets* (elementos dentro del juego), de pago o gratuitos, para que terceros los puedan emplear. Estos *assets* pueden contener modelos 3D, animaciones, sonidos, música, interfaces, *scripts*, etc.

4.3.2. C Sharp (C#)

Como se mencionó en el anterior apartado, Unity emplea el lenguaje de programación C# para su funcionamiento. Es un lenguaje creado en 1999 por Microsoft .NET como una extensión de C. El signo de '#' viene del mismo modo que el '+' de C++, expresando un incremento.

Se asemeja a Java, ya que ambos funcionan por un sistema de objetos, la principal diferencia entre ambos es que Java no permite el uso de co-rutinas, además de ser más difícil de aprender puesto que es necesario aprender a usar herramientas de depuración y desarrollo de entornos. Es el sistema de programación con el que se han creado los *scripts* del *newsgame* en Unity.

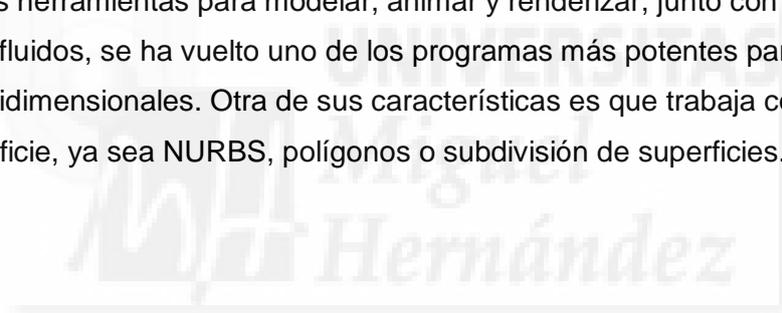
4.3.3. Microsoft Visual Studio

Es un entorno integrado (IDE) compatible con varios lenguajes de programación, como C++, C#, Visual Basic .NET, Java, Python, Ruby y PHP. La última versión y la más pulida es Visual Studio 2019, compatible con varias versiones de Windows y de frameworks de .NET. Este IDE es el empleado para el desarrollo del proyecto por su fácil manejo, buena visibilidad a la hora de leer el código y su herramienta de corrección.

4.3.4. Autodesk Maya

Un programa informático orientado a la creación y modelado de objetos 3D, efectos especiales y animación. Creado por Alias-Wavefront (posteriormente absorbido por Silicon Graphics). Es altamente personalizable en su interfaz, lo que le permite ser bastante compatible con programas de terceros.

Gracias a sus herramientas para modelar, animar y renderizar, junto con su capacidad para simular fluidos, se ha vuelto uno de los programas más potentes para el manejo de gráficos tridimensionales. Otra de sus características es que trabaja con cualquier tipo de superficie, ya sea NURBS, polígonos o subdivisión de superficies.



5. RESULTADOS

5.1. Post Mortem

Después de varios meses adquiriendo conocimientos sobre las diferentes herramientas, para el desarrollo del videojuego *Newsgame Palmeral* (2019) y poniéndolos a prueba, se ha conseguido comprender su funcionamiento a un buen nivel. También se ha logrado crear un juego que funcione correctamente y al mismo tiempo consiga comunicar un mensaje periodístico al usuario.

Se han solucionado diversos problemas, desde errores de programación como la imposibilidad de añadir objetos al inventario, o menús que no aparecían al pulsar su correspondiente tecla, hasta problemas de compatibilidad con programas externos o fallos en el HUD con la resolución de pantalla.

También hubo varias ideas desechadas por no encajar en la temática (poder coger objetos con el ratón), o por no tener tiempo/habilidad para implementarlas (un sistema que conforme avanza el tiempo en el palmeral se fuese degradando mientras se destruían las palmeras). También se pensó en incluir un tutorial en el que se mostrase al jugador todos los controles, sin embargo se descartó y se incluyó una nota de los controles en el menú de inicio. Finalmente, el descarte de estas ideas fue más una ayuda que un obstáculo, ya que con el tiempo se vio que estas mecánicas no eran tan importantes para que el videojuego mantuviese su esencia, en cambio fue totalmente necesario implementar el inventario y el sistema de interacción con objetos, para hacer que el juego fuese una experiencia más inmersiva.

En cuanto a los desafíos encontrados, el mayor de estos fue ausencia por completo de conocimientos de Unity y C#, con lo que se necesitó mucho más tiempo para investigar, reduciendo el tiempo de diseño y otras tareas. En la misma línea, fue compleja la creación de este videojuego, puesto que programarlo desde cero en solo un semestre era una tarea titánica, por lo que se optó por utilizar algunos recursos gratuitos para agilizar el proceso.

5.2. Competencias adquiridas

Una vez se ha conseguido crear un videojuego desde cero, se tiene una percepción más amplia del largo y costoso proceso de desarrollo. Es un trabajo que requiere la mezcla de habilidades de distintas áreas de conocimiento, incluso sin ser un trabajo de equipo, era necesario una precisa organización interna y administración de cada aspecto del proyecto.

Se ha mejorado la capacidad de manejo de lenguajes de programación, algo completamente nuevo, como puede ser C#, para poder crear entornos virtuales dinámicos e interactivos. Al principio, el escribir una sola línea podía llevar a que gran parte del proyecto dejase de funcionar, y puesto que funciones como mover al personaje o la cámara son fáciles de encontrar en la *asset store* que proporciona Unity, no era necesario programarlos; pero llegaron cuestiones mucho más concretas para las que no había una vía fácil, urgía empezar a programar siguiendo tutoriales, primero de webs españolas y posteriormente de fuentes internacionales, mucho más especializadas y actualizadas las segundas que las primeras.

Pero ante todo se han desarrollado facultades para el aprendizaje autónomo, la búsqueda de información y el criterio para la selección de la misma. Hay incontables páginas en Internet y no siempre los primeros resultados van a ser los que den la información precisa, se requieren horas para poder encontrar la solución a un problema. Y como se mencionó anteriormente, solo se encontraban en inglés los recursos correctos, por lo que el manejo del idioma era estrictamente necesario.

5.3. Conclusión

Antes de comenzar con el desarrollo, se tenían multitud de ideas y mecánicas que implementar en el juego. Descubrir que el proyecto era más complejo de lo que en un principio parecía fue un enorme desafío, que llevó a que el trabajo fuese ralentizado debido al proceso previo de aprendizaje de las herramientas.

Sin embargo, es en esta dificultad en donde surge el verdadero incentivo para llevar a término el proyecto. Lejos de querer dejarlo a medias, esta cuestión motivó el querer seguir adelante, enfrentándonos a un trabajo que requiere una buena administración, concentración e inversión del tiempo.

Pese a todo, el resultado final es satisfactorio, es decir, no tiene todas las mecánicas que fueron planeadas en un inicio, pero sí logra los objetivos expuestos. Se trata de un *newsgame* que habla sobre un problema de actualidad y propone soluciones para el mismo, de forma que cualquier persona puede acceder a la información sobre esta cuestión a través de este *newsgame*.

En definitiva, a pesar de todos los obstáculos, se ha conseguido finalizar el trabajo de fin de grado y cumplir con los objetivos planteados. Con todo lo aprendido, se abre todo un horizonte para la creación de próximos proyectos en el ámbito del desarrollo de productos videolúdicos.

Enlace a la descarga del videojuego:

<https://drive.google.com/open?id=1SBmnc6W9wtRYpNzCTrhTBRcv4umcu-FH>

6. BIBLIOGRAFÍA

Aarseth, J. E. (1997). *Cybertext: Perspectives on Ergodic Literature*. EEUU: John Hopkins UP.

Abt, C. C. (1987). *Serious Games*. EEUU: Viking Press.

Asociación de Medios de Información (2017). *Libro blanco de la Información 2017*. Madrid: Ami.

Asociación Española de Empresas Desarrolladoras de Videojuegos y Software de Entretenimiento (2016). *Libro blanco del desarrollo español de videojuegos 2016*. Madrid: ICEX

Bogost, I; Ferrari, S; y Scheweizer, B. (2010). *Newsgames: Journalism at Play*. Cambridge: MIT Press.

Cano, F. J. M. (2016). "Newsgames: Nuevas Tendencias en el Periodismo Ibérico". En *CEUR Workshop Proceedings. CoSECivi 2016*. Vol-1682 (págs 159-169).

Clasf (2016) "Videojuegos en España en 2016" [En línea] <https://www.clasf.es/noticias/pics/2017/02/estudio-clasf-espana-videojuegos-2016.pdf> [última consulta 2/6/20]

Djaouti, D; Alvarez, J; Jessel, J, P; Rampnoux, O. (2011). *Origins of Serious Games*. UK: Springer London.

Franchescin, T. (2016) "Una expansión de Minecraft demuestra que puede sustituir un curso universitario" [En línea] <http://edu4.me/una-expansion-de-minecraft-demuestra-que-un-juego-puede-reemplazar-a-un-curso-universitario/> [última consulta 21/5/2019]

Gonzalo Frasca, A (2005). "Jugar en los jardines en miniatura". Revista: *Cluster Magazine*. Nº V (págs 30-34).

Gonzalo Frasca, A (2009) "Juego, videojuego y creación de sentido. Introducción". Revista: *Comunicación*. Nº VII (págs. 37-44).

King, J (2018) "GTA 5 is now the most profitable entertainment product ever" [En línea] <https://www.trustedreviews.com/news/gta-5-now-profitable-entertainment-product-ever-3448487> [última consulta 4/6/2019]