

---

**Generación de herramientas para desarrollar  
juegos con geometrías no euclidianas**  
**Creation of tools to develop games with  
non-Euclidean geometries**

---



**Trabajo de Fin de Grado**  
**Curso 2023–2024**

**Autor**  
**Noelia Barranco Godoy**

**Director**  
**Ismael Sagredo Olivenza**

**Doble Grado en Ingeniería Informática y Matemáticas**  
**Facultad de Informática**  
**Universidad Complutense de Madrid**



Generación de herramientas para  
desarrollar juegos con geometrías no  
euclidianas

Creation of tools to develop games with  
non-Euclidean geometries

**Trabajo de Fin de Grado en Ingeniería Informática**

**Autor**  
**Noelia Barranco Godoy**

**Director**  
**Ismael Sagredo Olivenza**

**Convocatoria:** *Junio 2024*

**Doble Grado en Ingeniería Inforomática y Matemáticas**  
**Facultad de Informática**  
**Universidad Complutense de Madrid**

**16 de febrero de 2024**



# Dedicatoria

{**TODO TODO TODO:** Hacer dedicatoria}



# Agradecimientos

{**TODO TODO TODO:** Hacer agradecimientos}



# Resumen

## **Generación de herramientas para desarrollar juegos con geometrías no euclidianas**

He creado una serie de herramientas para Unity, con la finalidad de facilitar a los desarrolladores de videojuegos la creación de juegos que introduzcan algunos elementos con geometrías no euclidianas.

Además, todas estas herramientas estarán desarrolladas en código abierto, con el objetivo de que cualquier usuario pueda ampliarlas con sus necesidades y compartir las con la comunidad de Unity.

## **Palabras clave**

{**TODO TODO TODO:** Máximo 10 palabras clave separadas por comas}



# Abstract

## **Creation of tools to develop games with non-Euclidean geometries**

{**TODO TODO TODO:** An abstract in English, half a page long, including the title in English. Below, a list with no more than 10 keywords.}

## **Keywords**

{**TODO TODO TODO:** 10 keywords max., separated by commas.}



# Capítulo 1

## Introducción

**RESUMEN:** En este capítulo se hará una pequeña introducción a los juegos que implementan físicas imposibles y se explorará la razón por la que pueden generar interés en la industria del videojuego.

### 1.1. Motivación

Dentro del sector de los videojuegos, uno de los géneros más antiguos es el género de puzzles. Entre la basta cantidad de variaciones que tiene este género, existe un subgénero concreto, los juegos de puzzles basados en físicas, habiendo algunos muy famosos como por (2011) o por (2017).

Además, una de las tendencias de la industria estos últimos años, inspirada por zel (2017) y continuada en la secuela, zel (2023), ha sido la de introducir en juegos triple A de mundo abierto mecánicas de puzzles y acertijos relacionados puramente con las físicas del mundo.

La longevidad de este género, sumado a la tendencia de la industria, ha hecho que cada vez sea más difícil innovar en este ámbito, lo que ha llevado a algunos desarrolladores a crear juegos de este subgénero en los que introducen objetos (o la totalidad de los mismos) que se comportan con físicas imposibles.

No obstante, dado que estos comportamientos pueden llegar a ser muy anti-intuitivos, se considera interesante la creación de una herramienta que ayude a los desarrolladores en esta tarea.

### 1.2. Aproximaciones al problema

Primero de todo, hemos de comprender que hay dos aproximaciones diferentes a la hora de implementar físicas imposibles en un videojuego: hacer un juego sobre

un motor regido por geometrías no euclidianas o utilizar un motor común y emplear técnicas para generar la ilusión de físicas imposibles.

### 1.2.1. Geometrías euclidianas y no euclidianas

Euclides (1996) hizo cinco postulados que definían lo que se él, intuitivamente, entendía que era la geometría. Más adelante, Hilbert y Townsed (1921) formalizó estos postulados en 21 axiomas (aunque posteriormente se demostró que uno de ellos era redundante).

**COMENTARIO:** Tengo la sensación de que no sé como utilizar las citas de manera correcta, espero que podamos hablarlo en algún momento y aclararlo

La geometría eucliana es aquella que sigue los cinco postulados de Euclides (o más formalmente, los axiomas de Hilbert). Por ende, una geometría no eucliana es cualquiera que no sigue uno de esos axiomas. No obstante, cuando hablamos de geometrías no euclidianas, usualmente nos referimos a hiperbólicas o elípticas, que únicamente cambian el quinto postulado, que es el siguiente:

*Si una recta secante corta a dos rectas formando a un lado ángulos interiores, la suma de los cuales sea menor que dos ángulos rectos; las dos rectas, suficientemente alargadas se cortarán en el mismo lado.*

O, equivalentemente:

*Por un punto exterior a una recta, se puede trazar una única paralela.*

Las geometrías hiperbólicas y elípticas cambian el quinto postulado afirmando que por un punto exterior a una recta se pueden trazar respectivamente infinitas paralelas o ninguna (Coxeter, 1998).

### 1.2.2. Juegos implementados sobre físicas no euclidianas

La aproximación clásica de la física indica que el espacio sigue las reglas de la geometría eucliana (aunque en realidad el espacio-tiempo se comporta según las reglas de una geometría no eucliana, en escalas humanas esto no es apreciable, Einstein (2000)). Esto significa que los motores físicos en el campo de los videojuegos imitan este comportamiento eucliano. No obstante, si pensamos en el espacio como algo curvo, con una geometría no eucliana, dado que percibimos las cosas mediante líneas rectas (los rayos de luz), todo sería muy diferente.

Todo esto indica que una aproximación válida para esto, aunque con un resultado muy lejano de lo intuitivo, es modificar un motor físico para que los objetos se perciban cómo si existiéramos en un espacio no eucliano. Un juego que muestra esto es hyp (2022) (**COMENTARIO: NO TENGO MUY CLARO COMO CITAR BIBLIOGRÁFICAMENTE A UN VIDEOJUEGO NI A UN VIDEO DE YOUTUBE**). El proceso de creación de este juego podemos encontrarlo documentado en el canal de *Youtube* del desarrollador, CodeParade (2020).

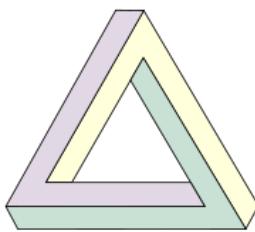


Figura 1.1: Triángulo de Penrose

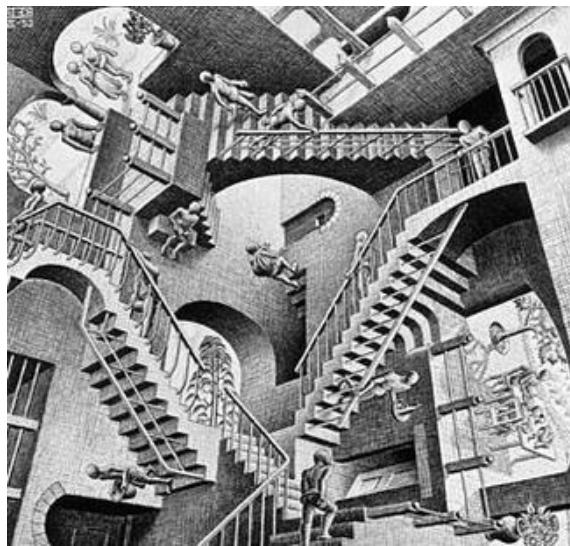


Figura 1.2: Escher, M. (1953). Relatividad

### 1.2.3. Juegos con elementos imposibles

La segunda aproximación, más común y en la que se centrará este trabajo, es la de introducir elementos que no siguen las reglas de la física clásica en un juego, sin hacer que todas las físicas del juego trabajen bajo reglas de geometrías no euclidianas. Estos elementos pueden ser simplemente físicas imposibles (ya porque desafían el concepto de gravedad o por ser espacios imposibles, como podemos ver en las figuras 1.1 y 1.2) o elementos pertenecientes a físicas no euclidianas (como una habitación que es más grande por dentro que por fuera). No obstante, esta diferencia no tiene importancia ni en implementación ni en resultado (ya que al usuario no va a importarle si el efecto proviene de un objeto imposible o de una interpretación de la geometría no euclidiana).

## 1.3. Objetivos

Como se comentó en la sección anterior, este trabajo se centrará en desarrollar herramientas para facilitar la implementación de objetos que no sigan las leyes de la

física habituales (la segunda aproximación de la sección anterior). En el capítulo 2 se estudiarán las mecánicas más comunes en juegos de este subgénero, y se concretará qué herramientas vamos a implementar.

## 1.4. Plan de trabajo

En esta sección describiremos las distintas fases que se seguirán para realizar este trabajo.

{TODO TODO TODO: Refinar}

### 1.4.1. Documentación y preparación

Esta fase se realizará antes de empezar el trabajo como tal y abarca los capítulos primero y segundo de este documento. Se recopila toda la información posible sobre físicas imposibles en juegos, se estudia qué tipos suelen ser usados y se deciden qué herramientas se van a desarrollar en base a la información obtenida.

También se estudia qué recursos hay ya disponibles para crear este tipo de juegos (que corresponde al segundo capítulo del documento).

### 1.4.2. Implementación de las herramientas

Para cada herramienta descrita en la sección 2.2, se crearán las herramientas necesarias en Unity.

### 1.4.3. Documentar las herramientas

Para cada una de las herramientas implementadas, se hará una sección del documento en la que se expondrán las complicaciones y los detalles del desarrollo de las mismas.

### 1.4.4. Creación de la librería

Todas las herramientas se juntarán en una librería que estará disponible bajo una licencia MIT en este repositorio de github.

### 1.4.5. Demo

Una vez tengamos todas las herramientas creadas, se creará una pequeña demo técnica que utilice todas las mecánicas implementadas.

#### **1.4.6. Conclusiones**

Por último, se hará un sumario de todo lo que se ha realizado en este trabajo, que quedará reflejado en el capítulo 4.



# Capítulo 2

## Estado de la Cuestión

**RESUMEN:** Para estudiar las mecánicas, así como las posibles implementaciones, lo más inmediato es obtenerlas de juegos que las utilicen.

### 2.1. Mecánicas

Antes de estudiar las mecánicas de los juegos, tenemos que hacer una distinción. Tanto a nivel de *gameplay* como a nivel de diseño, hay una gran diferencia entre juegos con una cámara móvil (ya sean en primera o tercera persona) y con cámara fija (por ejemplo juegos 2D o en perspectiva isométrica). Algunos ejemplos de juegos con cámara fija que podrían entrar en el tema de este trabajo son fez (2013) o mon (2022).

En este trabajo no vamos a tratar este tipo de juegos por dos motivos. Por un lado, debido a las diferencias en el funcionamiento, las herramientas para implementar una misma mecánica en los dos tipos de juegos son completamente distintas. Además, debido a la naturaleza de estos juegos y de sus ilusiones, la mayoría de estas mecánicas se implementan utilizando trucos como duplicar objetos o eliminar texturas, que no son susceptibles de ser generalizados fuera de ese uso en concreto (Kwan, 2017).

**COMENTARIO:** Esta es mi idea de qué voy a hacer y qué no, pero si lo ves conveniente, puede cambiarse. Es solo que me parece más interesante hacer una librería de herramientas para juegos en 1/3 persona antes que algo para cámaras fijas, que como ya vimos va mucho más sobre ruedas

- Portal (y su secuela)
- Antichamber
- Superliminal
- Manifold Garden
- Psychonauts 2
- Viewfinder

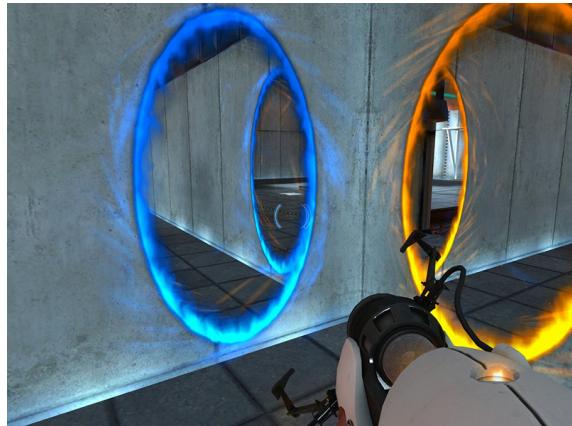


Figura 2.1: Captura *in-game* de Portal

De esta lista de videojuegos, podemos extraer una serie de mecánicas comunes, que estudiaremos individualmente.

### 2.1.1. Portales

Esta mecánica es la más simple e intuitiva de todas. Como ocurre en Portal (Figura 2.1) o Psychonauts 2, en ocasiones hay portales, que a nivel de implementación simplemente son parejas de objetos planos que, al entrar en contacto con un objeto, lo teletransportan a su pareja. Es importante para mantener la ilusión que la implementación mantenga propiedades como la velocidad (o para ser exactos, el módulo de la velocidad y el ángulo respecto del vector normal al portal). Dentro de las variaciones, los portales pueden dejar ver lo que hay al otro lado o no y pueden modificar propiedades como la gravedad o el tamaño.

Respecto al trabajo existente en este campo, existe una implementación bajo licencia MIT en este repositorio que mejoraremos y añadiremos a nuestra librería (**COMENTARIO: EN REALIDAD ESTO ESTÁ POR VER, PERO NO TIENE MUCHO SENTIDO HACER ALGO QUE YA ESTÁ HECHO, MENOS AÚN SI TIENE LICENCIA LIBRE. LA OTRA OPCIÓN ES HACER UNA IMPLEMENTACIÓN PROPIA** ).

**{TODO TODO TODO}**: Si al final uso este video para algo, mencionarlo y añadirlo a la bibliografía}

### Habitaciones en espacios imposibles

Una mecánica que podemos ver en juegos como antichamber (Figura 2.2) es hacer que en una habitación, dependiendo por la zona por la que entremos (o por la que miremos), haya diferentes objetos dentro.

Aunque no lo aparente, esto es solo un caso de uso más de los portales, pues se crea esa ilusión haciendo que cada entrada de la habitación sea en realidad un portal que lleve a otro sitio. Además con esta técnica también podemos crear la ilusión de

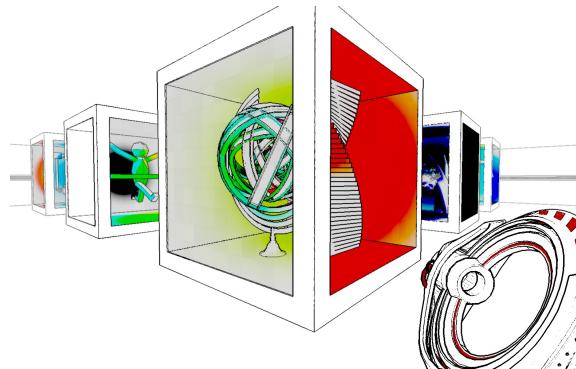


Figura 2.2: Captura *in-game* de Antichamber

que una habitación por dentro es más grande que por fuera.

### 2.1.2. Perspectiva forzada

La perspectiva forzada es una de las ilusiones ópticas más habituales, en la que se aprovecha el hecho de que en una imagen plana no hay referencias para medir las distancias para generar la ilusión de que un objeto tiene un tamaño diferente al que realmente tiene. Podemos encontrar uno de los ejemplos más habituales de esto en la figura 2.3.

Esto como tal no es una mecánica imposible, es solo un efecto óptico, no obstante juegos como Superliminal modifican realmente el tamaño de un objeto que tenemos agarrado de forma que lo que en principio es una ilusión afecta realmente al tamaño de los objetos, como podemos observar en 2.4.

Respecto a la implementación de esta mecánica, sabemos gracias a Francis (2020) que es tan simple como intuitiva. Cuando llevamos un objeto en la mano, siempre se va a proyectar todo lo lejos que el escenario permita. La forma en la que se determina esta distancia es mediante el uso de una densa capa de *raycast* para estimar la distancia a la que el objeto debe de estar. Luego simplemente se redimensiona el elemento, teniendo en cuenta la distancia inicial y la actual de este a la cámara.

### 2.1.3. Decal

Un *Decal* o calcomanía es un efecto óptico simple que está implementado en Unity (solo si utilizamos HDRP) y que también aparece en varias librerías (como esta). Es el resultado de proyectar una imagen plana sobre un objeto tridimensional, haciendo la ilusión de que la imagen está en tres dimensiones también, como podemos observar en la figura 2.5.

COMENTARIO: Esto que voy a poner ahora no sé si entraría dentro de la sección de Decal o, como me imagino que se implementará de una forma diferente, ponerlo en su propia sección

Esto puede hacerse de formas más complejas para crear, más que una ilusión



Figura 2.3: Imagen de una turista en la torre de Pisa

óptica, una mecánica jugable. Al hacer el decal, podemos querer que en lugar de aparecer sobre una sola superficie, se reparta entre varias distintas (ver figura 2.6), y permitir al jugador coger ese objeto (que inicialmente es solo una textura) solo si está en la posición en la que se ve correctamente.

#### 2.1.4. Capturas de objetos

Como podemos encontrar en Viewfinder, otra mecánica muy interesante es la de tomar una captura de lo que tiene en pantalla el jugador en un momento dado, para posteriormente crear un duplicado de esos elementos desde la perspectiva del jugador, como podemos ver en 2.7.

### 2.2. Objetivos

Habiendo visto las mecánicas que ofrecen los juegos que hemos estudiado, podemos finalmente hacer una lista con los objetivos que vamos a tener para este trabajo.

#### Portales

Se partirá del repositorio mencionado en 2.1.1, se mejorará y se integrará en la librería.



Figura 2.4: Pieza de ajedrez de gran tamaño debido a la perspectiva.



Figura 2.5: Calcomanía de un peón en Superliminal.

### Perspectiva forzada

Se implementarán un tipo de objetos que, mediante el mecanismo descrito en el artículo, permitan alterar su tamaño usando la perspectiva.

### Decal

Se hará, modificando una librería ya existente, un *decal* que permita proyectar sobre varios objetos. Además, utilizando esto, se creará una herramienta que permita al desarrollador crear objetos que aparezcan al estar en el punto adecuado introduciendo solamente el objeto a aparecer, el punto desde donde debe de verse y la dirección de la cámara sobre la que se proyectará este.

### Capturas

Se implementará una herramienta que permita capturar el entorno actual para su posterior proyección en otro lugar.

COMENTARIO: No sé si es demasiado ambicioso, pero me gustaría implementar una mecánica original además de todo esto, pero para eso primero se me tiene que ocurrir una mecánica...



Figura 2.6: Calcomanía de un cubo sobre varias superficies.



Figura 2.7: Proyección sobre la realidad de una fotografía tomada anteriormente.

Capítulo **3**

## Descripción del Trabajo

Aquí comienza la descripción del trabajo realizado. Se deben incluir tantos capítulos como sea necesario para describir de la manera más completa posible el trabajo que se ha llevado a cabo. Como muestra la figura 3.1, está todo por hacer.



Figura 3.1: Ejemplo de imagen

Si te sirve de utilidad, puedes incluir tablas para mostrar resultados, tal como se ve en la tabla 3.1.

Col 1	Col 2	Col 3
3	3.01	3.50
6	2.12	4.40
1	3.79	5.00
2	4.88	5.30
4	3.50	2.90
5	7.40	4.70

Tabla 3.1: Tabla de ejemplo



Capítulo **4**

## Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de grado, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF.



# Introduction

Introduction to the subject area. This chapter contains the translation of Chapter 1.



# Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter 4.



# Bibliografía

*Y así, del mucho leer y del poco dormir, se  
le secó el cerebro de manera que vino a  
perder el juicio.  
(modificar en Cascaras\bibliografia.tex)*

Miguel de Cervantes Saavedra

Portal. [Edición Digital], 2011.

Fez. [Edición Digital], 2013.

The legend of zelda: Breath of the wild. [Nintendo Switch Gamecard], 2017.

Portal 2. [Edición Digital], 2017.

Hyperbolica. [Edición Digital], 2022.

Monument valley. [Panoramic Edition], 2022.

The legend of zelda: Tears of the kingdom. [Nintendo Switch Gamecard], 2023.

CODEPARADE. Hyperbolica devlog. 2020.

COXETER, H. *Non-Euclidean Geometry*. American Mathematical Society, 1998.

EINSTEIN, A. *Meaning of Relativity*. ElecBook, 2000.

EUCLIDES. *Elementos*. Editorial Gredos, 1996.

FRANCIS, B. Designing the mind-bending perspective puzzles of superliminal. 2020.

HILBERT, D. y TOWNSEND, E. *The foundations of geometry*. Chicago, Open court Pub. CO., 1921.

KWAN, W. Creating the monument valley illusion in unity. 2017.



Apéndice

**A**

## Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.



# Apéndice **B**

## Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.



Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFGTeXiS.tex.

*-¿Qué te parece desto, Sancho? – Dijo Don Quijote –  
Bien podrán los encantadores quitarme la ventura,  
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero  
Don Quijote de la Mancha  
Miguel de Cervantes*

*-Buena está – dijo Sancho –; fírmela vuestra merced.  
–No es menester firmarla – dijo Don Quijote–,  
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero  
Don Quijote de la Mancha  
Miguel de Cervantes*

