

LAS CONSECUENCIAS DEL *GLITCH* EN EL ENTORNO VIRTUAL INTERACTIVO

Patricia S. Gracia Artero ^[1]

¹ Universidad Complutense de Madrid, Máster en Investigación en Arte y Creación, Madrid, España

patriciagraciartero@gmail.com

Resumen. Este documento examina las posibilidades del *glitch*, o fallos de software, dentro del entorno virtual, con el objetivo de analizar las consecuencias que acarrea el uso de estos efectos e intentar, por otra parte, rescatar resultados estéticamente relevantes. Para eso se recogen distintas definiciones del *glitch*, así cómo se estudian y valoran como son aplicados estos mismos efectos en distintos videojuegos. Se completa la investigación con un ejercicio práctico en dónde se aplican los efectos que se estudian para observar de manera más controlada sus resultados en la experiencia visual, aunque también se valoran en menor medida las consecuencias en la experiencia narrativa y en la jugabilidad. La investigación se beneficia de las metodologías *art-based research* y la programación extrema que se coordinan entre sí. Con todo esto, se ha podido concluir de manera resumida que, a través del proceso de construcción de espacios y aplicación de efectos *glitch*, encontramos distintos recursos que pueden ser útiles por su valor formal y cromático. Pueden rescatarse fragmentos de imagen que tienen interés, ya que proporcionan nuevas formas y representaciones debido a su carácter parcialmente aleatorio. También se puede concluir que no todos los *glitches* que se aplican a estas experiencias pueden dar como resultado imágenes o experiencias de calidad, y que no se puede pretender tampoco que en el *glitch* no controlado proporcione una experiencia enriquecedora.

Abstract. This document examines the possibilities of glitches, or software failures, within the virtual environment, with the aim of analysing the consequences of using these effects and attempting, on the other hand, to rescue aesthetically relevant results. To do so, different definitions of glitch are included, as well as how these same effects are studied and evaluated in different video games. The research is completed with a practical exercise in which the effects studied are applied in order to observe their results in a more controlled way in the visual experience, but also the consequences in the narrative experience and in the gameplay are considered in a lesser extent. The research benefits from art-based research methodologies and extreme programming that are coordinated with each other. With all this, it has been possible to conclude in a summarised way that, through the process of construction of spaces and application of glitch effects, we find different resources that can be useful for their formal and chromatic value. Fragments of images can be rescued that are of interest, since they provide new forms and representations due to their partially random char-

acter. It can also be concluded that not all glitches applied to these experiences can result in quality images or experiences, and that it cannot be claimed either that the uncontrolled glitch provides an enriching experience.

Palabras clave: videojuego, *glitch*, game-art, *glitch-art*, entorno interactivo

1 Introducción

Este documento recoge el prototipo experimental que se ha llevado a cabo para probar en la práctica los efectos *glitch* en el entorno interactivo virtual. El objetivo es averiguar el resultado, principalmente visual, aunque también narrativo y jugable, de aplicar diversos valores de distorsión sobre la experiencia de juego. Para ello, se ha hecho una investigación recorriendo diversas aproximaciones a la definición de *glitch*, abordándolo a través de algunos videojuegos que aprovechan al máximo estas características del error.

1.1 Definiciones del *glitch art*

En términos generales, se denomina *glitch* a todo efecto producido por el malfuncionamiento imprevisto de alguno de sus componentes en su sistema informático. En su forma original, estos efectos son aleatorios, intermitentes y, en ocasiones, de escasa duración. Son elementos, en primera instancia, indeseables, que condenan a un dispositivo a ser arreglado o cambiado. Dentro del mundo del arte pueden apreciarse influencias por parte de los movimientos Dadá y Fluxus en el movimiento *glitch-art*, los cuales tenían muy presente en sus actividades el juego, el azar, el accidente y la interacción, así como la experimentación con los límites del arte del momento. El *glitch-art* se caracteriza por aprovechar o provocar estos fallos digitales para sacar de ellos nuevas y, en ocasiones, poderosas imágenes. Varios teóricos han evaluado el *glitch* visual y propuesto sus propias clasificaciones.

Iman Moradi, artista y teórico *glitch*, lleva investigando en el medio desde principios de la década de los 2000. En su texto *Glitch Aesthetics* de 2004 [1], Moradi diferencia entre *pure glitch* y *glitch-alike*. Siendo el primero el *glitch* que se encuentra arbitrariamente, que no es producido o manipulado y que es realmente originario de un error de la máquina, efímero. Todo lo contrario al *glitch-alike* que se define como un *glitch* fabricado, artificial y controlado.

En 2010, Rosa Menkman trató el tema con su entrada de blog *glitches vs. glitch art* [2]. En el texto afirma que el *glitch-art*, como forma de expresión, está siendo tratado de múltiples maneras fuera de su convencionalismo debido a los múltiples actores que participan en él. Y defiende que esto crea glitches domesticados, artistas que se centran más en el diseño de éste, que en el proceso de ruptura.

Menkman, más adelante, en su texto *The Glitch Momentum* de 2011 [3], critica esta categorización entre *pure glitch* y *glitch-alike* y considera que, aunque a priori pueda parecer un punto de partida útil, se queda muy en la superficie. Defiende que esta división binaria atenta contra las pretensiones que tiene el género. Menkman tiene como principal pretensión romper estas categorías y con ello ver qué hay entre-medias, en los entresijos de los sistemas y ordenes establecidos. También hace una crítica a la visión puramente tecnológica que posee Moradi. Menkman dice que estos fallos no son únicamente digitales, sino que tienen un gran componente ideológico y estético. Defiende que es más interesante tratar el arte que se produce con el *glitch*, el *glitch-art*, dentro de todo un sistema cultural.

Tratando el concepto de *glitch*, Curt Cloninger (2010), en su texto *GlitchLinguistx: The Machine in the Ghost / Static Trapped in Mouths* [4] hace una interesante reflexión sobre la capacidad de elección del artista a la hora de recoger cual es el fenómeno más interesante. Diferencia la señal y el ruido. Y divide el ruido en dos categorías: el que es relevante, que merece la pena preservar, recolectar y compartir, y el que se edita o que se ignora. De esta manera salen las dudas. ¿Bajo qué criterio se evalúan estos fallos? ¿Qué sistemas usar para no elegir siempre el mismo *glitch*?

Otra manera de acercarse al *glitch* viene por parte de Kim Cascone. En su texto *Errormancy: glitch as divination* [5], Cascone (2011) relaciona el acto de la adivinación con el *glitch*. La adivinación era considerada una puerta para asomarse al reino del más allá, el marco espiritual y la comunicación astral. Se solía consultar a través de esta típica bola de cristal que hace de conector. Relaciona el *glitch* con esto comentando que de alguna manera es un dispositivo de ruptura en un sistema que nos permite observar las grietas, el espacio invisible por el observador.

Y finalmente, lo más reciente sobre el tema es lo dicho por Daniel Temkin en 2014 con *Glitch & Human/Computer Interaction* [6], con “There’s not much ‘glitch’ in glitch art”. Temkin comenta que el *glitch-art* ha perdido su esencia. Ahora hay más una colaboración con la máquina que un verdadero riesgo a que estas prácticas provoquen daños en el sistema, como originalmente provocaban. A pesar de que admite que aún hay artistas que siguen trabajando con *bugs* (errores de software), defiende que la mayoría de los trabajos podrían meterse dentro de la misma bolsa, proyectos donde se modifican los algoritmos o los códigos y se utilizan de manera imprevista. Algo así como proyectos sobre seguro. Pone el ejemplo de cuando se edita el contenido de una imagen JPG, cuando esta se abre en un programa que no es el suyo determinado, como el bloc de notas,

y se edita. Dice que es una manera habitual de trabajar. Esta práctica es un claro ejemplo de cuando no hay riesgo, estos archivos no suponen una amenaza, solo lo son si no cumplen con las expectativas estéticas.

La tendencia actual en el *glitch-art* es de ordenar la variedad entre: qué es lo más *glitch*, y qué se aleja más del origen para llevarlo así a una estética con posiblemente más potencial, más personalización y menos azar.

1.2 Videojuegos apropiándose de la estética *glitch*

Actualmente hay diversos títulos que aprovechan las posibilidades de este tipo de fallo para fortalecer sus mensajes. Éstos serían ejemplos de *glitches* fabricados con todas las intenciones de crear esas experiencias tan poco deseadas de manera nativa y ahora valoradas por su estética.

Assassin's Creed [7] es una saga que en sus introducciones y cinemáticas ha usado mucho los efectos *glitch*, ya que parte de su trama implica tecnología muy avanzada y viajes temporales. Un ejemplo muy vistoso es cuando el personaje pierde vida y la interfaz se ve modificada por efectos de distorsión. Estos efectos le hacen recordar al jugador que este personaje que está manejando se encuentra sometido en una simulación que está viviendo otro personaje dentro de una máquina. Lo cual no es solo efecto visual, si no también narrativo. Cabe mencionarlo para saber diferenciar las obras que hacen un uso más insípido del efecto de las que se comprometen con la estética y el concepto.

Pac-Man 256 [8], un videojuego originalmente de móviles de 2015 basado en el mítico *Pac-Man* [9], pero más específicamente en *glitch* que se daba cuando se llegaba al nivel 256. Este *glitch* afectaba a la memoria del juego, lo que provocaba un efecto visual que claramente rompía la experiencia de juego, haciendo de la actividad una misión imposible y totalmente injusta. Los personajes funcionan correctamente, y aunque falten algunos *power-ups* (potenciadores), es posible moverse. Sin embargo, la mitad derecha del mapa está totalmente corrupta con letras y símbolos entremezclados, ya no hay caminos visibles, aunque aún sean transitables. Lo interesante de esto es ese error. *Pac-Man 256* es en sí un nivel infinito, vas avanzando hacia arriba dentro de este laberinto tan característico. Lo destacable está en que, conforme se avanza, debajo del mapa una masa de números y letras desordenados se apodera del mapa lentamente, de manera que si vas lento y acabas en la parte baja del mapa, esta masa te come y, por tanto, es fin de juego. Esta masa de números y letras hace referencia a este famoso *glitch*.

Pony Island [10] es otro de los juegos sorprendentes que aplican todas las herramientas a su alcance para llevar al jugador de la mano por un camino de una oscura temática. En resumen, el videojuego trata de una recreativa a la que el protagonista se enfrenta. Esta recreativa está repleta de errores gráficos justificados por una corrupción diabólica. Esta entidad tratará de robar el alma del protagonista como lo ha hecho con los anteriores jugadores. Con respecto a las mecánicas, para avanzar por el juego hay que solucionar “errores” del código de programación del juego, o más bien, una simulación de él. Y todo esto no solo crean una estética atrayente y provocativa, sino que, están por un motivo justificado en su historia.

Doki Doki Literature Club! [11] es el nombre de una novela visual interactiva dentro del ámbito de los videojuegos y su temática es de terror psicológico. Lo interesante de este título es que, dentro de su simplicidad mecánica basada en elegir líneas de diálogo

go, es capaz de transmitir sensaciones completamente extremas de incomodidad, tensión y terror. Esto es posible, entre otros aspectos, por el buen uso de los efectos de distorsión y corrupción de la imagen. Aparte de los efectos visuales, el guion es impecable, llegando a romper en varias ocasiones la cuarta pared. El juego interpela al jugador como, por ejemplo, editando los archivos del ordenador y dirigiendo los diálogos directamente al mismo, haciendo así una experiencia más personal.

Memory of a Broken Dimension [12] es un videojuego que actualmente sigue en proceso de desarrollo y, aunque tiene una versión reducida para probarlo, puede cambiar radicalmente. Este título destaca especialmente en el apartado visual. Recrea en un espacio virtual con unos escenarios casi imposibles, tanto de navegar como de ver. A todo hay aplicado un efecto comparable al *datamoshing* y eso hace que el jugador tenga que mantenerse quieto para no distorsionar la imagen. Sin duda, será de esas experiencias necesarias experimentar cuando esté acabado para poder hablar de este tema de manera actualizada.

Los videojuegos que se apropian de la estética *glitch* pueden analizarse de una manera clara, ya que son proyectos pensados desde esa perspectiva. Por ejemplo, en *Pony Island*, han justificado los *glitches* con una historia sobre invocaciones demoniacas y eso convierte al *glitch* en una herramienta esencial para el buen funcionamiento de la narrativa. También es esencial el *glitch* en *Pac-Man 256* para su jugabilidad, igual que los *glitches* no modificados son límites en la experiencia de juego, en este caso los *glitches* creados para *Pac-Man 256* son la motivación y el atractivo del mismo. El efecto *glitch* no es solo efectivo en el apartado visual, sino también en los demás apartados que conforman la obra global.

2 “En estas cuatro paredes (In these four walls)”, entorno interactivo de experimentación con efectos *glitch*

La propuesta es crear un entorno virtual interactivo en 3D y en tercera persona con el que comprobar las consecuencias y las posibilidades de los *glitches*. La finalidad no es crear exactamente un videojuego, ya que varios componentes que suelen configurarlo aquí no se encuentran, sino que es crear una experiencia navegable que permita evaluar estos efectos en un entorno de juego. Para conseguirlo, se crea un escenario base y que, a través de portales, se pueda navegar por otros mapas. Estos mapas son iguales al primero pero cada uno tiene aplicado un efecto *glitch* diferente.

2.1 Desarrollo

Al personaje se le han aplicado los controles clásicos de movimiento en primera persona. Con WASD o las teclas de dirección se toma el control del rumbo del personaje, con la barra espaciadora se permite el salto, con el ratón o en su defecto el *trackpad* o panel táctil se controla dirección de la cámara del personaje y finalmente con la techa ESC o escape se accede al menú de pausa. [13]. Para el modelo estético se ha optado

por el estilo gráfico de *low poly*. Se ha elegido este modelo por su relativa sencillez tanto en la forma como en su construcción. Que sea sencillo en la forma es bueno para el proyecto, ya que no son objetos que llamen mucho la atención ni que lleven todo el protagonismo, sino que se encajan en el escenario y dejan a los efectos actuar. El escenario es de forma rectangular y se compone de diversos modelos en 3D que simulan una zona concreta de una aldea pequeña. Esta aldea está incluida en el interior de una esfera que simula el espacio alrededor. Para moverse entre los escenarios hay varias opciones, está la navegación desde el menú de pausa, desde el menú principal, o bien, desde los portales de dentro del juego. Estos portales se activan una vez se haya interactuado con la esfera. Hay una pieza flotante por escenario y el objetivo es encontrar esta pieza para que se abra el portal y así desplazarse al siguiente escenario. El motivo de tener que buscar esta pieza es animar al participante a navegar y explorar el escenario antes de pasar al siguiente.

Escenarios

El escenario base es el primero con el que se encuentra el usuario nada más darle a empezar. Este no tiene ningún efecto aplicado. Aquí se podrá explorar el mapa antes de someterse este a transformaciones. Los demás escenarios serán duplicados de éste.

Para la primera escena se ha querido introducir al *glitch* de manera suave. El *script GlitchEffect.cs* [14] permite controlar la intensidad del ruido, de las interferencias y del color, al igual que te da la opción de poner una imagen como *Displacement Maps* o mapeado por desplazamiento. (Ver Fig. 1)

AnalogGlitch.cs y *DigitalGlitch.cs* [15] son los *scripts* que modifican el apartado visual de la segunda escena. El primero imita los errores que se daban en los dispositivos más anticuados como las televisiones analógicas. Dispone de varios atributos que se pueden variar como las líneas verticales y horizontales, la distorsión en los canales de color, etc. El segundo efecto, el digital, solo tiene la variable de intensidad y provoca estas líneas horizontales como si grandes píxeles se estuvieran comiendo la imagen. (Ver Fig. 2)

En la tercera escena se juega con el movimiento de la cámara. La cámara está pensada para que sea estática y que solo se mueve por las acción del participante. Sin embargo, en esta ocasión se la ha provisto de un *script* llamado *JitterMotion.cs* [15] que variando sus valores se puede hacer que la cámara pierda de alguna manera su lógica, provocando así que su movimiento se vuelva aleatorio y errático. (Ver Fig. 3)

El cuarto escenario se caracteriza por poseer texturas *GIF* sobre los objetos. Se diferencia de los demás por no ser un efecto aplicado sobre la cámara, si no por estar aplicado sobre todos los objetos que forman parte de la escena. Para la realización se ha necesitado proceder a la división por fotogramas de todos los *GIFs* usados [16]. El *script* de inscribe dentro del componente en el que se desea que el *glitch* se represente. (Ver Fig. 4)

En la quinta escena se manifiestan efectos de deformación de los objetos. Aquí se ha hecho uso de un *script* llamado *TwistDeformer.cs* [17] que, según como se configuren las propiedades del efecto, crea una distorsión u otra en la figura. Lo que se ha hecho ha sido dejar el efecto desactivado y hacer que se active cuando el personaje colisione con un área invisible alrededor de ese objeto. De esta manera, mientras el participante busca la bola para activar el portal, se va metiendo, sin él saberlo, en las áreas de colisión. Este efecto tiene varias ramificaciones. Por una parte, el más visual de deformación, pero también modifica el movimiento del personaje. El personaje dentro de las zonas de colisión tiene la habilidad de salto alterada, le permite tener una altura mayor de lo programado. Esto, si se descubre, altera la percepción sobre el mapa también. Cuando el objeto está modificado crea nuevos caminos y nuevas zonas inaccesibles y bloqueadas, así como puede darse la casualidad de que el personaje se quede atrapado entre objetos y tenga que reiniciar el nivel. (Ver Fig. 5)

2.2 Resultado

El resultado puede experimentarse a través de este enlace ya que, debido a su naturaleza, el proyecto está alojado en la web. [18]



Fig. 1. Escenario 1

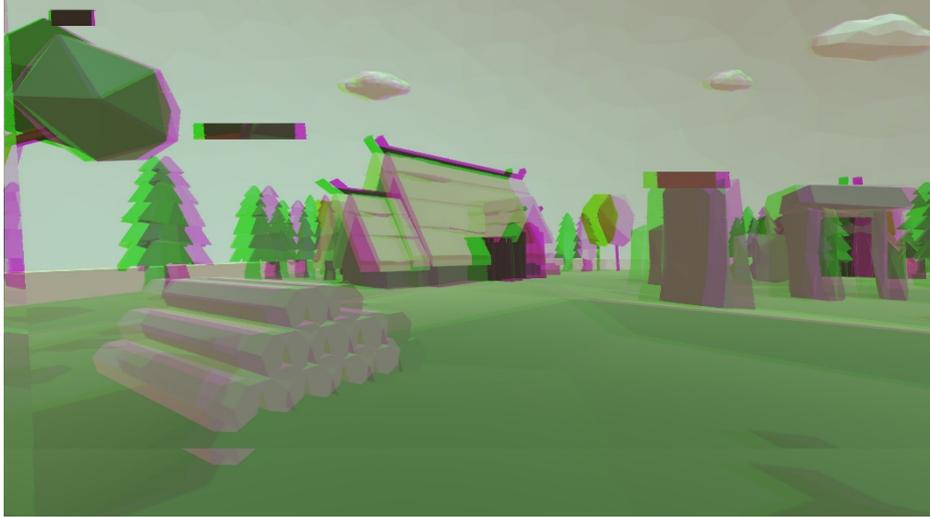


Fig. 2. Escenario 2

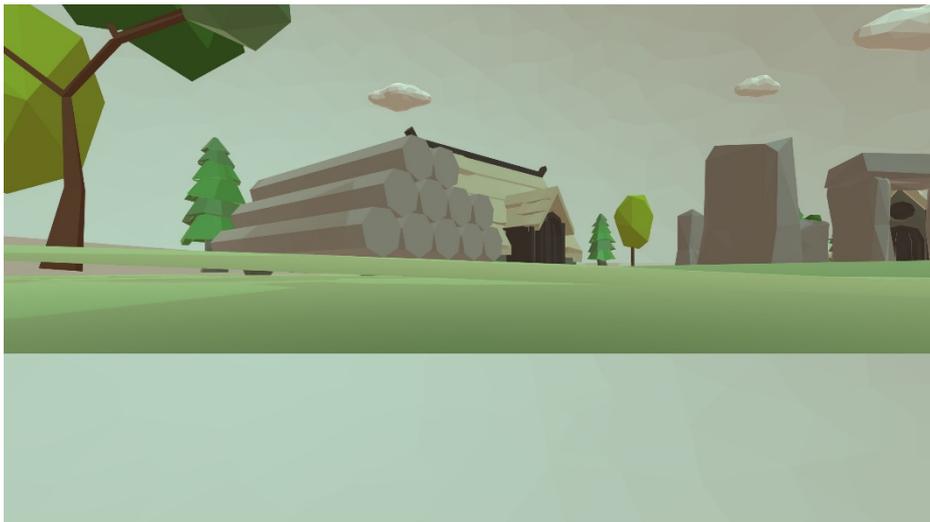


Fig. 3. Escenario 3



Fig. 4. Escenario 4



Fig. 5. Escenario 5

3 Conclusiones

A la hora de recoger las diferentes posiciones de los artistas y teóricos del arte con respecto a la función del efecto *glitch* en el entorno artístico, se ha podido observar que no hay un acuerdo uniforme. Se puede concretar que el *glitch* es una herramienta que cuando está tratada por artistas de los nuevos medios, en términos generales, los resultados que trae son innovadores y, en las discusiones deja de tener relevancia comprobar si está o no provocado el efecto *glitch* que se encuentra. Habiendo asimilado lo que puede aportar el *glitch* a las experiencias jugables, varios videojuegos han incluido este efecto en sus productos. Varias experiencias no solo se ven reforzadas en su apartado visual, que es donde más claramente destacan, sino que también se ha comprobado que añaden más profundidad narrativa y jugable.

Finalmente, con la realización del proyecto y las pruebas sobre él se puede concluir que:

- No todos los efectos aplicables a estas experiencias aportan calidad visual, varios de ellos se limitan a meter ruido por lo que hay que ser consciente de ello y controlarlo.
- Aquí el *glitch* no tratado no puede funcionar. Al ser todo en conjunto una experiencia controlada los parámetros tiene que ser conocidos y fijados.
- Hay ocasiones en las que estos efectos componen imágenes que, sacadas de su fuente de origen, pueden constituir un proyecto por sí mismo.
- Aunque la jugabilidad no es un elemento característico del proyecto, sí se puede observar que estos efectos alteran la forma en la que el participante interactúa con el espacio construido.

Este documento ha abierto la posibilidad de investigar tanto teórica como prácticamente el caso de estudio. Las limitaciones que han ido apareciendo por el camino, las conclusiones que han sido desveladas y las potenciales líneas de desarrollo paralelas que han ido surgiendo, han revelado la necesidad de profundizar en esta relación entre videojuego y *glitch* en un futuro.

Referencias

1. Morandi, I. (2004). *Glitch Aesthetics*. Recuperado el 2 de febrero, 2020, de <http://www.organised.info/wp-content/uploads/2016/08/Moradi-Iman-2004-Glitch-Aesthetics.pdf>
2. Menkman, R. (2010). *Glitches vs. Glitch-art* [Publicación en un blog]. Recuperado el 23 diciembre, 2019 de <http://rosa-menkman.blogspot.com/2010/04/glitches-vs-glitch-art.html>
3. Menkman, R. (2011). *The Glitch Momentum*. Recuperado el 30 diciembre, 2019. Disponible en https://networkcultures.org/_uploads/NN%234_RosaMenkman.pdf
4. Cloninger, C (2010) *GlitchLinguistx: The Machine in the Ghost / Static Trapped in Mouths*. Recuperado el 23 enero, 2020. Disponible en <http://lab404.com/glitch/>

5. Cascone, Kim. (2011). *Errormancy: glitch as divination*. Recuperado el 23 enero, 2020. Disponible en <https://sonicfield.org/2014/03/errormancy-glitch-as-divination/>
6. Temkin, D. (2014, 15 enero). *Glitch & Human/Computer Interaction*. NOOART. Recuperado el 24 diciembre, 2019 de <https://nooart.org/post/73353953758/temkin-glitchhumancomputerinteraction#ftnt4>
7. Assassin's Creed. Saga de Videojuegos. 2007-2018. Ubisoft Montreal, Gameloft y Griptonite Games. Ubisoft. Acción-aventura, ciencia ficción y ficción histórica. PlayStation 3, Xbox 360, Windows, Nintendo DS, Android, iOS, PlayStation Portable, PlayStation Vita, Wii U, PlayStation 4, Xbox One y PC.
8. Pac-Man 256. 2016. Hipster Whale, 3 Sprockets Pty Ltd y Bandai Namco Studios. Bandai Namco Entertainment y Bandai Namco Entertainment Europe SAS. Laberinto. Android, iOS, Microsoft Windows, PlayStation 4, Xbox One, macOS y Linux.
9. Pac-Man. 1980. Namco. Namco y Midway. Laberinto. Arcade, Android, Atari 2600, Atari 5200, Familia Atari de 8 bits, ColecoVision, Commodore 64, Commodore VIC-20, Consola Virtual, Famicom Disk System, Game Boy, Game Boy Advance, Intellivision, iPod, iOS, Microsoft Windows, MSX, NEC PC-6001, NEC PC-8001, NEC PC-8801, NEC PC-9801, Neo Geo Pocket Color, Nintendo Entertainment System, SAM Coupé, Sega Game Gear, Sharp MZ, Sharp X1, Sharp X68000, Teléfonos móviles, Windows Phone, Xbox Live Arcade y ZX Spectrum.
10. Pony Island. 2016. Daniel Mullins Games. Daniel Mullins Games. Acción y aventura. Microsoft Windows, OS X y Linux.
11. Doki Doki Literature Club! 2017. Team Salvato. Team Salvato. Novela visual y terror. Microsoft Windows, macOS y Linux.
12. Memory of a Broken Dimension. Proximamente. Ezra Hanson-White (XRA). Aventura. Microsoft Windows, macOS y Linux.
13. Brackeys (2019) FIRST PERSON MOVEMENT in Unity - FPS Controller. [Video] Recuperado el 2 de marzo, 2020, de https://www.youtube.com/watch?v=_QajrabyTJc
14. Widegarn, S. (2018) *Glitch* post-processing shader for Unity3d. [Repositorio] Recuperado el 11 de febrero, 2020, de <https://github.com/staffantan/unityglitch>
15. Takahashi, K. (2017) *KinoGlitch*. [Repositorio] Recuperado el 5 de febrero, 2020, de <https://github.com/keijiro/KinoGlitch>
16. Gamedev.stackexchange (2013) Can I use an animated gif as a texture? [Repositorio] Recuperado el 15 de febrero, 2020, de <https://gamedev.stackexchange.com/questions/35030/can-i-use-an-animated-gif-as-a-texture>
17. El Hajj, Wissam. (2019) Mesh and Object Deformers for Unity 3D. [Asset]. Recuperado el 6 de mayo, 2020, de <https://assetstore.unity.com/packages/tools/modeling/mesh-and-object-deformers-for-unity-3d-81427>
18. En estas cuatro paredes (in these four walls), https://ptrcia.github.io/tfm_unity/, último acceso julio 2020.