
Propuesta de curso sobre arte técnico para la especialización de videojuegos de la carrera de Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Jorge Manuel De León Pinelo



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Propuesta de curso sobre arte técnico para la
especialización de videojuegos de la carrera de
Ingeniería en Ciencias de la Computación y
Tecnologías de la Información**

Trabajo de graduación en modalidad de tesis presentado por
Jorge Manuel De León Pinelo
para optar al grado académico de Licenciado en Ingeniería en Ciencias
de la Computación y Tecnologías de la Información

Guatemala
2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

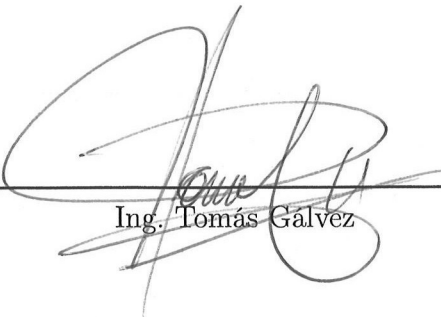


**Propuesta de curso sobre arte técnico para la
especialización de videojuegos de la carrera de
Ingeniería en Ciencias de la Computación y
Tecnologías de la Información**

Trabajo de graduación en modalidad de tesis presentado por
Jorge Manuel De León Pinelo
para optar al grado académico de Licenciado en Ingeniería en Ciencias
de la Computación y Tecnologías de la Información

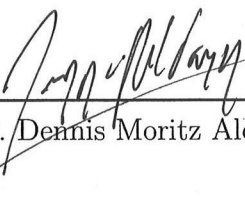
Guatemala
2023

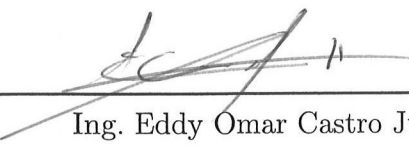
Vo.Bo.:

(f) 
Ing. Tomás Galvez

Tribunal Examinador:

(f) 
Ing. Tomás Galvez

(f) 
Ing. Dennis Moritz Aldana Moscoso

(f) 
Ing. Eddy Omar Castro Juaregui

Fecha de aprobación: Guatemala, 4 de diciembre de 2023.

Agradecimientos

Agradezco a los estudiantes de la promoción 2023 de la especialización de videojuegos y a los catedráticos Msc. Dennis Aldana, Ing. Alhvi Balcarcel, Ing. Tomás Gálvez, Ing. Eddy Castro y Msc. Douglas Barrios por su guía, comentarios, retroalimentación y apoyo durante el desarrollo de este proyecto.

Finalmente, se agradece a la Facultad de Ingeniería de la Universidad del Valle de Guatemala por aprobar la producción del proyecto.

Agradecimientos	I
Lista de figuras	XIII
Resumen	XIV
1. Introducción	1
2. Objetivos	2
2.1. Objetivo general	2
2.2. Objetivos específicos	2
3. Justificación	3
4. Marco teórico	5
5. Marco metodológico	10
6. Resultados	12
6.1. Competencias del curso	12
6.2. Material didáctico producido	13
6.2.1. Presentación sobre la creación de <i>Shaders</i>	13
6.2.2. Presentación sobre la creación de efectos de <i>post-processing</i>	16
6.3. Diseño preliminar del curso	18
6.4. Retroalimentación obtenida	20

6.4.1. Creación de <i>shaders</i>	21
6.10.1. Efectos de <i>post-processing</i>	26
6.16.1. Efectos visuales avanzados haciendo uso de múltiples herramientas	31
6.18. Retroalimentación a cursos actuales de la especialización de videojuegos	34
6.18.1. Retroalimentación para el curso de diseño de videojuegos	34
6.21.1. Retroalimentación para el curso de programación de videojuegos	36
6.24.1. Retroalimentación para el curso de arte y animación digital	39
6.27.1. Retroalimentación para el curso de escritura creativa	41
7. Discusión	45
7.1. Creación de <i>shaders</i>	45
7.2. Efectos de <i>post-processing</i>	47
7.3. Efectos visuales avanzados haciendo uso de múltiples herramientas	50
7.4. Retroalimentación a cursos actuales de la especialización de videojuegos	51
8. Conclusiones	54
9. Recomendaciones	55
10. Bibliografía	57
11. Anexos	59
A. Material didáctico producido	60
A.1. Presentación de introducción al curso	60
A.2. Presentación de efectos de <i>post-processing</i>	64
A.2.1. Enlace a repositorio de demostración usado para impartir el tema de creación de efectos de <i>post-processing</i>	64
A.3. Presentación de creación de <i>shaders</i>	71

A.3.1. Enlace a repositorio de demostración usado para impartir el tema de creación de <i>shaders</i>	71
A.3.2. Listado de funciones de HLSL cubiertas en la presentación para el tema de creación de <i>shaders</i> . . .	94
A.3.3. Listado de técnicas cubiertas en la presentación para el tema de creación de <i>shaders</i>	95
A.3.4. Listado de configuraciones cubiertas en la presentación para el tema de creación de <i>shaders</i>	95
B. Bitácora de ejecución de metodología	96

4.1.	Capturas de algunos pasos que toma la <i>rendering pipeline</i> de Unity para mostrar en pantalla una escena que contiene una pared, una caja, tres esferas flotantes, dos fuentes de luz cercanas, una azul y una naranja, y una fuente de luz lejana. Los pasos mostrados son los que muestra la herramienta de <i>frame debugging</i> de Unity [36], por lo que no se trata de una lista ilustrativa y no exhaustiva de los pasos.	6
4.2.	Monos de Blender demostrando el fenómeno conocido como <i>clipping</i> (discutido en el párrafo anterior). Se muestra a un mono sin el problema (izquierda) y a dos monos superpuestos demostrando el problema (derecha).	7
4.3.	Capturas de pantalla del juego «God of War: Ragnarok» [23] que muestran una escena en la que se aplica un efecto de <i>post-processing</i> que superpone una tormenta de arena sobre del juego (arriba) y una escena en la misma locación (Alfheim) en la que no está presente la tormenta, por lo que no hace uso del efecto (abajo).	9
6.1.	Novena diapositiva de la presentación creada para impartir el tema de Creación de <i>shaders</i> que ilustra la diferencia entre usar un <i>shader</i> de vértices y un <i>shader</i> de fragmento para computar un efecto de iluminación.	14

6.2.	Dispositiva 22 de la presentación creada para el tema de creación de <i>shaders</i> que ilustra rápidamente las ventajas (Paralelismo y Hardware especializado) y desventajas (Crea un cuello de botella y limita el uso de bifurcaciones en el flujo de ejecución de un algoritmo) que se tienen al computar operaciones de álgebra lineal en una GPU. . . .	15
6.3.	Demostración creada para impartir el tema de creación de <i>shaders</i> , muestra un efecto de simulación de olas en una isla, animado en base a senos, cosenos y parametrizado en base a texturas que indican cómo y qué se debe deformar durante la animación.	16
6.4.	Dispositiva 5 de la presentación creada para el tema de creación de efectos de <i>post-processing</i> que muestra el propósito de un componente de volumen, proveer una manera de configurar un efecto de <i>post-processing</i> por medio de controles amigables a usuarios no técnicos.	17
6.5.	Demostración creada para impartir el tema de creación de <i>shaders</i> , muestra una comparación de la escena sin los efectos de aplicados (izquierda) y con los efectos aplicados (derecha).	18
6.6.	Tabla de cronograma preliminar para el curso.	19
6.7.	Tabla de ponderación preliminar para el curso.	20
6.8.	Gráfico de la pregunta «¿Cómo calificaría la calidad de la clase?» que fue hecha en forma de escala Likert de 1 (Mala) a 10 (Buena). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	21
6.9.	Gráfico de la pregunta «¿Entendió los temas presentados?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	21

-
- 6.10. Gráfico de la pregunta «Se presentaron varios aspectos técnicos sobre cómo se puede configurar herramientas como Unity para modificar el comportamiento de un *Shader* (Transparencia, *Blending*, *Face Culling*, *Depth Testing*, Máscaras de color), pero muy pocos ejemplos de cómo esto sería aplicable. ¿Qué tan importante considera la inclusión de estos ejemplos para la retención de los temas?» que fue hecha en forma de escala Likert de 1 (De poca importancia) a 10 (De suma importancia). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. 22
- 6.11. Gráfico de la pregunta «¿Cree que ese conocimiento [cómo crear *shaders*] le sería útil como estudiante de la especialización de videojuegos?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, sumamente útil). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. . . . 23
- 6.12. Gráfico de la pregunta «¿Cree que los conocimientos impartidos en los cursos de la especialización de videojuegos y en la carrera en general le dan una buena base para aprender sobre la creación y configuración de *shaders*?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. 23
- 6.13. Gráfico de la pregunta «¿Cómo califica la calidad de la clase?» que fue hecha en forma de escala Likert de 1 (Mala) a 10 (Excelente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. 26

6.14. Gráfico de la pregunta «¿Qué tanto entendió el tema de creación de efectos de <i>post-processing</i> ?» que fue hecha en forma de escala Likert de 1 (Para nada) a 10 (Completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	26
6.15. Gráfico de la pregunta «¿Se considera capaz de realizar el ejercicio presentado al final de la clase?» que fue hecha en forma de escala Likert de 1 (Para nada) a 10 (Completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	27
6.16. Gráfico de la pregunta «¿Considera que los temas presentados le serían útiles a los futuros estudiantes de la especialización?» que fue hecha en forma de escala Likert de 1 (Son irrelevantes) a 10 (Su inclusión en el programa sería muy buena). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	28
6.17. Gráfico de la pregunta «¿El video explica bien los temas y su valor?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, es muy claro). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos)	31
6.18. Gráfico de la pregunta «¿Tiene sentido incluir este tema en un curso sobre el Arte Técnico?» que fue hecha en forma de escala Likert de 1 (No, no tiene sentido) a 10 (Sí tiene sentido). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos) .	31

-
- 6.19. Gráfico de la pregunta «¿Aprender sobre estas técnicas aporta valor a un estudiante de la especialización de videojuegos?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, es sumamente útil). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos) 32
- 6.20. Gráfico de la pregunta «¿Considera necesario impartir estos temas en relación a herramientas específicas (Blender, Maya, Unity, Godot, Unreal...)?» que fue hecha en forma de escala Likert de 1 (No, los fundamentos son agnósticos a las herramientas) a 10 (Sí, se requiere un ejemplo práctico en una herramienta para comprender los temas). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos) 33
- 6.21. Gráfico de la pregunta «¿Considera que los estudiantes de la especialización de videojuegos tienen los conocimientos previos necesarios para aplicar los temas?» que fue hecha en forma de escala Likert de 1 (No, requieren muchos temas base que no están incluidos en los cursos del programa) a 10 (Sí, son capaces de aplicarlo solo con lo que aprendieron en la especialización y la carrera en general). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos) 33

6.22.	Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Diseño de videojuegos) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. . . .	34
6.23.	Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Diseño de videojuegos) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	34
6.24.	Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Programación de videojuegos) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. . . .	36
6.25.	Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Programación de videojuegos) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	37

6.26. Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Arte y animación digital) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. . . .	39
6.27. Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Arte y animación digital) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	39
6.28. Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Escritura creativa) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	41
6.29. Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Escritura creativa) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.	42
A.1. Diapositiva 1 de la presentación de introducción al curso .	60
A.2. Diapositiva 2 de la presentación de introducción al curso .	61
A.3. Diapositiva 3 de la presentación de introducción al curso .	61
A.4. Diapositiva 4 de la presentación de introducción al curso .	62
A.5. Diapositiva 5 de la presentación de introducción al curso .	62
A.6. Diapositiva 6 de la presentación de introducción al curso .	63

A.7. Diapositiva 7 de la presentación de introducción al curso .	63
A.8. Diapositiva 8 de la presentación de introducción al curso .	64
A.9. Diapositiva 1 de la presentación de introducción al curso .	64
A.10. Diapositiva 2 de la presentación de introducción al curso .	65
A.11. Diapositiva 3 de la presentación de introducción al curso .	65
A.12. Diapositiva 4 de la presentación de introducción al curso .	66
A.13. Diapositiva 5 de la presentación de introducción al curso .	66
A.14. Diapositiva 6 de la presentación de introducción al curso .	67
A.15. Diapositiva 7 de la presentación de introducción al curso .	67
A.16. Diapositiva 8 de la presentación de introducción al curso .	68
A.17. Diapositiva 9 de la presentación de introducción al curso .	68
A.18. Diapositiva 10 de la presentación de introducción al curso .	69
A.19. Diapositiva 11 de la presentación de introducción al curso .	69
A.20. Diapositiva 12 de la presentación de introducción al curso .	70
A.21. Diapositiva 13 de la presentación de introducción al curso .	70
A.22. Diapositiva 14 de la presentación de introducción al curso .	71
A.23. Diapositiva 15 de la presentación de introducción al curso .	71
A.24. Diapositiva 1 de la presentación de introducción al curso .	72
A.25. Diapositiva 2 de la presentación de introducción al curso .	72
A.26. Diapositiva 3 de la presentación de introducción al curso .	73
A.27. Diapositiva 4 de la presentación de introducción al curso .	73
A.28. Diapositiva 5 de la presentación de introducción al curso .	74
A.29. Diapositiva 6 de la presentación de introducción al curso .	74
A.30. Diapositiva 7 de la presentación de introducción al curso .	75
A.31. Diapositiva 8 de la presentación de introducción al curso .	75
A.32. Diapositiva 9 de la presentación de introducción al curso .	76
A.33. Diapositiva 10 de la presentación de introducción al curso .	76
A.34. Diapositiva 11 de la presentación de introducción al curso .	77
A.35. Diapositiva 12 de la presentación de introducción al curso .	77
A.36. Diapositiva 13 de la presentación de introducción al curso .	78
A.37. Diapositiva 14 de la presentación de introducción al curso .	78
A.38. Diapositiva 15 de la presentación de introducción al curso .	79
A.39. Diapositiva 16 de la presentación de introducción al curso .	79
A.40. Diapositiva 17 de la presentación de introducción al curso .	80
A.41. Diapositiva 18 de la presentación de introducción al curso .	80
A.42. Diapositiva 19 de la presentación de introducción al curso .	81

A.43.	Diapositiva 20 de la presentación de introducción al curso	81
A.44.	Diapositiva 21 de la presentación de introducción al curso	82
A.45.	Diapositiva 22 de la presentación de introducción al curso	82
A.46.	Diapositiva 23 de la presentación de introducción al curso	83
A.47.	Diapositiva 24 de la presentación de introducción al curso	83
A.48.	Diapositiva 25 de la presentación de introducción al curso	84
A.49.	Diapositiva 26 de la presentación de introducción al curso	84
A.50.	Diapositiva 27 de la presentación de introducción al curso	85
A.51.	Diapositiva 28 de la presentación de introducción al curso	85
A.52.	Diapositiva 29 de la presentación de introducción al curso	86
A.53.	Diapositiva 30 de la presentación de introducción al curso	86
A.54.	Diapositiva 31 de la presentación de introducción al curso	87
A.55.	Diapositiva 32 de la presentación de introducción al curso	87
A.56.	Diapositiva 33 de la presentación de introducción al curso	88
A.57.	Diapositiva 34 de la presentación de introducción al curso	88
A.58.	Diapositiva 35 de la presentación de introducción al curso	89
A.59.	Diapositiva 36 de la presentación de introducción al curso	89
A.60.	Diapositiva 37 de la presentación de introducción al curso	90
A.61.	Diapositiva 38 de la presentación de introducción al curso	90
A.62.	Diapositiva 39 de la presentación de introducción al curso	91
A.63.	Diapositiva 40 de la presentación de introducción al curso	91
A.64.	Diapositiva 41 de la presentación de introducción al curso	92
A.65.	Diapositiva 42 de la presentación de introducción al curso	92
A.66.	Diapositiva 43 de la presentación de introducción al curso	93
A.67.	Diapositiva 44 de la presentación de introducción al curso	93
A.68.	Diapositiva 45 de la presentación de introducción al curso	94
A.69.	Diapositiva 46 de la presentación de introducción al curso	94
B.1.	Bitácora de ejecución (continuado)	96
B.2.	Bitácora de ejecución (continuado)	97
B.3.	Bitácora de ejecución.	97

Se recopilaron opiniones de estudiantes y catedráticos para recomendar varios cambios a la especialización de videojuegos de la carrera de Ciencias de la Computación y Tecnologías de la Información de la Universidad del Valle de Guatemala. Estos se hicieron como cambios propuestos a los cursos provistos actualmente por la especialización y como la propuesta de introducir un curso nuevo que sirva para culminar la especialización al combinar conocimientos de producción de arte digital y programación de videojuegos; recontextualizando temas para que sean más valiosos para los estudiantes e introduciendo temas nuevos que no se incluyen en los cursos actuales, siendo estos temas la creación de *shaders*, efectos de *post-processing* y manejo de *assets*. El valor y viabilidad de la introducción de cada tema fue evaluado individualmente para dar un mejor contexto al departamento de computación en dado caso se desee solamente introducir uno de los temas a uno de los cursos actualmente provistos por el departamento.

Se encontró que la respuesta general a los cambios propuestos fue positiva, tanto por parte de estudiantes como de catedráticos. La retroalimentación obtenida, a pesar de no ser evidencia concluyente debido a que se trató de una muestra de solo seis estudiantes de promoción 2023 de la especialización de videojuegos, indica que el departamento de computación debe considerar aplicar los cambios propuestos en este trabajo al pènsum de la especialización de videojuegos de la carrera y al contenido de los cursos en sí.

En este trabajo se investigó sobre el estado actual de la especialización de videojuegos de la carrera de ciencias de la computación y tecnologías de la información de la Universidad del Valle de Guatemala con el objetivo de crear recomendaciones generales para mejorar la especialización, incluyendo la proposición de la inclusión de un nuevo curso orientado a culminar los conocimientos adquiridos por los estudiantes a lo largo de la especialización al combinar habilidades de las disciplinas de programación y arte digital para crear efectos visuales avanzados que otorgan un acabado visual único y más profesional a los videojuegos que los implementan.

Se tomaron en cuenta las opiniones de la mayoría de estudiantes de la promoción 2023 de la especialización y la retroalimentación de los catedráticos que han diseñado e impartido cursos específicos a la especialización para producir material didáctico que se puede usar para integrar los temas propuestos en los cursos actuales o introducir los temas propuestos como un añadido a los cursos que actualmente provee el departamento, en dado caso el departamento de computación determine que el valor proporcionado a los estudiantes por aprender el tema sea lo suficiente como para desear incluir los temas en los cursos.

2.1. Objetivo general

Proveer al departamento de computación de herramientas didácticas y bibliografía para mejorar la especialización de videojuegos de la carrera en la forma del prototipo de un curso que amplíe el contenido actualmente ofrecido por la especialización, apoyándose de la retroalimentación de estudiantes y catedráticos.

2.2. Objetivos específicos

- Definir los prerrequisitos del curso y las competencias que un estudiante debe desarrollar al tomar el curso.
- Redactar un temario completo, junto con la justificación para la inclusión del tema, tomando en cuenta el interés de los estudiantes y el valor provisto por aprender los temas.
- Crear un cronograma que priorice los temas según el tiempo necesario para impartirlos y el interés en los temas por parte de los estudiantes.
- Diseñar y probar material didáctico para el curso, con por lo menos un laboratorio por tema o proyectos que integren varios temas; con tal de que ningún tema quede sin ser aplicado prácticamente durante el curso.

La proposición de un nuevo curso para la especialización de videojuegos implica que esta tiene puntos que mejorar y el espacio para acomodar un curso nuevo. Actualmente esta cuenta con seis cursos y la mitad de estos, Arte y animación digital, Música computacional y Escritura creativa, buscan complementar el conocimiento sobre computación y desarrollo de software con el que cuentan los estudiantes para que tengan una base en disciplinas que no tienen que ver directamente con computación, pero que, considerando que se tratan de obras audiovisuales que presentan una narrativa, son necesarias para crear un videojuego.

Habiendo establecido lo anterior, se debe considerar que los estudiantes han expresado insatisfacción hacia algunos de los cursos de la especialización y consideran que algunos cursos, en especial Escritura creativa, no justifican su inclusión en el programa; mientras que expresan que cursos como Arte y animación digital necesitan de un curso complementario para poder abarcar los temas relevantes para la disciplina. (ver 6.18) Considerando que según las opiniones recopiladas en el mismo 6.18, existe un deseo por parte de los estudiantes de se hagan cambios a la especialización de videojuegos y que estos estén enfocados en que los estudiantes desarrollen un perfil más especializado, aunque este necesite complementarse con ayuda de terceros capaces en otras disciplinas necesarias para desarrollar un videojuego.

La inclusión de un curso sobre el Arte técnico sería un paso hacia reenfocar la especialización hacia cultivar las habilidades de programación

específicas al desarrollo de videojuegos de los estudiantes en vez de complementarlas con otras disciplinas. Cabe resaltar que el curso no buscará preparar a los estudiantes para ser artistas técnicos, ya que se trata de una disciplina desempeñada por gente que tiene amplios conocimientos sobre arte digital y programación de gráficas computacionales y requiere un conocimiento experto de alguna de las dos disciplinas, sino es que ambas. El objetivo es inspirarse en la disciplina del Arte técnico, aprovechando el conectivismo[26] para construir sobre los conocimientos adquiridos en los cursos de programación y arte digital al combinar las disciplinas. Con el objetivo final de que los estudiantes tengan mejores herramientas para darle un acabado visual profesional y único a sus juegos, así como poder crear herramientas de desarrollo para los miembros no técnicos de un equipo de desarrollo.

El aporte que daría un curso inspirado en el Arte técnico es un contexto capaz de enseñar técnicas populares en la industria[2] que no son incluidos en los cursos de la especialización de videojuegos por ser demasiado avanzados, requerir el conocimiento de temas cubiertos en otros cursos o simplemente por no estar alineados con ninguno de los cursos con los que la especialización cuenta actualmente. Estos temas son el desarrollo y configuración de *Shaders*, materiales y efectos visuales avanzados, la configuración de software de arte como, Blender y Maya, y *Game Engines* como, Unity o Godot, y el desarrollo de herramientas para el equipo de desarrollo. El valor de aprender sobre estos temas proveerá a los estudiantes de la especialización con capacidades distintivas que les permitirán crear juegos tomando decisiones creativas previamente impedidas por su discapacidad técnica; a la vez que eficientizan los procesos usados durante la creación de un videojuego.

El arte técnico es una disciplina desarrollada por profesionales que son expertos en las áreas de arte digital y programación de gráficas.[41] Utilizan sus conocimientos para la resolución de problemas cuya resolución requiere de conocimientos de ambas disciplinas e involucran la coordinación del equipo de programación con el equipo de arte.[40] La preparación necesaria para ejercer en este puesto supera con creces el alcance de un curso, siendo este objetivo más adecuado para una maestría. Inclusive algunas de las capacidades que debe tener un artista técnico, como el desarrollo de efectos visuales, el desarrollo de *shaders* o la creación de herramientas para el equipo de desarrollo, son temas lo suficientemente amplios como para poder hacer uno o varios cursos para abarcarlos.[5]

Las responsabilidades que tiene un artista técnico varía drásticamente a lo largo de la industria. Esto ocurre por las diferentes necesidades que puede tener un proyecto según su visión artística, qué tan grande y especializado sea el equipo. A esto se suma el que se trata de una disciplina reciente, que se define al combinar dos disciplinas, por lo que es de esperar que sea difícil de definir. Aún así, se entiende que un artista técnico tiene la responsabilidad de crear herramientas que apoyen al equipo de arte, ya sea automatizando trabajo o abstrayendo conceptos técnicos, crear efectos visuales y facilitar la comunicación entre los equipos de arte y programación. [41]

La teoría más importante sobre lo discutido en este trabajo es el proce-

so de cómo se determina qué se muestra en pantalla durante la ejecución de un videojuego. Se lo como «*rendering*» y se castellaniza informalmente como «*renderizado*» [1]. Para llevar a cabo el proceso, se hace uso de algo que se conoce como *renderin pipelines*, que parten el proceso en varios pasos que puede o no ser personalizables por el usuario de una GPU según el estándar establecido el grupo Khronos, entidad conformada por fabricantes de GPUs como Nvidia, Intel y AMD. [11] Una *rendering pipeline* común tiene dos pasos grandes, la definición de vértices y la definición de fragmentos, que respectivamente definen si un objeto aparecerá en y qué parte de la pantalla lo hará, y qué color se muestra en pantalla para cada objeto.

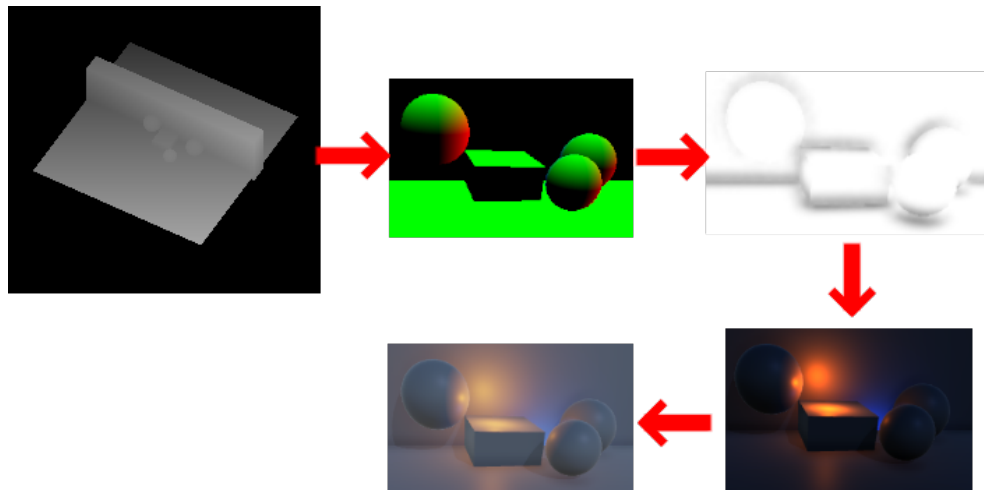


Figura 4.1: Capturas de algunos pasos que toma la *rendering pipeline* de Unity para mostrar en pantalla una escena que contiene una pared, una caja, tres esferas flotantes, dos fuentes de luz cercanas, una azul y una naranja, y una fuente de luz lejana. Los pasos mostrados son los que muestra la herramienta de *frame debugging* de Unity [36], por lo que no se trata de una lista ilustrativa y no exhaustiva de los pasos.

El proceso se repite para cada objeto en el mundo del juego. Para cada uno su respectivo «*shader* de vértices» calcula su posición final, su rotación y escala. Con base en esto se determina qué partes de cada objeto, si es que alguna, se muestran en pantalla y el «*shader* de fragmento» de cada objeto pasa a definir cómo se colorearán las partes que se muestran en pantalla según su color, profundidad y las fuentes de luz presentes, con sus respectivas sombras. Esos dos *shaders* son los pasos programables por el usuario más sencillos que define una *rendering pipeline* [11] y son donde los desarrolladores de videojuegos pueden expresar su visión artística al modificar cómo se ve su juego. [13]

Durante el proceso se crean dos resultados agregados en base a cada objeto, la imagen que será mostrada en pantalla, conocida como *frame buffer*. El segundo se conoce como *depth buffer* y es una imagen que lleva el registro de cuál es el objeto más cercano a la pantalla. Se lo crea en base a la siguiente heurística: Si el objeto para el que se está corriendo el proceso tiene una profundidad menor a la profundidad que ya está registrada en el *depth buffer*, se determina que ese objeto está enfrente de lo que estaba anteriormente. Por lo que muestra a ese objeto por sobre lo que estaba antes en esa posición y se registra la profundidad del objeto como la profundidad más cercana para que se tenga referencia de ese objeto para los demás. [17] Es importante notar que esto se hace para cada píxel que un objeto ocupará en pantalla, por lo que se pueden dar casos en los que se muestre un objeto con partes de otro protuberando de este:

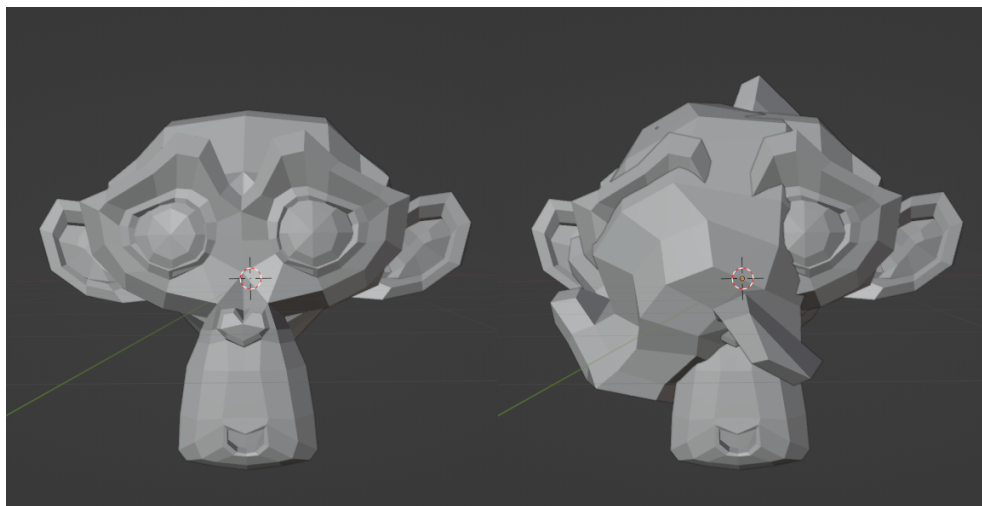


Figura 4.2: Monos de Blender demostrando el fenómeno conocido como *clipping* (discutido en el párrafo anterior). Se muestra a un mono sin el problema (izquierda) y a dos monos superpuestos demostrando el problema (derecha).

Típicamente se entiende a los *shaders* como programas que corren en una GPU. Aunque originalmente el uso de la palabra se limitaba a programas que se usan para colorear y sombrear objetos. El mismo «*shader*» viene de la técnica de dibujo y se traduce a «sombreador». [7] El desarrollo de un *shader* requiere de conocimientos de álgebra lineal [12] y es un componente importante de la producción de videojuegos, puesto que hacen posible la introducción de efectos visuales que van más allá de copiar y pegar una imagen en pantalla, en tiempo real. En un juego moderno,

son utilizados para definir la gran mayoría de lo que se termina mostrando al usuario en pantalla y su uso apoya mucho a la implementación de la línea visual y el lenguaje artístico deseados para un juego. Los tipos de efectos que se pueden lograr al usar *shaders* incluyen efectos de *post-processing*, partículas, animación de objetos suaves, simulación de fluidos e iluminación, aunque cabe resaltar que esta lista no es exhaustiva. [13]

Los efectos de *post-processing* se caracterizan por ser baratos y eficientes, pues son retoques finales que se hacen a la imagen final que se mostrará al usuario y no se aplican a objetos individuales. Esto hace que al desarrollar efectos de este tipo, se tenga menos información específica a cada objeto que se muestran en pantalla, pero se gana información holística sobre todos los objetos mostrados y la relación que estos tienen en base a su posición final en pantalla, cosa que no se tiene para los efectos que se corren un objeto a la vez.[8] Además de que, gracias a que se está trabajando con una imagen entera, es posible aplicar técnicas de procesamiento de imágenes, que se ven típicamente en aplicaciones de ciencias de datos, como lo es la detección de bordes.[3]

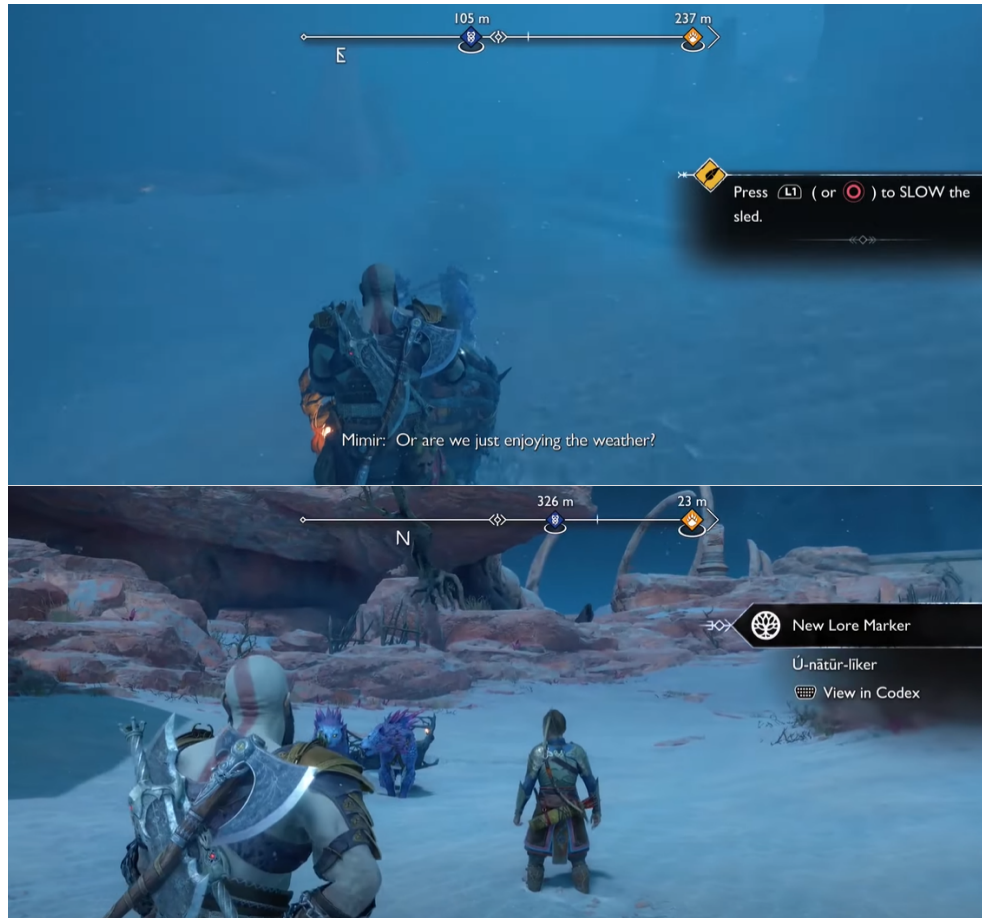


Figura 4.3: Capturas de pantalla del juego «God of War: Ragnarok» [23] que muestran una escena en la que se aplica un efecto de *post-processing* que superpone una tormenta de arena sobre del juego (arriba) y una escena en la misma locación (Alfheim) en la que no está presente la tormenta, por lo que no hace uso del efecto (abajo).

Con el objetivo de validar los temas a incluir y consecuentemente el curso propuesto, para cada tema se produjo material didáctico que fue utilizado en una cátedra que se impartió a estudiantes de quinto año de la especialización de videojuegos. Luego de recibir la cátedra, se encuestó a los estudiantes sobre si consideran que aprender el tema les aportaría valor, sobre su retroalimentación hacia el material didáctico producido y sobre si los conocimientos adquiridos por ellos al tomar los cursos de la carrera y la especialización de videojuegos les permitieron entender los temas expuestos.

Adicionalmente, se obtuvo retroalimentación de los catedráticos sobre si aporta valor incluir el tema en un curso de la especialización, si es posible, además de su retroalimentación sobre el material didáctico producido. Notablemente, esto se hizo sin que los catedráticos recibieran la cátedra correspondiente a cada tema y su retroalimentación fue tomada en forma de entrevista libre y no como una encuesta para darles mayor libertad sobre los comentarios que desearon hacer.

Originalmente se puso como meta terminar este proceso para, por lo menos, un tema y en la práctica se logró para dos. Para un tercer tema no hubo tiempo de producir material ni pasar una cátedra, por lo que se improvisó el siguiente método:

Con el objetivo de obtener los mismos resultados, se optó por reemplazar al material didáctico que no se pudo producir con un video explicativo hecho por un tercero. Luego de que los estudiantes y catedráticos vieron

el video, se les encuestó sobre si consideran que aprender el tema les aportaría valor y sobre si consideran que los conocimientos adquiridos al tomar los cursos de la especialización al tomar los cursos de la carrera y la especialización de videojuegos permitirán a los estudiantes aprender los temas. En este caso, se omitieron las preguntas que buscaban recolectar recolectar retroalimentación sobre el material didáctico producido, puesto que no tuvieron sentido cuando no se pudo producir el material.

Una bitácora de la ejecución se encuentra disponible en los anexos. B

6.1. Competencias del curso

Competencias del curso redactadas según la guía oficial [39]. Competencias específicas:

- Comprende cómo configurar software para la creación de arte digital y videojuegos.
- Crea efectos visuales haciendo uso de conocimiento sobre el funcionamiento interno de las herramientas de software que usa.
- Crea herramientas que eficientizan la manera en la que trabajan los miembros de un equipo de desarrollo de videojuegos que carecen de capacidades técnicas o que habilitan al equipo el lograr resultados previamente imposibles.

Competencias previas a tomar el curso:

- Comprende el proceso de producción de arte digital a un nivel que le permite crear piezas de arte para un videojuego.
- Comprende el proceso de producción de un videojuego a un nivel que le permita crear juegos sencillos.
- Conoce los conceptos básicos de cómo crear herramientas para que estas sean entendibles y utilizables por diferentes públicos.

-
- Recuerda conceptos generales de programación que le permiten crear código para configurar o modificar herramientas de software desarrolladas por terceros.
 - Recuerda conocimientos de álgebra lineal que le permiten desarrollar software que crea y modifica imágenes. [12]

6.2. Material didáctico producido

6.2.1. Presentación sobre la creación de *Shaders*

La presentación, proyecto de Unity de ejemplo y laboratorio propuestos para impartir el tema de Creación de *Shaders* fueron basados, en su mayoría, en las lecciones gratuitas de Freya Holmér [13], *The Book of Shaders* por Patricio Gonzalez Vivo [21] y el manual de uso de Unity [38], mientras que las otras fuentes en el listado de referencias de la presentación (ver A.3) fueron utilizadas para soportar la validez de puntos específicos.

La presentación abre con una sección de contextualización que cubre ¿Qué es un *shader*? y delimita el uso de la palabra a lo que comúnmente se conoce como un *shader* de vértices y un *shader* de fragmento, pues son el tipo de *shader* en que se concentra la presentación. Se termina esta sección haciendo un paréntesis técnico, que sirve dos propósitos, esclarecer con un ejemplo visual la diferencia entre los dos tipos de *shaders*; y resaltar el hecho de que computar datos necesarios para efectos visuales en un *shader* de vértices es menos demandante para el hardware, pero resulta en una menor calidad. Este último punto se ilustra en la novena diapositiva:

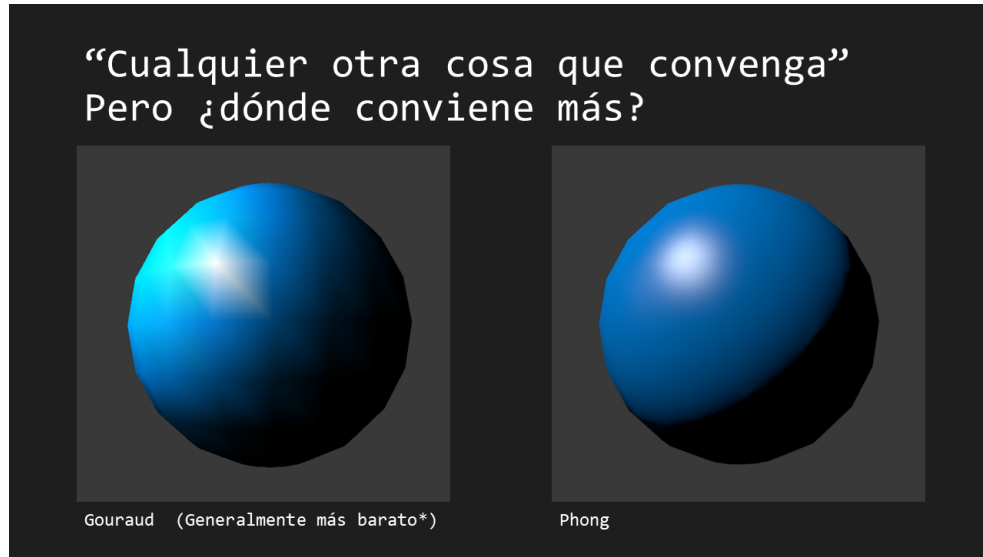


Figura 6.1: Novena diapositiva de la presentación creada para impartir el tema de Creación de *shaders* que ilustra la diferencia entre usar un *shader* de vértices y un *shader* de fragmento para computar un efecto de iluminación.

Luego se discuten los diferentes tipos de herramientas que se pueden usar para crear un *shader* y configurar interfaces que permiten parametrizarlos para crear herramientas reutilizables. Además, se resaltó el hecho de que utilizar lenguajes diseñados para el uso de GPUs en la creación de *shaders*, como GLSL [10] y HLSL [18], es ventajoso porque estos son agnósticos a las *game engines* que los utilizan. Cosa que no se cumple para las herramientas de programación visual, como el *Shader Graph* de Unity [37] que se utilizan por comodidad, pero que dependen de las *engines* por ser parte de las mismas.

Se continuó contextualizando la disciplina del álgebra lineal en la creación de *shaders*, pues esta permite representar discretamente imágenes y modelos 3D, lo que facilita la computación paralela de un *shader*. [12] Esto se logró al hacer una equivalencia entre los conceptos de cada disciplina, por ejemplo, un color en formato RGBA puede ser representado como un vector en R^4 e imágenes pueden ser representadas como matrices de colores, o bien, vectores en R^4 . Esto se hizo con ayuda de las diapositivas 13 a 22. (ver A.3) Finalmente, se discutió sobre las implicaciones de procesar este álgebra lineal en una GPU, el que son excelentes para computar un *shader* paralelamente [20], pues este es su propósito original, pero introducen limitantes en cuánto a la cantidad de información que se puede procesar a la vez, pues hace falta copiarla desde RAM,

procesarla en la GPU y copiarla de regreso [15]:

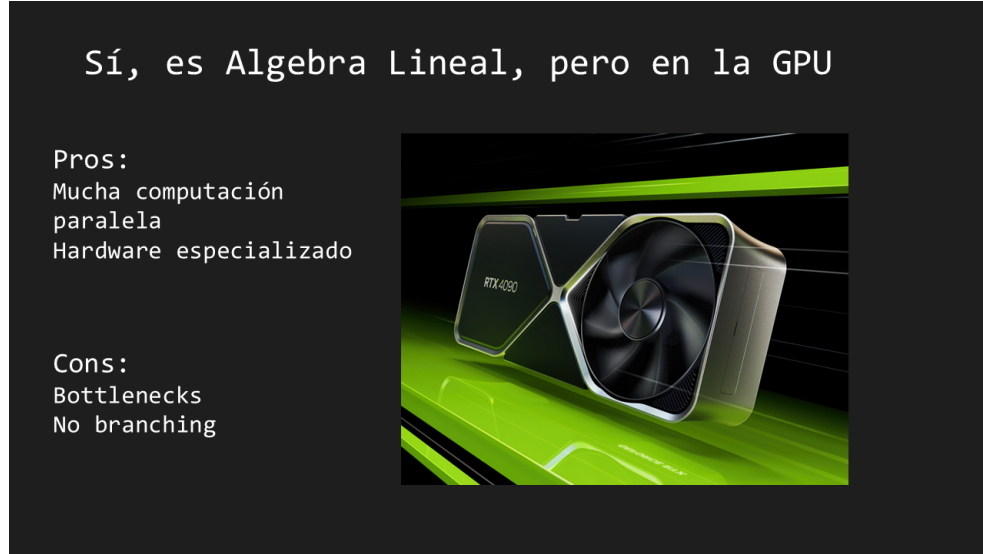


Figura 6.2: Dispositiva 22 de la presentación creada para el tema de creación de *shaders* que ilustra rápidamente las ventajas (Paralelismo y Hardware especializado) y desventajas (Crea un cuello de botella y limita el uso de bifurcaciones en el flujo de ejecución de un algoritmo) que se tienen al computar operaciones de álgebra lineal en una GPU.

Se continuó cubriendo una lista de funciones de HLSL (ver A.3.2), que si bien no es exhaustiva [19], compila las funciones que son comúnmente útiles en el desarrollo de *shaders* de vértice y fragmento [13]. Además, se cubrieron técnicas que utilizan las funciones discutidas para crear efectos diferentes (ver A.3.3), así como configuraciones de Unity que sirven para cambiar la manera en la que se muestran los objetos sobre los que actúa un *shader*, como por ejemplo, transparencia (ver A.3.4). Todos estos temas fueron explicados con la ayuda visual de mostrar sus efectos en vivo en el repositorio de demostración o con videos integrados en la presentación (ver A.3.1).

La presentación termina con la demostración principal y la asignación del laboratorio, que es reproducir un efecto parecido. La demostración consta de un efecto que se logra con *shaders* de vértices y fragmentos, y que aplica algunas de las técnicas y conceptos matemáticos cubiertos. El efecto es un *shader* que simula agua que fue creado deformando un plano en base a texturas que determinan qué tan altas deben ser las olas, hacia qué dirección deben fluir las olas y qué partes del plano se deben deformar. Mientras que las olas se animaron en base a senos y cosenos computados en base a las coordenadas UV del plano. Finalmente, en base a la altura

final de las olas, se añadió un efecto de reflejo especular caricaturesco que las ilumina en sus crestas.

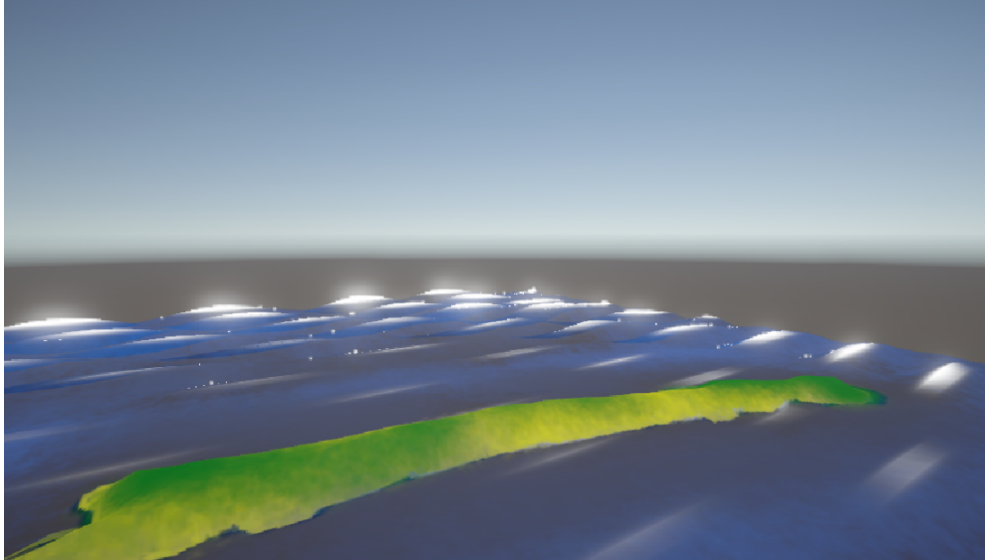


Figura 6.3: Demostración creada para impartir el tema de creación de *shaders*, muestra un efecto de simulación de olas en una isla, animado en base a senos, cosenos y parametrizado en base a texturas que indican cómo y qué se debe deformar durante la animación.

6.2.2. Presentación sobre la creación de efectos de *post-processing*

La presentación que se hizo para impartir el tema de Creación de efectos de *post-processing* fue basada en la teoría y el ejemplo práctico presentado por Matt Gambell [8] y en el manual de Unity sobre efectos de *post-processing* [35].

La presentación comienza estableciendo qué es un efecto de *post-processing* y cómo se ejecuta un efecto de este tipo en el contexto de una *rendering pipeline* [11]. Este último punto es importante, porque determina una de las mayores diferencias entre los efectos de *post-processing* y efectos de otro tipo, la información sobre la cuál se busca operar y consecuentemente, el tipo de operaciones a computar y sus resultados. En concreto, se trata de la imagen producida por la *rendering pipeline*, a la cual se le hacen retoques finales antes de presentarla al usuario. Siguiendo el constructivismo [26], el objetivo de introducir estos conceptos es plantear las preguntas: «¿Para qué sirve hacer un efecto de este tipo?» y «¿Cómo podemos pasar de operar sobre objetos para producir imágenes a operar sobre imágenes para producir imágenes?».

Para responder la primer pregunta, la presentación utiliza dos ejemplos,

el de Gambell [8] y un ejemplo creado para la presentación que busca imitar parte de la línea visual usada para acentuar el foco de la cámara de la película, *Spider-Man: Into the Spider-Verse* [29]. Pero primero, introduce las herramientas usadas para crearlos, la URP de Unity. Esto sirve dos propósitos, apoyar a los estudiantes a comprender los ejemplos y cómo reproducirlos e introducir una tercer enseñanza clave, el estándar que maneja la industria para efectos de este tipo. Este se trata de la idea de que los efectos de este tipo deben ser pensados, no como un método para crear un efecto visual en específico, sino como herramientas que pueden ser configuradas y reutilizadas por artistas y diseñadores para redefinir la línea visual de un juego durante su producción.



Figura 6.4: Dispositiva 5 de la presentación creada para el tema de creación de efectos de *post-processing* que muestra el propósito de un componente de volumen, proveer una manera de configurar un efecto de *post-processing* por medio de controles amigables a usuarios no técnicos.

La presentación continua mostrando los dos ejemplos y explicando el ejemplo creado para la presentación. La presentación solo cuenta con diapositivas representativas del código que fungen como marcadores de secciones e introducen el propósito del código, por lo que se espera que el presentador muestre el código y explique el código disponible en el repositorio de ejemplo (ver A.2.1). Primero se explica el propósito del componente de volumen, que sirve para definir la interfaz con la que los usuarios configurarán el producto final. Luego se toca el *Scriptable Render Feature*, que sirve para introducir un paso en la *rendering pipeline* de Unity. El propósito de este paso, es definir cuándo se corre el ejemplo y

pasar los parámetros especificados por el volumen al ejemplo. Finalmente, se cubre el *Scriptable Render Pass*, que define cómo correr el efecto de *post-processing* en sí, el cuál es un *shader*. [8]

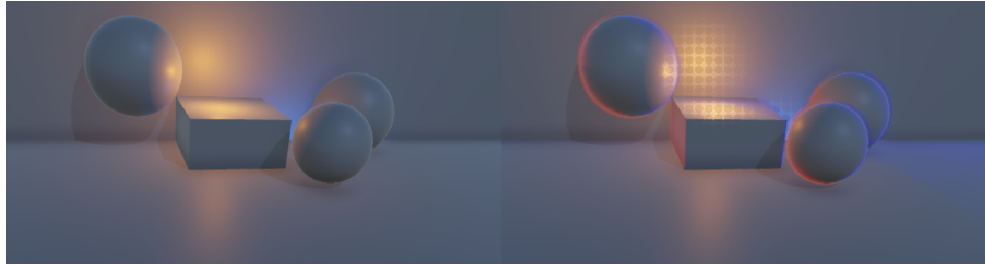


Figura 6.5: Demostración creada para impartir el tema de creación de *shaders*, muestra una comparación de la escena sin los efectos de aplicados (izquierda) y con los efectos aplicados (derecha).

La presentación culmina explicando a detalle el *shader* creado para realizar el ejemplo y cómo este es diferente de un *shader* de vértices o de fragmento. En sí, sólo consiste de una función que actúa como un *shader* de fragmento, pero que recibe la imagen a mostrar en pantalla como una textura y debe calcular los cambios a aplicar, escribirlos y devolverlos para que reemplacen la imagen final que verá el usuario. Dado que el objetivo del ejemplo es ocasionalmente resaltar los bordes de los objetos, también se usa el *depth buffer* para usar el algoritmo Sobel de detección de bordes para distinguir dónde termina y comienza un objeto con base en la profundidad del agregado de todos los objetos. [3] Es este último detalle el que ayuda a responder la pregunta «¿Cómo podemos pasar de operar sobre objetos para producir imágenes a operar sobre imágenes para producir imágenes?» al mostrar que es posible aplicar técnicas que pertenecen al área del procesamiento de imágenes [28].

6.3. Diseño preliminar del curso

A continuación se muestra una tabla que detalla un cronograma para el curso propuesto, incluyendo los temas de Desarrollo de *shaders* y Desarrollo de efectos de *post-processing*, discutidos en las secciones siguientes. En vez del tema de Desarrollo de efectos visuales avanzados, se optó incluir un tema que lo engloba y que tiene un alcance más adecuado para un módulo de curso, Desarrollo de herramientas. El resto de temas del módulo fueron sugeridos por Dennis Aldana al discutir sobre las nece-

sidades de la especialización 6.18 o, en el caso de las curvas de Bézier, tomados de casos de uso reales [14].

Asumiendo semanas en las que se cuenta con dos sesiones de clase, se utilizaron 15 para la impartición de temas y entrega de laboratorios y proyectos. Además se dejaron 2 semanas extra que, bajo el criterio del instructor, se puede usar para aplazar tiempos de entrega, reforzar temas, ajustarse a eventualidades como emergencias o feriados o para revisión de notas. Esta flexibilidad es especialmente valiosa en el contexto de un curso nuevo, pues le permite al instructor compensar por su inherente falta de experiencia impartiendo el curso y experimentar para mejorar el curso en el futuro. Esto se conforma con los 5 meses de clase en cada semestre de la universidad, tomando en cuenta la semana de descanso y que las clases comienzan en la segunda semana del primer mes y terminan en la tercer semana del último.

Módulo	Laboratorio	Tema	Semana y actividad				
			1	2	3	4	
Introducción al curso		Propósito del curso, modalidad y calificación	Clase				
Desarrollo de Shaders	1	Shaders de vértice y fragmento	Clase	Clase			
		Álgebra lineal para transformación de geometría y color	Clase	Clase			
	2	Semánticas de HLSL (Manipulación de datos de entrada)			Clase	Clase	
		Uso de normales de mundo y objeto				Clase	
		Uso de UVs				Clase	
				Semana y actividad			
				5	6	7	8
	3	Shaders animados	Shaders de emisión	Clase			
			Shaders de transparencia y uso de <i>depth buffer</i>		Clase		
		Face culling				Clase	
						Clase	
Desarrollo de efectos visuales	4	Efectos de <i>post-processing</i> (Uso de <i>Frame buffer</i>)				Clase	
						Clase	
			Semana y actividad				
			9	10	11	12	
	5	Investigación y replicación de efecto de <i>post-processing</i>		Trabajo en clase			
Desarrollo de herramientas	6	Principios y objetivos del desarrollo de herramientas				Clase	
		Serialización y persistencia en una <i>Game Engine</i>				Clase	
		Desarrollo de GUIs para el editor				Clase	
		Herramientas para efectos visuales				Clase	
			Semana y actividad				
			13	14	15	16	
	Herramientas para creación de <i>assets</i> y replicación	Clase		Entrega de proyecto 3	Revisión de notas		
	Herramientas basadas en curvas de Bézier		Clase				

Figura 6.6: Tabla de cronograma preliminar para el curso.

Es importante notar que el cronograma incluye columnas para los módulos y laboratorios del curso. Cada proyecto corresponde a cada módulo del curso, mientras que cada laboratorio corresponde a los temas que se

muestran al lado. Se espera que cada proyecto se entregue alrededor de un mes luego de su asignación luego de que se cubran los temas pertenecientes al módulo, sin embargo, esto no es posible para el último proyecto por lo que este debe ser asignado previo a que se cubran todos los temas del módulo. Para cada laboratorio, la entrega será luego de dos semanas de su asignación, la cual será cuándo se terminen de cubrir los temas asociados a cada laboratorio.

Tipo de actividad	Cantidad	Ponderación	
		Acumulada	Individual
Proyecto	3	60%	20%
Laboratorio	6	40%	7%

Figura 6.7: Tabla de ponderación preliminar para el curso.

6.4. Retroalimentación obtenida

La retroalimentación presentada en este capítulo fue recolectada luego de que los estudiantes recibieron una cátedra usando tres presentaciones, Introducción al Arte Técnico, Introducción a *shaders* y efectos de *post-processing*, para referencia están disponibles en A.1, A.3 y A.2, respectivamente.

6.4.1. Creación de *shaders*

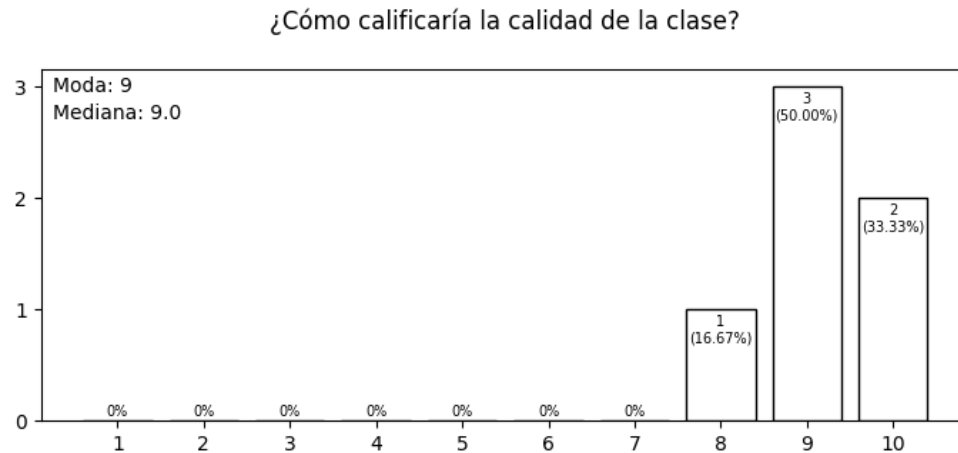


Figura 6.8: Gráfico de la pregunta «¿Cómo calificaría la calidad de la clase?» que fue hecha en forma de escala Likert de 1 (Mala) a 10 (Buena). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

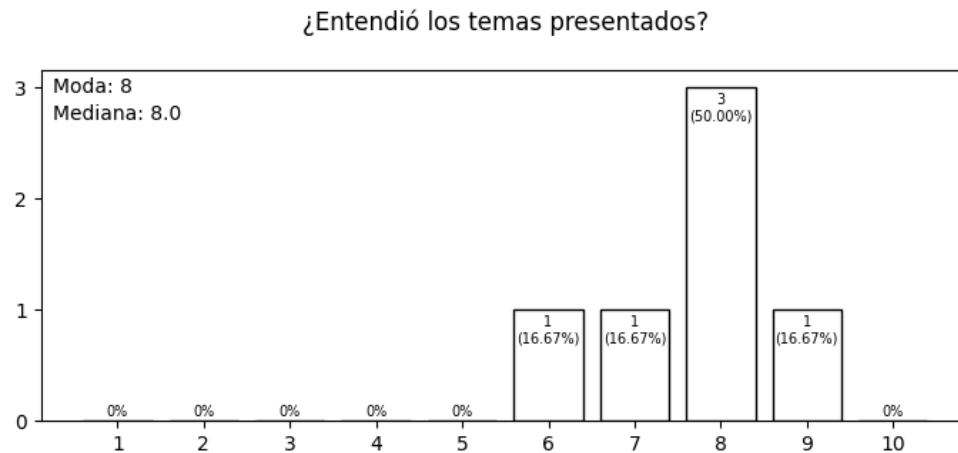


Figura 6.9: Gráfico de la pregunta «¿Entendió los temas presentados?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.5. *Respuestas abiertas a la pregunta, «¿Qué cambiaría de la clase? ¿Por qué?»:*

- «*Por tantos tips que diste me hubiera gustado que también la hubieras grabado, ya que no tengo buena memoria.*»
- «*Nada, la explicación estuvo bien.*»

- «*Me gustaría que alguno de los ejemplos (por ejemplo hacer un objeto con glowing) se hiciera desde cero en un proyecto nuevo, para aprender como hacerlo.*»
- «*La acortaría un poco porque después de un tiempo perdí la atención.*»

Pregunta de opción múltiple 6.6. *Respuestas a la pregunta, «¿Cuál cree que es el alcance apropiado para los temas impartidos?»:*

- «*Módulo de curso.*» (4)
- «*Sección de módulo.*» (1)
- «*Curso*» (1)

¿Qué tan importante considera la inclusión de estos ejemplos para la retención de los temas?

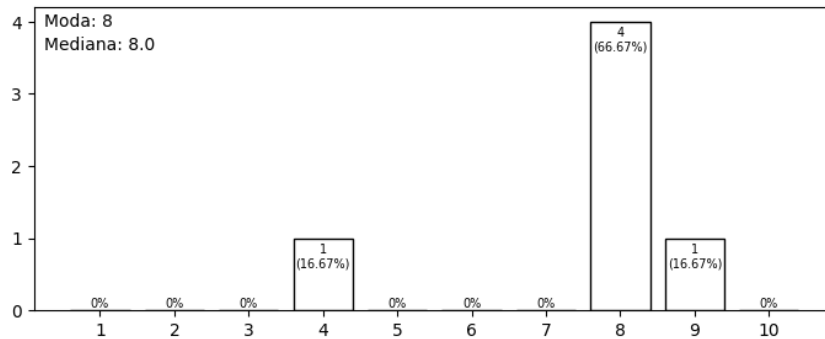


Figura 6.10: Gráfico de la pregunta «Se presentaron varios aspectos técnicos sobre cómo se puede configurar herramientas como Unity para modificar el comportamiento de un *Shader* (Transparencia, *Blending*, *Face Culling*, *Depth Testing*, Máscaras de color), pero muy pocos ejemplos de cómo esto sería aplicable. ¿Qué tan importante considera la inclusión de estos ejemplos para la retención de los temas?» que fue hecha en forma de escala Likert de 1 (De poca importancia) a 10 (De suma importancia). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta de opción múltiple 6.7. *Respuestas a la pregunta, «¿Cree que es más útil aprender los temas a base de ejemplos de casos de uso?»:*

- «*Sería mejor utilizar ejemplos de aplicaciones y recibir clases magistrales sobre aspectos técnicos*» (6)

¿Cree que ese conocimiento le sería útil como estudiante de la especialización de videojuegos?

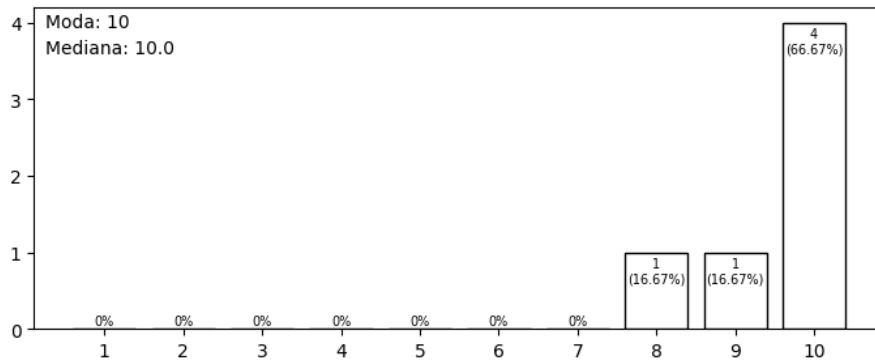


Figura 6.11: Gráfico de la pregunta «¿Cree que ese conocimiento [cómo crear *shaders*] le sería útil como estudiante de la especialización de videojuegos?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, sumamente útil). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

¿Cree que los conocimientos impartidos en los cursos de la especialización de videojuegos y en la carrera en general le dan una buena base para aprender sobre la creación y configuración de Shaders?

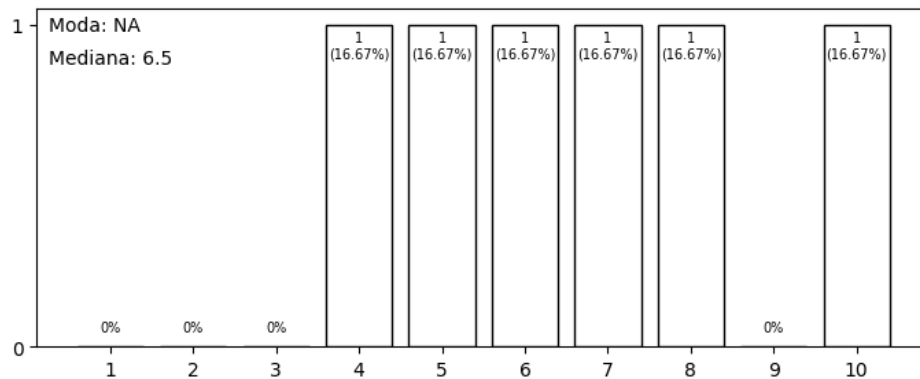


Figura 6.12: Gráfico de la pregunta «¿Cree que los conocimientos impartidos en los cursos de la especialización de videojuegos y en la carrera en general le dan una buena base para aprender sobre la creación y configuración de *shaders*?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.8. *Respuestas abiertas a la pregunta, «¿Qué cursos considera necesarios para aprender sobre la creación y configuración de *shaders*?»:*

- «Principalmente álgebra lineal ya que se debe tener una buena base para poder comprender a profundidad cómo manejar ciertas herra-

mientas. Creo que el curso de Diseño de videojuegos también puede ser útil para dar una explicación inicial sobre este tema.»

- «Graficas»*
- «Todos los de cálculo, programación de videojuegos para tener el contexto de unity y un poco la de animación, para que las personas puedan crear sus props y así configurarlo con shaders»*
- «Progra de juegos y álgebra lineal»*
- «Poo, Progra de videojuegos y todas las mates»*
- «Programación de vídeo juegos»*

Comentarios de catedrático 6.9. *Comentarios dados por Dennis Aldana luego de ver la presentación, sin recibir la clase:*

«En la tercera [diapositiva] mencionas que el concepto de shader ya no aplica mucho, solo asegúrate de mencionar su origen en sombreado. También puedes exponer su importancia, por ejemplo. Cómo Pixar obtuvo un Oscar por su software el único software que ha obtenido un Oscar. ...¿En esta [presentación] ya no vas a hablar de un render pipeline?»

«Los shaders esencialmente te dejan programar o personalizar un pipeline antes dependías de las funciones que programara el creador de la tarjeta de video.»

«De ahí, la presentación está bien, aunque veo algo cargada la sección de preguntas importantes, pero depende de como lo uses en la clase. Ten cuidado que no sea mucha sobrecarga de información.»

«En la sección de código de HLSL, te recomiendo empezar con un hello world o algunos ejemplos exitosos, antes del debugging.»

«Lo demás está bien, solo nuevamente tené cuidado de no cargar mucho a tu audiencia... el contenido que estas tratando de mostrar es como un mes de clases.»

«Veo que te enfocas en HLSL, pero entiendo que la mayoría de usuarios de Unity usan el UI de nodos o algún plugin. ¿No será mejor apoyarse en estas herramientas? Yo prefiero [la] programación directa de shaders la verdad, pero creo que soy la minoría.»

En respuesta a la pregunta: ¿Crees que se deba cubrir a fondo el tema de creación de shaders en la especialización de videojuegos? ¿Por qué?»

«Sí, los shaders son una parte importante del desarrollo de juegos. Hay muchas cosas que son muy costosas sin shaders, tanto en assets, como en efectos.»

Comentarios de catedrática 6.10. *Comentarios dados por Alhvi Balcarcel luego de ver la presentación, sin recibir la clase:*

«El contenido me pareció muy bien... tal vez lo que faltaría es una forma en la que los alumnos puedan aplicar lo que van aprendiendo como tareas o proyectos. Creo que al final a muchos se les olvida lo que uno les dice, es hasta que se meten a programar y a toparse con clavos, cuando van aprendiendo. Por eso creo que es importante esa parte.»

En respuesta a la pregunta «Qué opina de los temas que cubiertos en las presentaciones, creación de shaders y Creación de efectos de post-processing. ¿Cree que vale la pena cubrirlos en la especialización? ¿Cree que se pueda?»:

«Definitivamente creo que son temas interesantes para la especialización y que si se deben incluir. Lo que no estoy segura es si debería ser un curso por aparte o si se deben integrar al de programación de videojuegos. Creo que dependería mucho del resto de temas para el posible nuevo curso»

6.10.1. Efectos de *post-processing*

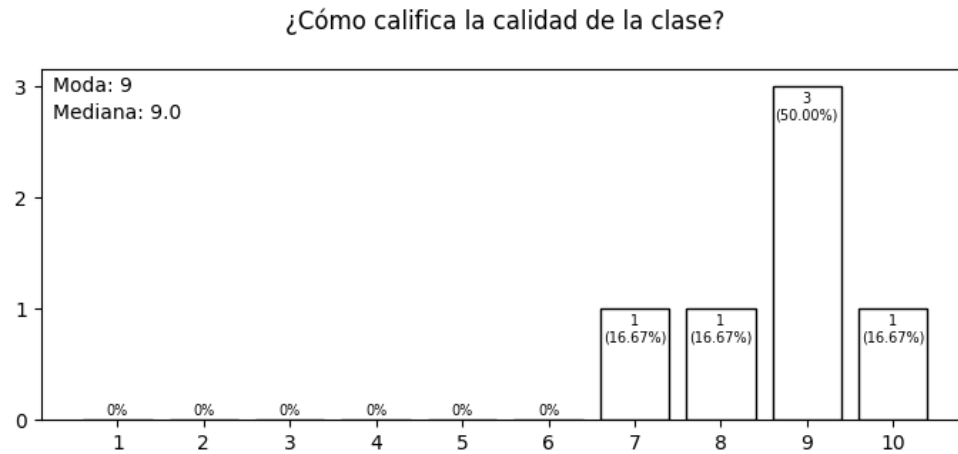


Figura 6.13: Gráfico de la pregunta «¿Cómo califica la calidad de la clase?» que fue hecha en forma de escala Likert de 1 (Mala) a 10 (Excelente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

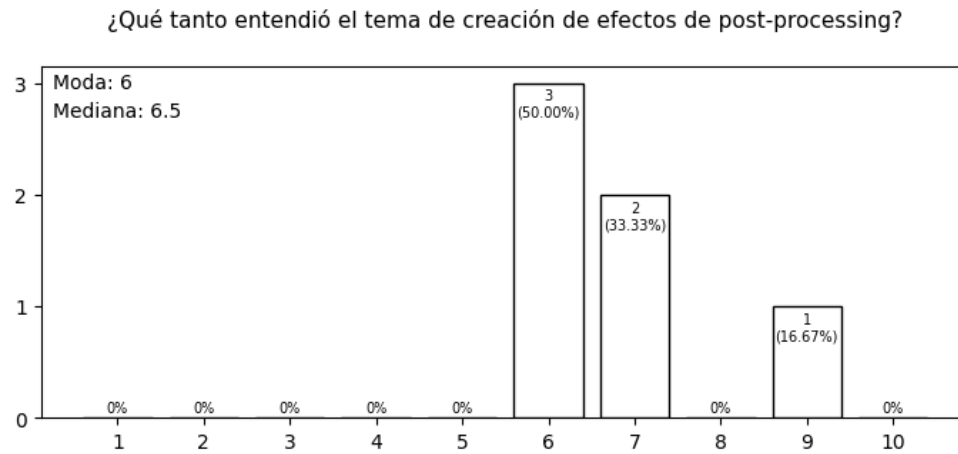


Figura 6.14: Gráfico de la pregunta «¿Qué tanto entendió el tema de creación de efectos de *post-processing*?» que fue hecha en forma de escala Likert de 1 (Para nada) a 10 (Completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

¿Se considera capaz de realizar el ejercicio presentado al final de la clase?

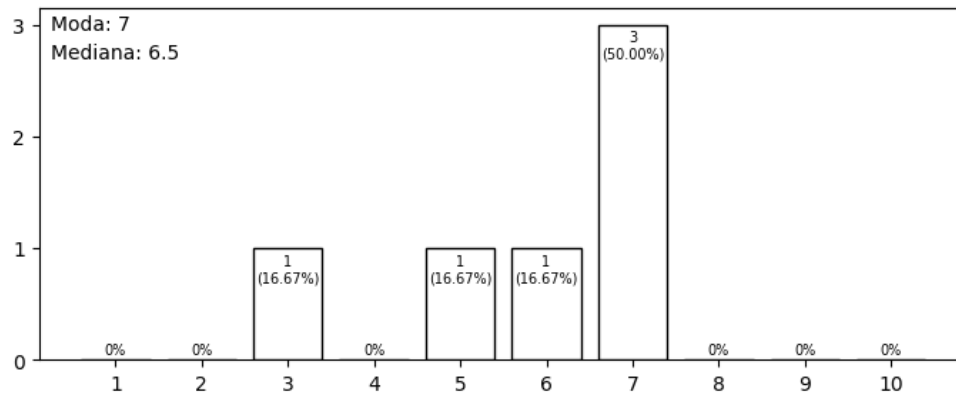


Figura 6.15: Gráfico de la pregunta «¿Se considera capaz de realizar el ejercicio presentado al final de la clase?» que fue hecha en forma de escala Likert de 1 (Para nada) a 10 (Completamente). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.11. *Respuestas a la pregunta «¿Con qué temas le queda duda?»:*

- «Por el momento considero que no tengo ninguna duda.»
- «[render]ring pipelines.»
- «La implementación por no aplicarlo todavía.»
- «Shaders.»
- «La teoría estuvo muy bien para todos los temas, pero me perdí con los ejemplos.»
- «Mas que nada los temas de matemática, me hubiera servido un breve recordatorio.»

¿Considera que los temas presentados le serían útiles a los futuros estudiantes de la especialización?

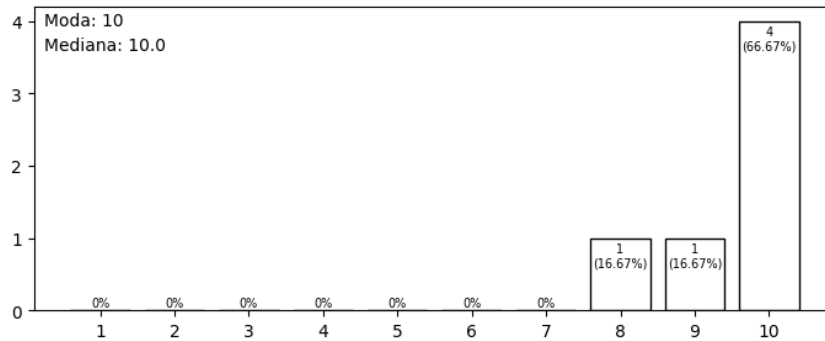


Figura 6.16: Gráfico de la pregunta «¿Considera que los temas presentados le serían útiles a los futuros estudiantes de la especialización?» que fue hecha en forma de escala Likert de 1 (Son irrelevantes) a 10 (Su inclusión en el programa sería muy buena). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.12. *Respuestas a pregunta «Justifique su respuesta anterior» (Referente a la pregunta ¿Considera que los temas presentados le serían útiles a los futuros estudiantes de la especialización?, ver 6.16):*

- «Ayuda a ver más allá de lo que puede enseñarnos YouTube o algún curso de Udemy.»
- «Es un campo que no se trabaja durante el pénsum para la aplicación en juegos.»
- «Este es el tipo de temas que le agregan más valor a un videojuego, no solo en cuando las mecánicas sino también para poder aprender mejor sobre otras herramientas que Unity presenta..»
- «Sí, es un curso muy específico de la rama de gráficas entonces tal vez sea mas útil para los que busquen especializarse en ese campo.»
- «En las clases que llevamos de juegos no nos enseñan mucho sobre todo lo que se puede hacer al momento de utilizar Unity, y por ende los juegos que hacemos son muy básicos.»
- «Me hubiera gustado saber más sobre estos temas, creo que la calidad de mis juegos sería mucho mejor.»

Pregunta de opción múltiple 6.13. *Respuestas a pregunta «Como estudiante de la especialización de videojuegos de 5to año, ¿Su conocimiento previo le fue suficiente para entender los temas?»:*

-
- «Creo haberlo visto en su momento, pero no lo recuerdo.» (3)
 - «Me faltó contexto en algunas cosas.» (2)
 - «Creo que nos falta un poco de base.» (1)

Pregunta abierta 6.14. *Respuestas a pregunta «¿Qué temas incluidos en otros cursos de la especialización (si es que alguno), considera necesarios para comprender lo presentado en clase?»:*

- «Álgebra lineal.»
- «Gráficas (no especialización) y progra avanzada para conocer mas el engine en este caso, Unity.»
- «Un poco más sobre shaders estaría genial para poder comprender en su totalidad los temas.»
- «Un repaso de gráficas y la mate detrás.»
- «shaders y luces en escenas.»
- «No entiendo muy bien la pregunta.»

Comentarios de catedrático 6.15. *Comentarios dados por Dennis Aldana luego de ver la presentación, sin recibir la clase:*

«Hola ya revisé lo que me mandaste y esta bastante bien hecho. Como estas manejando los conceptos aparte y los ejemplos en Unity está excelente.» Ahora, algunos comentarios puntuales: «Tu presentación está bien hecha porque conecta sus temas, como que es post-procesing -> que es el proceso de renderizado -> rendering pipeline -> URP. Pero el salto a Unity y al componente de volumen se siente grande. ¿Qué tiene que ver esto con lo que estamos hablando?»

Otra cosa que sentí requería profundizar un poco más es el concepto de render pipeline. Está bien que lo generalices pero creo que deberías elegir los pasos del pipeline que vas a estar usando en Unity y explicar sus [entradas y salidas] paso por paso. Luego, para cada paso, ¿cuál es su análogo en Unity?»

«Los ejemplos de código que pusiste, no estoy seguro si el código es solo ilustrativo, pero sea como sea, los estudiantes lo van a tratar de leer. Así que si es ilustrativo, hazlo mas pequeño, y si no lo es, hazlo más grande y mas claro aunque gastes mas diapositivas»

«Realmente no hay un beneficio de tener pocas diapositivas nunca, si tienes más, solo pasas mas rápido en el mismo tiempo.»

«En tu ejemplo, mencionas que el componente de volumen en código define los parámetros que se muestran en el inspector, pero creo que sería útil ver lado a lado el ejemplo»

«Igual el tema de shaders es gigante y lo vi muy simplificado. Incluso considera romper render pipeline / Unity / shaders en varios módulos separados para la clase. Todos son temas considerables.»

«Por último, considera reducir la teoría a favor de un enfoque más práctico en el que los estudiantes deben hacer algo y vos solo les tomas la mano un poco en el camino. Todo esto que expones no se les va a quedar de todas maneras hasta que lo usen.»

En respuesta a la pregunta «¿Cree que valga la pena incluir el tema de Creación de efectos de post-processing en la especialización?»

Paráfrasis de una conversación en persona:

Sí, creo que vale la pena. La ventaja con ese tema es que es corto, pero los estudiantes necesitan tener bases en rendering pipelines y shaders.

Comentarios de catedrática 6.16. *Comentarios dados por Alhvi Balcarcel luego de ver la presentación, sin recibir la clase:*

«Me gustó bastante, aunque me hubiera gustado ver el demo»

En respuesta a la pregunta «Qué opina de los temas que cubiertos en las presentaciones, creación de shaders y Creación de efectos de post-processing. ¿Cree que vale la pena cubrirlos en la especialización? ¿Cree que se pueda?»:

«Definitivamente creo que son temas interesantes para la especialización y que si se deben incluir. Lo que no estoy segura es si debería ser un curso por aparte o si se deben integrar al de programación de videojuegos. Creo que dependería mucho del resto de temas para el posible nuevo curso»

6.16.1. Efectos visuales avanzados haciendo uso de múltiples herramientas

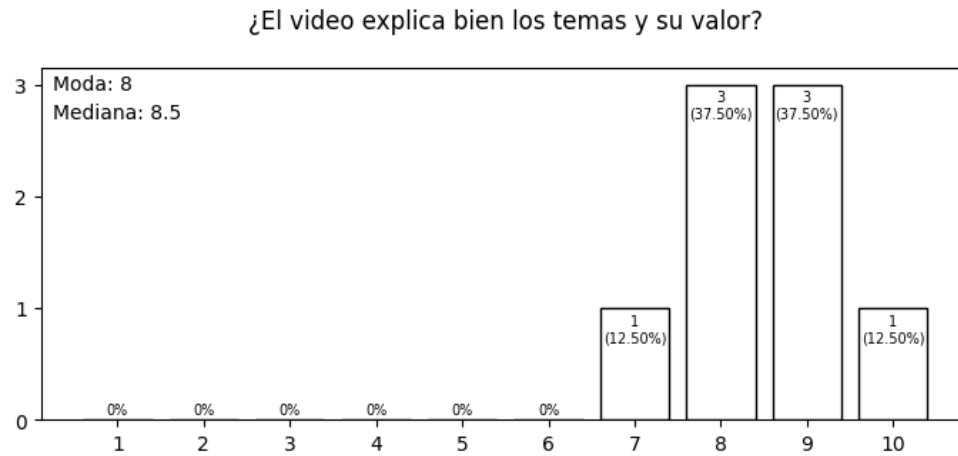


Figura 6.17: Gráfico de la pregunta «¿El video explica bien los temas y su valor?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, es muy claro). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos)

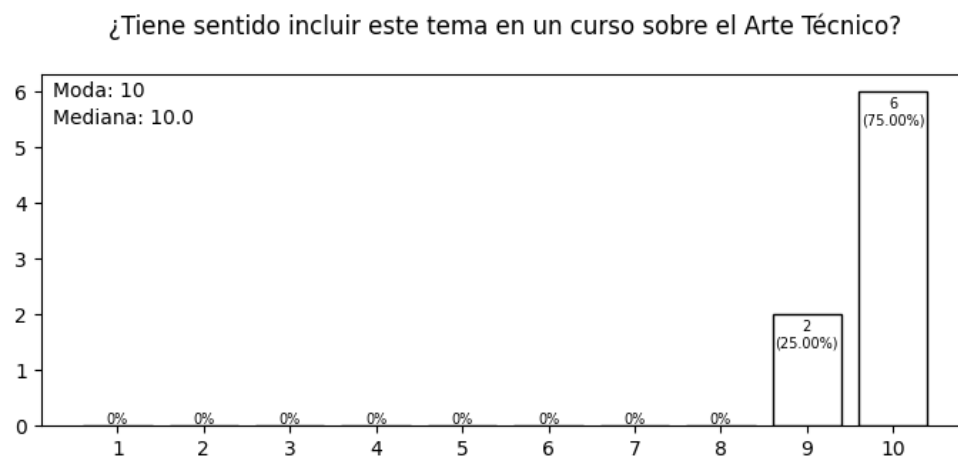


Figura 6.18: Gráfico de la pregunta «¿Tiene sentido incluir este tema en un curso sobre el Arte Técnico?» que fue hecha en forma de escala Likert de 1 (No, no tiene sentido) a 10 (Sí tiene sentido). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos)

¿Aprender sobre estas técnicas aporta valor a un estudiante de la especialización de videojuegos?

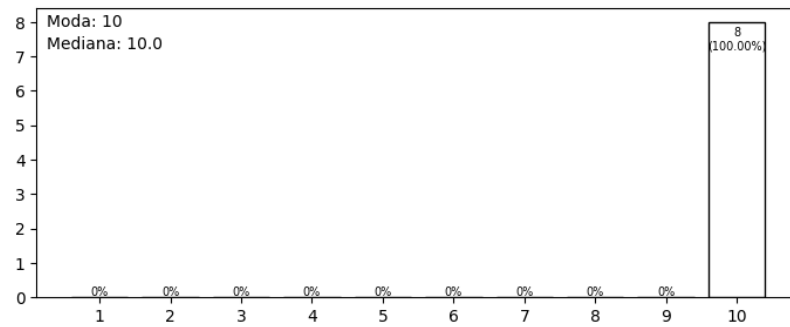


Figura 6.19: Gráfico de la pregunta «¿Aprender sobre estas técnicas aporta valor a un estudiante de la especialización de videojuegos?» que fue hecha en forma de escala Likert de 1 (No, para nada) a 10 (Sí, es sumamente útil). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos)

Pregunta de opción múltiple 6.17. *Respuestas a pregunta «Este tema entra dentro de la categoría del uso de herramientas de scripting dentro de herramientas de arte y game engines para crear efectos previamente no posibles. ¿Cuánta importancia deberían tener estos temas dentro de un curso de Arte Técnico?»:*

- «Parte de un módulo (Unas cuantas lecciones).» (5)
- «Un módulo de curso.» (1)

Respuestas de catedráticos:

- «Un módulo de curso.» (2)

¿Considera necesario impartir estos temas en relación a herramientas específicas (Blender, Maya, Unity, Godot, Unreal...)?

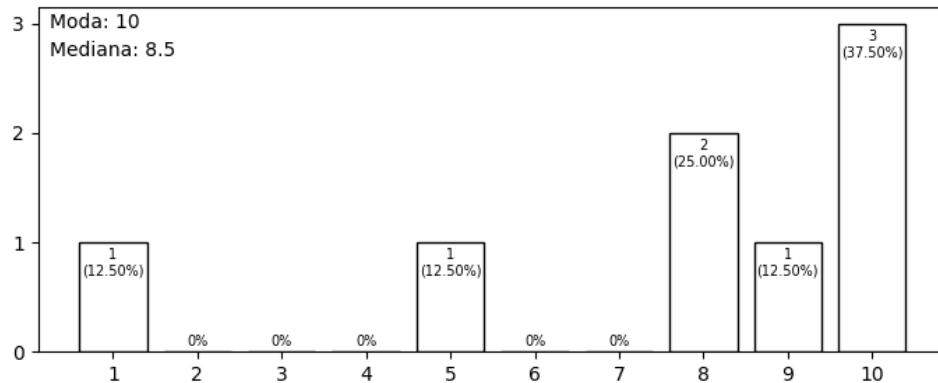


Figura 6.20: Gráfico de la pregunta «¿Considera necesario impartir estos temas en relación a herramientas específicas (Blender, Maya, Unity, Godot, Unreal...)?» que fue hecha en forma de escala Likert de 1 (No, los fundamentos son agnósticos a las herramientas) a 10 (Sí, se requiere un ejemplo práctico en una herramienta para comprender los temas). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos)

¿Considera que los estudiantes de la especialización de videojuegos tienen los conocimientos previos necesarios para aplicar los temas?

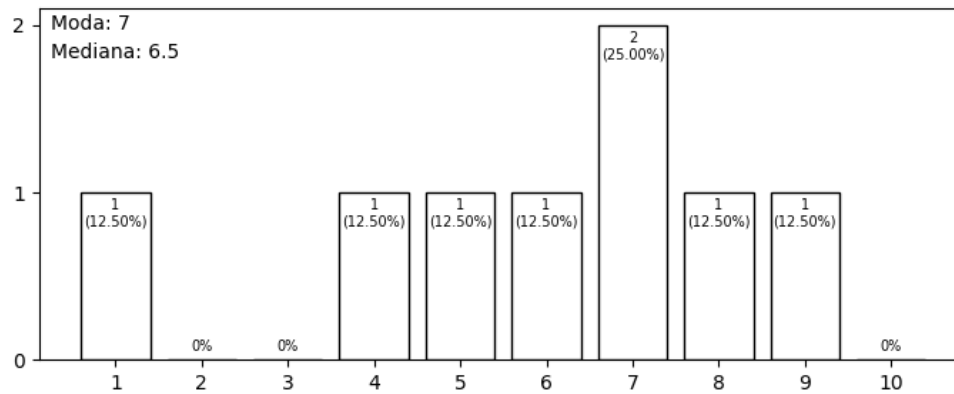


Figura 6.21: Gráfico de la pregunta «¿Considera que los estudiantes de la especialización de videojuegos tienen los conocimientos previos necesarios para aplicar los temas?» que fue hecha en forma de escala Likert de 1 (No, requieren muchos temas base que no están incluidos en los cursos del programa) a 10 (Sí, son capaces de aplicarlo solo con lo que aprendieron en la especialización y la carrera en general). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda. (Incluyendo las respuestas de dos catedráticos)

6.18. Retroalimentación a cursos actuales de la especialización de videojuegos

6.18.1. Retroalimentación para el curso de diseño de videojuegos

¿Considera que el Diseño de videojuegos le da las herramientas necesarias para construir una carrera en el área que aborda?

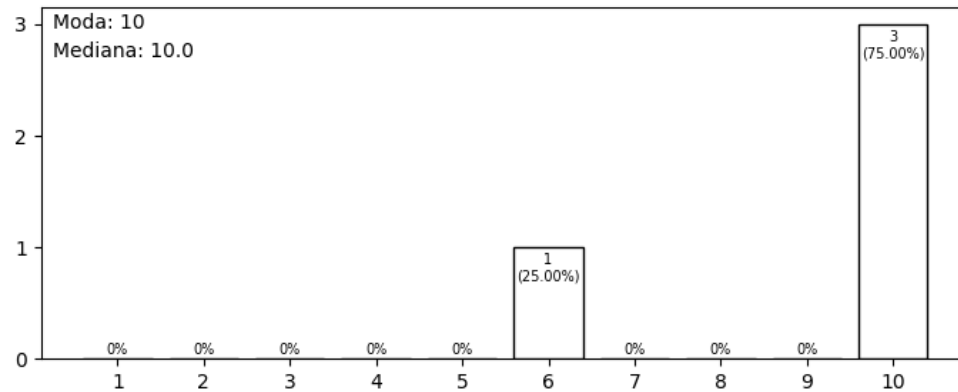


Figura 6.22: Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Diseño de videojuegos) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

¿Cree que los contenidos del curso de Diseño de videojuegos justifican su inclusión en el programa?

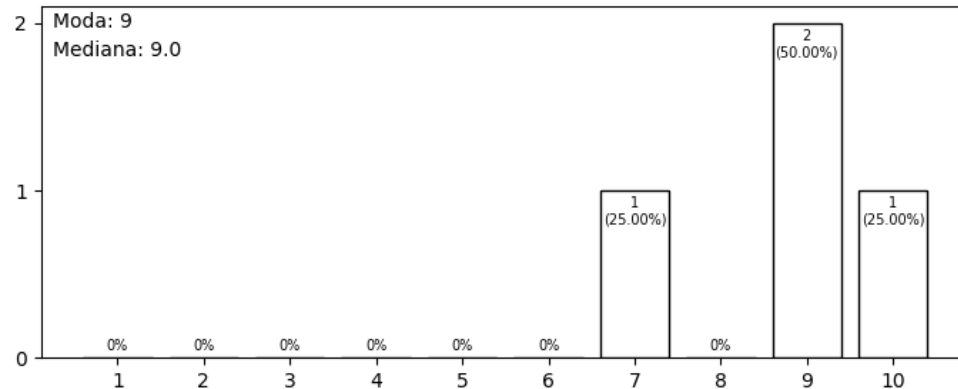


Figura 6.23: Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Diseño de videojuegos) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.19. *Respuestas a pregunta «¿Cree que el curso está bien enfocado para cubrir los temas de mayor importancia en el área que aborda? ¿Por qué?» en referencia al curso de Diseño de videojuegos:*

- *«Es bastante completo.»*
- *«Sí considero que es una buena base, ya que contiene mucha información relevante como las presentaciones Pitch para vender una idea, o la misma generación de ideas. Asimismo, enseña los elementos necesarios para completar el diseño de un videojuego, concentrándose especialmente en las mecánicas.»*
- *«Sí, el programa del curso abordar muchos de los temas de diseño de videojuegos. Al final del curso ya se tiene una idea de como iniciar un proyecto.»*
- *«No, porque creo que la falta de un curso previo que permita tener comprensión básica de programación de juegos, pues el que se cursó, solo fue de medio semestre.»*

Pregunta abierta 6.20. *Respuestas a pregunta «¿Considera que el curso se complementa bien con el resto de cursos del programa? ¿Con cuáles? ¿Por qué?» en referencia al curso de Diseño de videojuegos:*

- *«Ninguno.»*
- *«Sí, se complementa bien con el curso de Aplicaciones y programación de videojuegos, también con el curso de Escritura Creativa, y con el curso de Arte y Animación Digital.»*
- *«Sí, se complementa con escritura creativa y animación digital ya que son cursos necesarios para poder diseñar un videojuego.»*
- *«Sí, principalmente con programación, pues permite empezar a programar, ya teniendo ideas más definidas.»*

Pregunta abierta 6.21. *Respuestas a pregunta «¿Tiene alguna sugerencia para mejorar el curso? (Sugerir retirar el curso del programa también es válido)» en referencia al curso de Diseño de videojuegos:*

- *«No.»*

- «Complementar el curso con una segunda etapa (otro curso) que continúe con los contenidos vistos en el primero, de modo que exista una oportunidad de abarcar más contenido. Un apartado específico de diseño de niveles me parecería muy bueno.»
- «No, considero que la catedrática hizo un excelente trabajo en la elaboración del programa del curso. Creo que lo mejor que se podría hacer es partir el curso en dos y llevarlo en paralelo con escritura y animación.»
- «Creo que se beneficiaría más de laboratorios cortos en los que se reproduzcan mecánicas de juegos (como Mix and Jam en YouTube). Esto permitiría condensar ciertos temas para poder tenerlos más concretos.»

6.21.1. Retroalimentación para el curso de programación de videojuegos

¿Considera que el curso de Programación de videojuegos le da las herramientas necesarias para construir una carrera en el área que aborda?

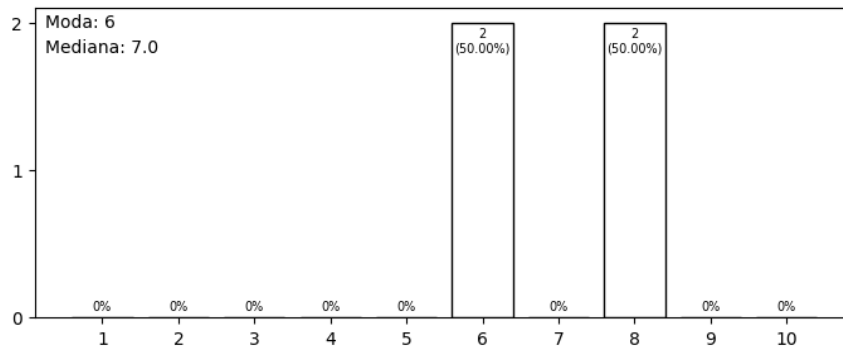


Figura 6.24: Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Programación de videojuegos) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

¿Cree que los contenidos del curso de Programación de videojuegos justifican su inclusión en el programa?

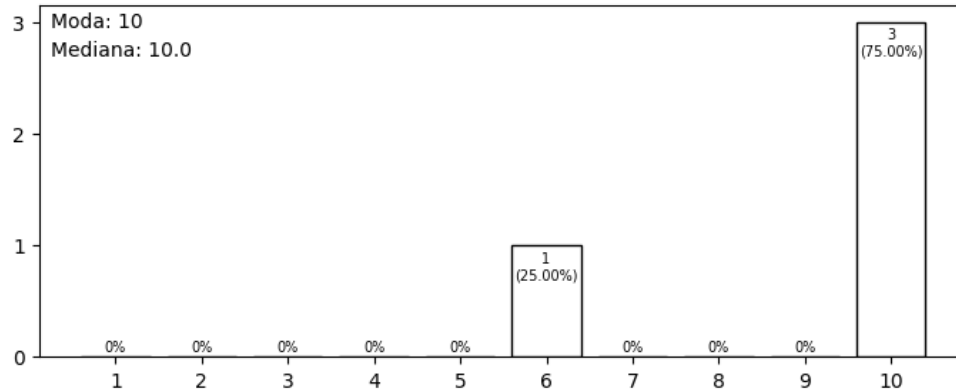


Figura 6.25: Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Programación de videojuegos) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.22. *Respuestas a pregunta «¿Cree que el curso está bien enfocado para cubrir los temas de mayor importancia en el área que aborda? ¿Por qué?» en referencia al curso de Programación de videojuegos:*

- «Si tienen un buen enfoque, pero sufre mucho de tener que empezar desde lo más básico, por lo que se beneficiaría de un curso de 'Programación de videojuegos básica' para que pueda prepararnos de mejor manera»
- «No mucho, varios temas quedan al aire.»
- «Sí, en clase cubrimos los temas necesarios para poder elaborar 2 videojuegos en un semestre. El primero fue una replica de Overcooked y el segundo fue el juego que estuvimos diseñando desde el primer año de la especialización.»
- «Sí, realmente se trata de una base para guiar el desarrollo de futuros proyectos, se cubren las áreas más importantes de la programación de un videojuego durante el desarrollo del proyecto final del curso.»

Pregunta abierta 6.23. *Respuestas a pregunta «¿Considera que el curso se complementa bien con el resto de cursos del programa? ¿Con cuáles? ¿Por qué?» en referencia al curso de Programación de videojuegos:*

-
- *«Se complementa con todos, pues es lo que concreta la unión de los diferentes ámbitos.»*
 - *«Sí, Análisis y diseño de algoritmos»*
 - *«Sí, se complementa con todos los cursos que llevamos previamente; diseño de videojuegos, escritura creativa y animación digital. En esos cursos se diseñó un videojuego en grupo y en programación de videojuegos hicimos un prototipo más completo.»*
 - *«Sí, se complementa bien con el curso de diseño de videojuegos, con el curso de arte y animación digital, y especialmente con el curso de aplicaciones y videojuegos.»*

Pregunta abierta 6.24. *Respuestas a pregunta «¿Tiene alguna sugerencia para mejorar el curso? (Sugerir retirar el curso del programa también es válido)» en referencia al curso de Programación de videojuegos:*

- *«Tener uno previo de progra básica de videojuegos (que dure el semestre entero).»*
- *«Hacer los cursos acumulativos, ver más que lleva la creación del juego, buenas prácticas, estructura del juego.»*
- *«Sugiero que haya una segunda parte del curso. Aun queda mucho por aprender y 6 meses no es suficiente para cubrir todo lo interesante.»*
- *«Este curso también podría tener una continuación ya que existen muchas áreas que no se pudieron explorar por falta de tiempo, pero en general el curso está muy bien planificado, según mi opinión.»*

6.24.1. Retroalimentación para el curso de arte y animación digital

¿Considera que el curso de Arte y animación digital le da las herramientas necesarias para construir una carrera en el área que aborda?

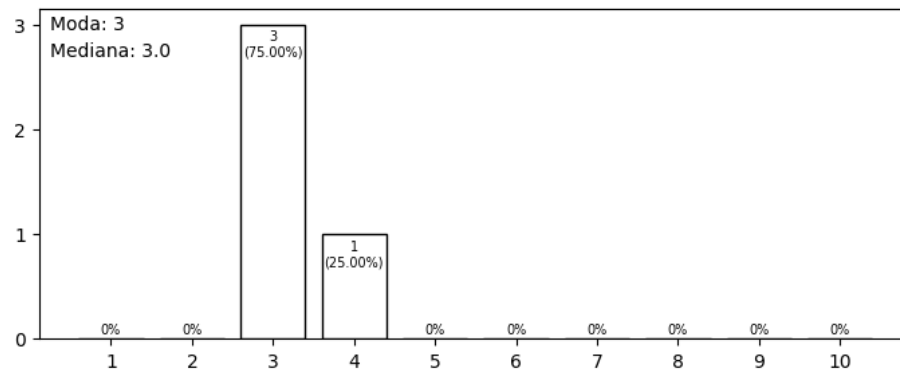


Figura 6.26: Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Arte y animación digital) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

¿Cree que los contenidos del curso de Arte y animación digital justifican su inclusión en el programa?

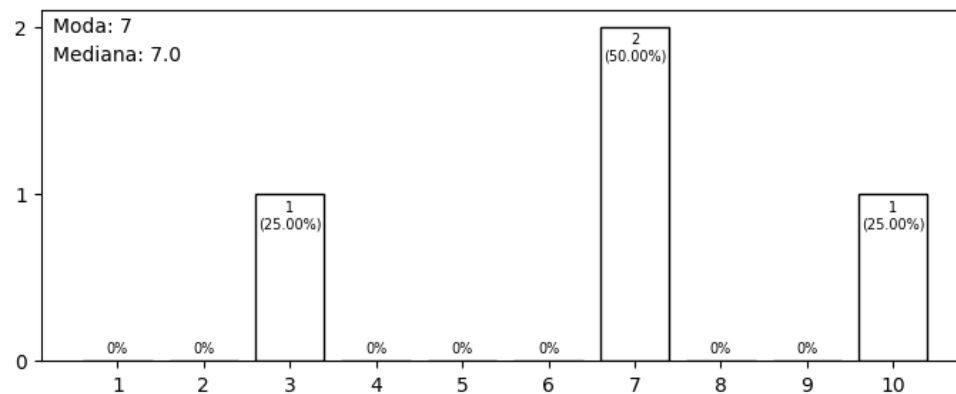


Figura 6.27: Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Arte y animación digital) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.25. *Respuestas a pregunta «¿Cree que el curso está bien enfocado para cubrir los temas de mayor importancia en el área que aborda? ¿Por qué?» en referencia al curso de Arte y animación digital:*

-
- *«Creo que el curso le falta enfoque, pues tiene que cubrir muchos temas. Entonces, faltan maneras en las que se tenga un workflow definido para la producción de assets.»*
 - *«Este curso dejo mucho que desear.»*
 - *«No, solo nos enfocamos principalmente en el diseño de personajes y en el desarrollo de modelos 3d en blender. No vimos mucho relacionado a animación digital.»*
 - *«Considero que el curso no cubre los temas de mayor importancia de este sector de la industria, ya que considero que faltó mucha teoría y conceptos para comprender de una mejor manera el procedimiento que se realiza. Por ejemplo, conceptos de perspectiva, del cuerpo humano, conceptos de dibujo y arte en general, etc.»*

Pregunta abierta 6.26. *Respuestas a pregunta «¿Considera que el curso se complementa bien con el resto de cursos del programa? ¿Con cuáles? ¿Por qué?» en referencia al curso de Arte y animación digital:*

- *«Creo que con el que más se complementa es diseño, pues es una manera de plasmar el área de ambiente, personaje, etc. en algo menos abstracto.»*
- *«No, ninguno. Podría relacionarse con HCI pero no lo hacen.»*
- *«Sí, lo que aprendimos en el curso sirvió para hacer los modelos de los juegos que elaboramos en programación de videojuegos.»*
- *«Se complementa bien con todos los cursos de videojuegos, ya que en muchos de estos como Programación de videojuegos y Diseño de videojuegos es necesario realizar modelos para incluir en los niveles diseñados.»*

Pregunta abierta 6.27. *Respuestas a pregunta «¿Tiene alguna sugerencia para mejorar el curso? (Sugerir retirar el curso del programa también es válido)» en referencia al curso de Arte y animación digital:*

- *«Creo que el curso no se da [abasto] Diseño de props en general estuvo bien Diseño de ambiente no se logró afrontar mucho Modelación de personajes es un tema en el que no se pudo profundizar Rigging y*

animación fue muy poco por falta de tiempo Texturizado se vio algo muy básico, y faltaron tocar temas como shaders, implementación y realización de mapas normales y de transparencia, etc.»

- «Un mejor catedrático»
- «Enfocarse más en animación digital.»
- «Considero que es necesario llevar un curso introductorio antes de llevar este curso, donde se enseñen los conceptos teóricos necesarios para entender las bases de arte, dibujo, perspectiva, modelado, etc. Pero que no solamente cuente con teoría sino que se pueda aplicar en la práctica sin ir demasiado rápido. Por ejemplo yo llevé como selectivo el curso de Dibujo el semestre pasado, y considero que me hubiera servido mucho llevarlo antes de este curso.»

6.27.1. Retroalimentación para el curso de escritura creativa

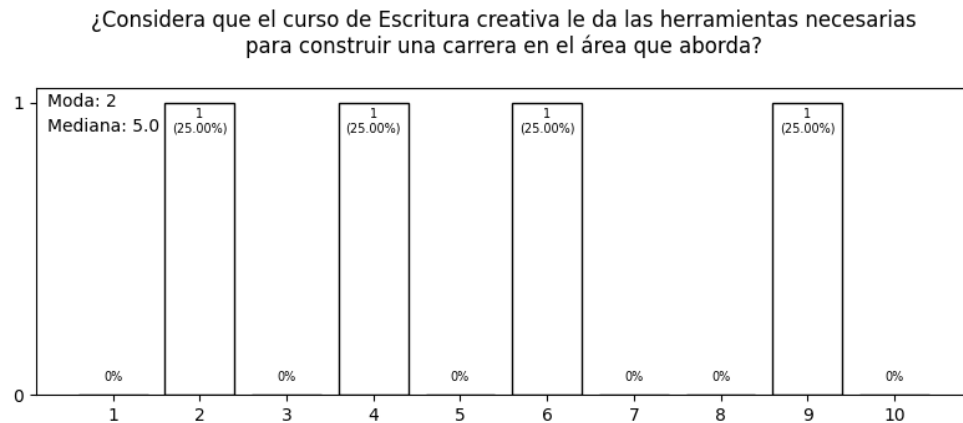


Figura 6.28: Gráfico de la pregunta «¿Considera que el curso le da las herramientas necesarias para construir una carrera en el área que aborda?» (Referente al curso de Escritura creativa) que fue hecha en forma de escala Likert de 1 (No, solo es introductorio) a 10 (Sí, es muy completo). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

¿Cree que los contenidos del curso de Escritura creativa justifican su inclusión en el programa?

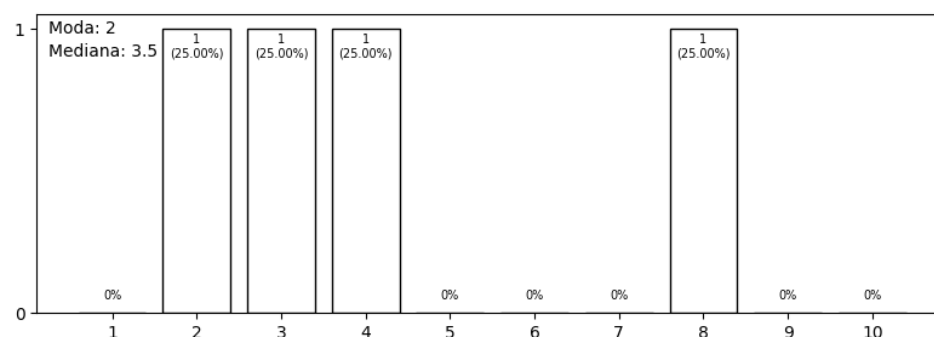


Figura 6.29: Gráfico de la pregunta «¿Cree que los contenidos del curso justifican su inclusión en el programa?» (Referente al curso de Escritura creativa) que fue hecha en forma de escala Likert de 1 (No) a 10 (Sí). Se incluye el total de respuestas para cada valor junto con su porcentaje dentro del total de respuestas, la moda y la mediana, mostradas en la esquina superior izquierda.

Pregunta abierta 6.28. *Respuestas a pregunta «¿Cree que el curso está bien enfocado para cubrir los temas de mayor importancia en el área que aborda? ¿Por qué?» en referencia al curso de Escritura creativa:*

- «Creo que es un curso que, al no ser de computación, no está enfocado a como escribir para un videojuego.»
- «No.»
- «No me acuerdo muy bien, pero creo que si abordamos los temas básicos para escribir personajes, mundos, etc...»
- «Considero que para un semestre, los contenidos del curso abarcan una gran parte del área abordada, aunque no es muy enfocado a videojuegos realmente.»

Pregunta abierta 6.29. *Respuestas a pregunta «¿Considera que el curso se complementa bien con el resto de cursos del programa? ¿Con cuáles? ¿Por qué?» en referencia al curso de Escritura creativa:*

- «Creo que el no ser de computación, hace que no se complemente.»
- «No.»
- «No realmente, creo que lo más importante de este curso lo cubrimos en diseño de videojuegos y animación digital.»
- «Considero que este es un curso que se complementa bien con el curso de Diseño de videojuegos, principalmente por la creación de

una historia durante el curso. Sin embargo, los temas que abarca este curso son un área bastante independiente del diseño de videojuegos.»

Pregunta abierta 6.30. *Respuestas a pregunta «¿Tiene alguna sugerencia para mejorar el curso? (Sugerir retirar el curso del programa también es válido)» en referencia al curso de Escritura creativa:*

- *«Darle un enfoque hacia los videojuegos. Definir como cada 'nivel' tiene storybeats, emotional beats, sin necesidad de escribir en prosa.»*
- *«Removerlo.»*
- *«Retirarlo.»*
- *«Me gustaría que se agregara un enfoque específico para videojuegos, considero que los estudiantes se emocionarían más con este cambio.»*

Durante una discusión que se tuvo con Dennis Aldana, catedrático que coordina e imparte los cursos *Game Engine Architecture* y *Gráficas por computadora*, se recomendó incluir los temas de *Creación y manejo de Assets* y *Creación de efectos de partículas*.

Síntesis de lo discutido:

La justificación dada para incluir el tema de *Creación y manejo de Assets* fue el que el curso de *Arte técnico* tendría el espacio y el enfoque necesario para recontextualizar los conocimientos adquiridos en el curso de *Arte y animación digital*, para que los estudiantes sepan cómo integrar el arte que producen en los juegos que hacen con el mayor contexto posible sobre las mejores prácticas y las herramientas que una *game engine* como *Unity* provee para llevar a cabo esa integración.

La inclusión del tema de *creación de efectos de partículas* fue el hecho de que no es cubierto en ningún curso y el hecho de que un curso de *Arte técnico* tiene la temática perfecta para incluirlo, puesto que es un tema muy alineado con los temas discutidos en este documento y es algo que un artista técnico hace.

En la misma discusión con Dennis Aldana, se mencionó la reciente controversia que hubo alrededor de *Unity* y el agresivo cambio propuesto a la estructura de costos del producto [16]. Esto hizo que muchos usuarios individuales y estudios perdieran la fe en el producto, pues usarlo suponía que perderían dinero en sus proyectos futuros, llevando a muchos

a jurar jamás usar la herramienta. Considerando esto y el hecho de que Unity tiene varios defectos técnicos relacionados a su edad y arquitectura, Dennis planteó la posibilidad de que el Departamento de computación de la universidad buscara una *game engine* más moderna con la cual impartir muchos temas. Se mencionó que Godot puede ser una buena opción para el cambio, puesto que es completamente gratuita y se trata de un proyecto de código abierto desarrollado comunitariamente, por lo que su código fuente está disponible en línea bajo la licencia del MIT.

La retroalimentación para este trabajo fue obtenida de seis estudiantes de quinto año de la especialización de videojuegos y dos catedráticos que imparten cursos específicos a la especialización, Alhvi Balcarcel y Dennis Aldana. Por lo que, si bien seis estudiantes es más de la mitad de la promoción de la especialización de videojuegos, al analizar las figuras presentadas se debe tomar en cuenta que la muestra no es estadísticamente significativa, por lo que los resultados basados en la retroalimentación deben ser interpretados como indicaciones y no como evidencia autoritativa.

7.1. Creación de *shaders*

Para el análisis de la retroalimentación obtenida para este tema (ver 6.4.1), es importante notar que esta fue recolectada luego de una cátedra de ciento veinte minutos y tomó dos semanas para que todos los participantes respondieran la encuesta. Para tener más detalle sobre el material producido y su respectiva, vea A.3.

Originalmente, no se consideró necesaria la inclusión de este tema, puesto que se lo cubre superficialmente en cursos como Gráficas por computadora y Programación de videojuegos. Sin embargo, cuándo se impartió la clase del tema de efectos de *post-processing* la retroalimentación más común fue que fue complicado entender el tema porque no se tenía una buena base en cómo crear *shaders*. Considerando que esto es un factor que puede reducir el nivel de aprendizaje de los estudiantes [25],

se decidió incluir este tema para reforzar las introducciones dadas en los demás cursos, dándole el espacio de un módulo entero del curso para poder abordar un tema tan amplio. Considerando lo anterior, es de esperar que el tema sea bien recibido por los estudiantes, quienes expresando que les sería muy útil aprenderlo, como se evidencia en 6.11.

En cuanto a la calidad del material didáctico producido para el tema de creación de *shaders*, por lo general la retroalimentación es positiva. Esto se puede ver en las figuras 6.8, 6.9, donde los estudiantes expresaron que les pareció que la información fue bien presentada y que, por lo general, pudieron entender los temas.

Lo discutido anteriormente también se apoya en las respuestas a la pregunta «¿Qué cambiaría de la clase? ¿Por qué?» (ver 6.4.1), donde tres estudiantes no tuvieron observaciones, ya sea no respondiendo o comentando que no harían cambios. Sin embargo, las tres sugerencias dadas resaltan la mayor falla que tuvo la presentación, su longitud y densidad. Esto es claro en los comentarios dados, los cuales sugieren incluir más ejemplos y que estos se den paso a paso, y reducir el tiempo de la lección. Esto concuerda con las observaciones dadas por Dennis Aldana (ver 6.4.1), quien recomendó incluir más ejemplos y comentó que el contenido es más adecuado para un mes entero de clases; y los comentarios de Alhvi Balcarcel (ver 6.4.1), quien también sugirió la inclusión de ejercicios y resaltó la importancia de estos para la retención del conocimiento.

Las respuestas a la pregunta «¿Qué tan importante considera la inclusión de ejemplos para la retención de los temas» (ver 6.10) y la pregunta «¿Cree que es más útil aprender los temas a base de ejemplos de casos de uso?» (ver 6.4.1) también apoyan al caso de que la falla de la presentación producida para el tema de Creación de *shaders* fue cubrir demasiado contenido, ya que se necesita más tiempo para explicar los ejemplos. En ambos casos, los estudiantes resaltaron la importancia de acompañar el contenido con ejemplos, lo que concuerda con el constructivismo, pues tener más recursos les facilitará desarrollar aprendizajes. [26]

Finalmente, la narrativa de que el contenido fue demasiado denso es soportada por las respuestas a la pregunta: «¿Cuál cree que es el alcance apropiado para los temas impartidos?» (ver 6.4.1), en la cual la gran mayoría indicó que considera que los temas deben ser cubiertos en un módulo entero. Aunque por la formulación de la pregunta, esto también

indica que los estudiantes creen que el tema es lo suficientemente importante como para que tenga una prevalencia tan grande dentro de un curso.

Un punto importante que ha quedado en el tintero es el hecho de que, más allá de ejemplos, los estudiantes necesitan practicar para retener el conocimiento impartido. Esto lo soporta [24] y los comentarios de Alhvi Balcarcel. Nuevamente, se resalta la importancia de partir los temas cubiertos en diferentes lecciones que los vean a mayor detalle, con más ejemplos, tiempo para absorberlos y practicar lo aprendido.

Por último, el que las respuestas a la pregunta: «¿Cree que los conocimientos impartidos en los cursos de la especialización de videojuegos y en la carrera en general le dan una buena base para aprender sobre la creación y configuración de *shaders*?» no hayan sido positivas, indican a la posible existencia de un problema, el que los estudiantes de quinto año de la especialización no sean capaces de aprender sobre un tema tan prevalente en la industria como lo es la creación de *shaders*. En este caso, es importante considerar que la muestra fue de solamente seis estudiantes, por lo que sin relevancia estadística es imposible confirmar esta noción. Las respuestas a esta pregunta deben ser interpretadas meramente como una indicación para plantear la hipótesis «Los estudiantes de quinto año de la especialización de videojuegos de la carrera de Ingeniería de ciencias de la computación y tecnologías de la información de la universidad del Valle de Guatemala no cuentan con los conocimientos para poder desarrollar la capacidad de crear *shaders* para videojuegos a un nivel profesional».

7.2. Efectos de *post-processing*

Para el análisis de la retroalimentación obtenida para este tema (ver 6.10.1), es importante notar que esta fue recolectada luego de una cátedra de cuarenta minutos y tomó dos semanas para que todos los participantes respondieran la encuesta. Para tener más detalle sobre el material producido y su respectiva, vea A.2.

Las respuestas a la pregunta: «¿Cómo califica la calidad de la clase?» (ver 6.13) apuntan a que el material didáctico producido para el tema de Efectos de *post-processing* tuvo una buena calidad dado que los es-

tudiantes lo reportaron directamente. La noción también fue soportada por los comentarios de los catedráticos Dennis Aldana y Alhvi Balcarcel, quienes expresaron que el material estuvo bien hecho, salvo unas pequeñas observaciones sobre la claridad de la información presentada por las diapositivas 5 y 6 (ver A.2), para las cuales ella requirió una aclaración.

La noción de que el material estuvo bien hecho se contradice con las respuestas a la pregunta «¿Qué tanto entendió el tema de creación de efectos de *post-processing*?» (ver 6.14), pues la mayoría de estudiantes respondieron un 6 de 10 en la escala de Likert. Por lo que se puede interpretar esto como que los estudiantes reportaron no haber entendido gran parte de los temas expuestos. Además, las respuestas a la pregunta «¿Se considera capaz de realizar el ejercicio presentado al final de la clase?» (ver 6.15) también indican que hubo un pobre entendimiento por parte de los estudiantes, ya que muchos se reportaron incapaces de hacer el ejercicio propuesto y los estudiantes más entusiastas hacia el prospecto no demostraron mucha confianza, siendo la respuesta más alta 7 de 10 y la más baja 3 de 10. Sin embargo, es importante considerar que los estudiantes no tuvieron el tiempo de intentar resolver el ejercicio antes de responder.

Las respuestas a la pregunta: «¿Con qué temas le queda duda?» (ver 6.10.1) pueden apoyar a entender la razón de por qué los estudiantes no se mostraron entusiastas sobre su entendimiento de los temas ni su capacidad de resolver el ejercicio propuesto. Los estudiantes mencionaron que les quedó duda con temas como *shaders*, *rendering pipelines* y matemáticas. Todos son temas que se esperaba que los estudiantes conocieran previamente, puesto que se cubren en Gráficas por computadora, Álgebra lineal I y Ecuaciones diferenciales I, siendo todos cursos que fueron aprobados por los estudiantes encuestados. Posiblemente el error fue asumir que los estudiantes pudieran abordar el tema sin primero tener un recordatorio de los temas que sirven como base. Se espera que este problema se agrave una vez todos los cursos de la especialización se impartan en quinto año, pues muchos de los temas serán requeridos por los estudiantes luego de varios años de no haberlos practicado directamente.

Al examinar las respuestas a las preguntas «¿Considera que los temas presentados le serían útiles a los futuros estudiantes de la especialización?» (ver 6.16) y «Justifique su respuesta anterior (Referente a la pre-

gunta «¿Considera que los temas presentados le serían útiles a los futuros estudiantes de la especialización.» (ver 6.10.1), se encontró evidencia que apunta a que vale la pena incluir el tema de creación de efectos de *post-processing* en un curso de la especialización de videojuegos. Cuando se les pidió a los estudiantes elaborar sobre el por qué de su opinión, la mayoría expresó que piensa que es una habilidad que les permitiría añadir mucho valor a sus videojuegos y que cubrirla en un curso universitario sería más valioso que simplemente referirse a los recursos disponibles en línea. Sin embargo, hubo una opinión que difirió, expresando que se trata de una habilidad muy específica y que sería mejor dejar que solo los estudiantes más interesados en la rama de gráficas por computadora la aprendan, lo cual es una opinión válida y que se debe considerar, puesto que el objetivo es aumentar el valor que la especialización provee a todos los estudiantes. Del mismo modo que las opiniones mostradas en la Figura 6.16 fueron positivas.

Al tratar con el valor provisto por la inclusión del tema de Creación de efectos de *post-processing* también fue importante el considerar los comentarios de los catedráticos, quienes consideraron que vale la pena incluir el tema, pero que puede ser complicado lograr incluirlo. Dennis Aldana expresó que la inclusión del tema sería sencillo tomando en cuenta que sería un tema corto de impartir, pero que este tiene el problema de requerir una mejor base en los temas de *shaders* y *rendering pipelines*, añadiendo que es posible dar un mayor enfoque a los temas en el curso de gráficas por computadora. Mientras que Alhvi Balcarcel expresó que valdría la pena incluir el tema en uno de los cursos actuales, pero que sería difícil conseguir espacio para un curso nuevo. Debido a que las muestras, tanto de estudiantes como de catedráticos, son estadísticamente insignificantes, se requerirá investigar más para determinar si es posible y valioso incluir el tema, ya sea en un curso nuevo o modificando uno de los cursos actuales.

Dennis Aldana mencionó que uno de los problemas que puede traer el incluir este tema es que puede que los estudiantes necesiten una mejor base en los temas de Creación de *shaders* y contexto sobre *rendering pipelines*. Si bien esto no está confirmado, las opiniones expresadas por los estudiantes en las respuestas a la pregunta «Como estudiante de la especialización de videojuegos de 5to año, ¿Su conocimiento previo le fue

suficiente para entender los temas?» (ver 6.10.1) indican que es una noción correcta, puesto que los estudiantes expresaron no haber aprendido o no recordar material que necesitan para entender el tema. Además, las respuestas a la pregunta «¿Qué temas incluidos en otros cursos de la especialización (si es que alguno), considera necesarios para comprender lo presentado en clase?» (ver 6.10.1) contradicen la noción, pues los estudiantes identificaron los mismos temas que identificó Dennis Aldana. Nuevamente, se requiere hacer una investigación más grande para obtener resultados conclusivos.

7.3. Efectos visuales avanzados haciendo uso de múltiples herramientas

Es importante tomar en cuenta que la retroalimentación para esta sección fue tomada luego de que los estudiantes y catedráticos vieran un video explicativo corto sobre el tema que demostró un caso de uso y explicó a groso modo cómo reproducirlo. El video fue creado y publicado en YouTube por Martin Donald [6] y tiene una duración de diez minutos. Esto difiere de la metodología utilizada para los otros dos temas, para los cuales se produjo material y se le dio una cátedra a los estudiantes y dado que no tuvieron material didáctico sobre el cual dar comentarios, la retroalimentación de los catedráticos fue recolectada junto con la de los estudiantes. Si bien hubo una variación en la metodología que reemplazó una cátedra con un video de 10 minutos, tanto catedráticos como estudiantes reportaron tener un buen entendimiento del tema luego de ver el video, como se evidencia en la Figura 6.17.

Las figuras 6.18 y 6.19 apuntan a que el tema entra dentro de la temática del curso propuesto y que el aprenderlo aportaría valor a los estudiantes de la especialización. Notablemente, este es el tema con mayor aprobación de todos, puesto que todos los estudiantes y catedráticos respondieron un 10 de 10, cosa que no ocurrió para los otros dos temas. A pesar de esto, la evidencia presentada sigue siendo insuficiente para determinar definitivamente las nociones planteadas por las preguntas.

Las opiniones de los estudiantes y catedráticos difieren en las respuestas a la pregunta: «¿Cuánta importancia deberían tener estos temas dentro de un curso de Arte técnico?» (ver 6.16.1), donde la gran mayoría de los

estudiantes consideraron más adecuado incluir este tema dentro de un curso como parte de un módulo, mientras que los catedráticos opinaron que el alcance es más adecuado para un módulo entero. Debido a que las muestras son demasiado pequeñas, no es posible dar una respuesta definitiva basándose en las opiniones recolectadas, aunque las opiniones de los catedráticos deben considerarse más que la de los estudiantes.

Las respuestas mostradas en las figuras 6.20 y 6.21 son importantes para poder determinar la viabilidad de incluir el tema en un curso de la especialización. Ambas indican que la inclusión del tema sería difícil de lograr. Primero, la opinión de los estudiantes y catedráticos tiende a que el tema sería demasiado complicado de cubrir sin que el conocimiento sea altamente dependiente de las herramientas usadas, lo que no es adecuado para enseñar el tema como tal, sino solo para enseñar cómo reproducir efectos similares con las herramientas utilizadas. En sí, indica que el tema sería demasiado específico para incluirlo en un curso universitario. En cuanto a la segunda pregunta, si bien las opiniones difieren mucho, la mediana y la moda están alrededor de 6 y 7 de 10, lo que indica que por lo general la opinión no demuestra confianza en que los estudiantes sean capaces de aprender el tema en base a los conocimientos que adquieren en los cursos de la especialización. Este resultado no es definitivo, pero es otra indicación que apunta a que el tema sería demasiado difícil de incluir y que se ver soportada por [25], pues se sabe que los estudiantes necesitan el conocimiento previo que no indicaron tener.

7.4. Retroalimentación a cursos actuales de la especialización de videojuegos

Como se puede ver en las secciones 6.18.1 y 6.21.1, ambos cursos fueron tienen un buen nivel de aceptación por los estudiantes encuestados, quienes indicaron que los cursos justificaban su inclusión en el p \acute{e} nsum y que estos los prepararon bien para desempeñarse en las áreas que cubren. Aunque, nuevamente, se debe tomar en cuenta que los resultados no son estadísticamente significativos y consecuentemente son inconcluyentes.

Un punto importante que se repite en la retroalimentación dada para ambos cursos fue el que el curso de Programación de plataformas móviles y juegos, retirado en la última revisión que se le dio al p \acute{e} nsum, fue re-

ferenciado repetidamente. Algunos comentarios resaltan que el curso de Diseño de videojuegos dependió mucho de ese curso para poder cubrir su tema, esto tiene sentido, puesto que para poder diseñar los estudiantes necesitan poder prototipar sus diseños. De manera similar, el curso de Programación de videojuegos se benefició de tener un público familiarizado con las herramientas utilizadas. Si bien se trata de observaciones hechas por pocos estudiantes, el que los estudiantes futuros de la especialización no hayan tomado el curso de Programación de plataformas móviles y juegos es algo que se debe considerar a la hora de impartir y actualizar los cursos.

Como se puede ver en la Figura 6.26, según los estudiantes, el curso de Arte y animación digital no logró prepararlos para desarrollarse como artistas digitales. Como se puede ver en las respuestas a la pregunta 6.24.1, los estudiantes identificaron que el curso necesita más tiempo para cubrir sus temas a completitud y de satisfactoria, mientras que en las respuestas a la pregunta 6.24.1, expresan frustración por no haber aprendido sobre varios temas, incluidas las bases de conocimiento técnico sobre arte más allá del uso de herramientas como Blender. Según [25], tanto el no contar con esa teoría previa, como el cubrir demasiados temas en un solo curso, pueden dificultar el aprendizaje de los estudiantes, por lo que el departamento de computación debe considerar la introducción de un curso de arte con el objetivo de darle a los estudiantes una mejor base artística y cubrir los temas de la disciplina sin sobrecargar a los estudiantes.

A pesar de la retroalimentación negativa, la mayoría de estudiantes expresaron que el curso justifica su inclusión en el p \acute{e} nsum de la especialización (ver Figura 6.27) y que se complementa bien con los demás cursos de la especialización (ver 6.24.1). Además de que, por lo general, consideran que el curso complementa bien a los demás cursos del p \acute{e} nsum. (ver 6.24.1) Lo que argumenta positivamente al caso para que el departamento busque apoyar al curso, en vez de cambiarlo por completo.

Como se puede ver en la sección 6.27.1, el curso de Escritura creativa fue retroalimentado divisivamente, pero en su mayoría con poco entusiasmo por parte de los estudiantes, siendo el curso con la menor medianas menor a 5 en las dos figuras correspondientes a si los estudiantes consideran que su inclusión en el p \acute{e} nsum es justificada (ver 6.29) y si el curso les es útil para desarrollarse en la disciplina de escritura creativa (ver 6.28).

Si bien estos resultados no son estadísticamente significativos, se deben considerar junto con la retroalimentación abierta recibida; los estudiantes opinan que, a pesar de que se hicieron esfuerzos por contextualizar los aprendizajes bajo el lente del desarrollo de videojuegos, el curso logró cubrir los fundamentos de la escritura creativa, pero no les enseñó cómo escribir para videojuegos, pues no fue creado con la especialización en mente. Todos los estudiantes sugirieron reenfocar el curso para que este cubra mejor los temas pertinentes a la escritura para videojuegos o sugirieron retirarlo del programa. Si bien los resultados no presentan evidencia que permite concluir que el curso requiere cambios, el departamento, con el objetivo de asegurar que los conocimientos impartidos sean útiles para el desarrollo de videojuegos, debe evaluar la posibilidad de crear un curso nuevo o introducir temas de narrativa en el curso de diseño de videojuegos, pues es el curso de la especialización más afín a incluir los temas, puesto que la narrativa puede ser sumamente importante para el diseño de un videojuego[22].

Durante la discusión que se tuvo con Dennis Aldana, se mencionó el tema de que quizá pueda ser beneficioso para la especialización que se intente hacer el cambio de Unity a otra *game engine* como Godot. Godot sería una buena opción dado que es completamente gratuita y, gracias a que utiliza la licencia MIT, el uso, modificación y redistribución de las versiones disponibles actualmente seguirá siendo gratuita a perpetuidad.[27] Además de que el código fuente está disponible bajo la misma licencia, lo que permitiría a un estudiante estudiar y modificar el código fuente en dado caso sea necesario. En cuanto a la utilidad de Godot en comparación con Unity, esta última no tiene muchas ventajas competitivas fuera de aplicaciones como la realidad virtual y la realidad aumentada.[42] Además, Godot ha estado cobrando relevancia en los últimos años, aunque su popularidad aún está lejos de equipararse a Unity.[4]

A pesar de que el material didáctico producido es de alta calidad, de ser utilizado para impartir clases, el material debe ser editado para que se cubra en más lecciones y enriquecido con más ejemplos prácticos y ejercicios, con la meta de diluir el contenido para darle más tiempo a los estudiantes de practicar y absorber lo aprendido.

Se compilaron las opiniones de estudiantes de la especialización de videojuegos y catedráticos que han diseñado e impartido cursos específicos a la especialización que apuntan a que vale la pena introducir los temas de Creación de *shaders* y Creación de efectos *post-processing*, aunque debido a que la muestra es demasiado pequeña, no es posible afirmar esa noción.

Si bien la evidencia no es conclusiva, esta indica que la inclusión del tema de Efectos visuales avanzados haciendo uso de múltiples herramientas sería muy difícil de lograr en un curso de la especialización de videojuegos y que el valor que aportará a los estudiantes sería demasiado específico, hasta el punto de ser poco aplicable. Razón por la cual no se recomendó que el departamento de computación evalúe su inclusión.

Con base en los comentarios de estudiantes y catedráticos recopilados en este trabajo, se recomienda al departamento de computación investigar con mayor profundidad para poder confirmar las nociones presentadas en este documento, con el objetivo de reevaluar la inclusión del curso de Escritura creativa en el p nsu m de la especializaci n, puesto que las opiniones reportadas apuntan a que el curso no est a sirviendo el prop sito para el cu al fue introducido y que existen otros temas de mayor importancia que no son cubiertos por ning n curso.

Se recomienda al departamento de computaci n investigar a fondo sobre el valor que el aprender sobre los temas de Creaci n de *shaders*, Creaci n y manejo de *assets* y Creaci n de efectos de *post-processing* aportar a a los estudiantes de la especializaci n de videojuegos, y si es posible para los estudiantes aprender los temas en base a lo que se cubre actualmente en los cursos de la especializaci n. Con el objetivo de evaluar la inclusi n de los temas en los cursos actuales o la introducci n de un curso nuevo que cubra los temas.

Ahora que la especializaci n ser a cubierta solamente en quinto a no, se recomienda que el departamento de computaci n investigue sobre si vale la pena incluir material que sirva como recordatorio de temas base para todos los cursos de la especializaci n, puesto que puede ser dif cil para los estudiantes recordar los temas base para los cursos luego de varios a nos de no haberlos utilizado ni visto en ning n curso. Que ser a el caso para muchos temas de matem tica y lo cubierto en Gr ficas por computadora.

Con base en la que importancia que según los estudiantes tuvo el curso de Programación de plataformas móviles y juegos y el hecho de que este fue retirado del p nsum hace un a o, se recomienda que el departamento de computaci n investigue sobre si los cursos de la especializaci n necesitan ser adaptados para acomodar a los estudiantes de la promoci n 2025 en adelante, pues no contar n con la base provista por el curso en el uso de Unity; y seg n los estudiantes, es algo que se debe considerar para los cursos de Dise o de videojuegos y Programaci n de videojuegos.

Considerando el golpe que sufri  la reputaci n de Unity a los ojos de la industria de producci n de videojuegos, se recomienda al departamento de computaci n seguir de cerca el desarrollo de la situaci n para poder adaptarse a la posible eventualidad de que Unity caiga en desuso, haciendo que la especializaci n pase a utilizar otra *game engine* con capacidades equiparables, como Godot.

Seg n la narrativa expresada por los estudiantes sobre el curso de Arte y animaci n digital se recomienda al departamento de computaci n reenfocar el curso o introducir un curso complementario, con el objetivo de asegurar que los temas presentados en el curso se cubran satisfactoriamente, puesto que, seg n la opini n los estudiantes encuestados, el curso se siente muy cargado y debido a la falta de tiempo, temas como la animaci n de personajes, la creaci n de texturas y mapas de normales, y las bases del dibujo y dise o de personajes, fueron impartidos burdamente u omitidos del todo.

- [1] renderizar | FundeuRAE. *Shading Techniques - How to Shade with a Pencil*. URL: <https://www.fundeu.es/consulta/renderizar-3/>. (accessed: 3.12.2023).
- [2] Agate. *A Closet Look at Technical Art in the Gaming Industry*. URL: <https://agate.id/a-closer-look-at-technical-art-in-the-gaming-industry/>. (accessed: 30.11.2023).
- [3] Ned Atton. *Toon Outlines in Unity URP, Shader Graph Using Sobel Edge Detection!* URL: <https://www.youtube.com/watch?v=RMt6DcaMxcE>. (accessed: 14.7.2023).
- [4] itch corp. *Most used Engines - itch.io*. URL: <https://itch.io/game-development/engines/most-projects>. (accessed: 1.11.2023).
- [5] DePaul. *Technical Artist Concentration*. URL: <https://www.cdm.depaul.edu/academics/pages/current/Requirements-MA-In-Animation-Technical-Artist.aspx>. (accessed: 31.10.2023).
- [6] Martin Donald. *Vertex animation textures, beanbags and boneless animations*. URL: <https://www.youtube.com/watch?v=NQ5D11bxbz4>. (accessed: 01.10.2023).
- [7] Matt Fussell. *Shading Techniques - How to Shade with a Pencil*. URL: <https://thevirtualinstructor.com/shading-techniques-basics.html>. (accessed: 3.12.2023).
- [8] Matt Gambell. *Unlocking The Power Of Unity's Scriptable Render Pipeline*. URL: <https://www.youtube.com/watch?v=9fa4uFm1eCE>. (accessed: 14.7.2023).
- [9] Epic Games. *Hair Simulation and Rendering Quick Start Guide in Unreal Engine | Unreal Engine 5.2 Documentation*. URL: <https://docs.unrealengine.com/5.2/en-US/hair-simulation-and-rendering-quick-start-guide-in-unreal-engine/>. (accessed: 2.11.2023).
- [10] Khronos Group. *OpenGL Shading Language*. URL: https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language. (accessed: 30.11.2023).
- [11] Khronos Group. *Rendering Pipeline Overview - OpenGL Wiki*. URL: https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview. (accessed: 14.7.2023).
- [12] Freya Holmér. *Math For Game Devs (2020)*. URL: https://www.youtube.com/playlist?list=PLImQaTpSAdsD88wprTConznD10Y1EfK_V. (accessed: 31.10.2023).
- [13] Freya Holmér. *Shaders For Game Devs*. URL: <https://www.youtube.com/playlist?list=PLImQaTpSAdsCnJon-Eir92SZM17tPBS4Z>. (accessed: 8.9.2023).
- [14] Freya Holmér. *The Continuity of Splines*. URL: <https://www.youtube.com/watch?v=jvPPXbo87ds>. (accessed: 15.3.2024).
- [15] et al. John Owens. «GPU Computing». En: *Proceedings of the IEEE* 96.5 (2008). DOI: <http://dx.doi.org/10.1109/JPROC.2008.917757>.

-
- [16] Mike Isaac Kelen Browning. *How a Pricing Change Led to a Revolt by Unity's Video Game Developers*. URL: <https://www.nytimes.com/2023/10/02/technology/how-a-pricing-change-led-to-a-revolt-by-unitys-video-game-developers.html>. (accessed: 29.10.2023).
- [17] LearnOpenGL. *LearnOpenGL - Depth testing*. URL: <https://learnopengl.com/Advanced-OpenGL/Depth-testing>. (accessed: 3.12.2023).
- [18] Microsoft. *High-level shader language (HLSL) - Win32 apps | Microsoft Learn*. URL: <https://learn.microsoft.com/en-us/windows/win32/direct3dhls1/dx-graphics-hls1>. (accessed: 14.7.2023).
- [19] Microsoft. *Intrinsic Functions - Win32 apps | Microsoft Learn*. URL: <https://learn.microsoft.com/en-us/windows/win32/direct3dhls1/dx-graphics-hls1-intrinsic-functions>. (accessed: 2.12.2023).
- [20] Peter Pacheco. *An Introduction to Parallel Programming*. Elsevier Inc, 2011. ISBN: 978-0-12-374260-5.
- [21] Jen Lowe Patricio Gonzalez. *The Book of Shaders: uniforms*. URL: <https://thebookofshaders.com/03/>. (accessed: 8.9.2023).
- [22] Alena Porokh. *Unveiling the art of narrative game design: a comprehensive guide*. URL: <https://kevrugames.com/blog/unveiling-the-art-of-narrative-game-design-a-comprehensive-guide/>. (accessed: 30.11.2023).
- [23] Santa Monica Studios. *God of War: Ragnarok*. 26 de mar. de 2023. URL: <https://www.playstation.com/es-gt/games/god-of-war-ragnarok/>.
- [24] Abatihun Alehegn Sewagegn y Boitumelo M. Diale. «Empowering Learners Using Active Learning in Higher Education Institutions». En: *Active Learning*. Ed. por Sílvia Manuel Brito. Rijeka: IntechOpen, 2019. Cap. 3. DOI: 10.5772/intechopen.80838. URL: <https://doi.org/10.5772/intechopen.80838>.
- [25] *Information Overload in Technology-based Education: a Meta-Analysis*. 2013.
- [26] George Siemens. *A Learning Theory for the Digital Age*. URL: <https://pressbooks.pub/lidtfoundations/chapter/connectivism-a-learning-theory-for-the-digital-age/>. (accessed: 31.10.2023).
- [27] Yuri Sizov. *godot-website/LICENSE.txt at master · godotengine/godot-website · GitHub*. URL: <https://github.com/godotengine/godot-website/blob/master/LICENSE.txt>. (accessed: 1.11.2023).
- [28] Irwin Sobel. «An Isotropic 3x3 Image Gradient Operator». En: *Presentation at Stanford A.I. Project 1968* (feb. de 2014).
- [29] *Spider-Man: Into the Spider-Verse*. 2018.
- [30] StackOverflow. *opengl - What is the difference between a sample, a pixel and a fragment? - Stack Overflow*. URL: <https://stackoverflow.com/questions/31173002/what-is-the-difference-between-a-sample-a-pixel-and-a-fragment>. (accessed: 8.9.2023).
- [31] Benjamin Swee. *Writing Unity Shaders Using Depth Textures*. URL: <https://www.youtube.com/watch?v=yUVrtPCsCb0>. (accessed: 14.7.2023).
- [32] Unity Technologies. *How Shader Graph can be applied to your 2D or 3D game | Unity*. URL: <https://unity.com/features/shader-graph>. (accessed: 30.11.2023).
- [33] Unity Technologies. *How to create a custom Renderer Feature | Universal RP | 12.1.13*. URL: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.1/manual/containers/create-custom-renderer-feature-1.html>. (accessed: 14.7.2023).
- [34] Unity Technologies. *Perform a full screen blit in URP*. URL: <https://github.com/UnityTechnologies/Graphics/blob/master/Packages/com.unity.render-pipelines.universal/Documentation~/renderer-features/how-to-fullscreen-blit.md>. (accessed: 14.7.2023).

-
- [35] Unity Technologies. *Post-processing in the Universal Render Pipeline | Universal RP | 12.1.13*. URL: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.1/manual/integration-with-post-processing.html>. (accessed: 14.7.2023).
- [36] Unity Technologies. *Unity - Manual: Frame Debugger*. URL: <https://docs.unity3d.com/560/Documentation/Manual/FrameDebugger.html>. (accessed: 3.12.2023).
- [37] Unity Technologies. *Unity - Manual: ShaderLab: commands*. URL: <https://docs.unity3d.com/Manual/shader-shaderlab-commands.html>. (accessed: 8.9.2023).
- [38] Unity Technologies. *Unity - Manual: Shaders*. URL: <https://docs.unity3d.com/Manual/Shaders.html>. (accessed: 8.9.2023).
- [39] Dirección de estudios de la Universidad del Valle de Guatemala. *Manual de redacción de competencias*. Dirección de estudios de la Universidad del Valle de Guatemala. 17 págs. (accessed: 31.10.2023).
- [40] Jason Weimann. *8 game dev jobs to choose from - which is right for you? unity3d?* URL: <https://www.youtube.com/watch?v=v24j8vzROCY>. (accessed: 31.10.2023).
- [41] Alan Wen. *How to get a job as a technical artist*. URL: <https://www.gamesindustry.biz/how-to-get-a-job-as-a-technical-artist>. (accessed: 31.10.2023).
- [42] Zenva. *Unity vs Godot - A Game Engine Analysis*. URL: <https://gamedevacademy.org/unity-vs-godot/>. (accessed: 1.11.2023).

A.1. Presentación de introducción al curso



Figura A.1: Diapositiva 1 de la presentación de introducción al curso

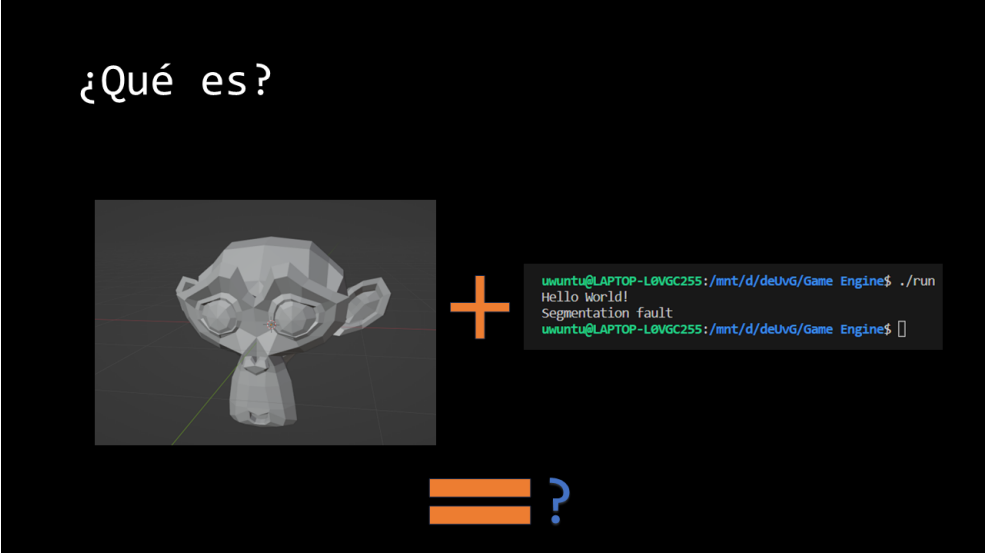


Figura A.2: Diapositiva 2 de la presentación de introducción al curso



Figura A.3: Diapositiva 3 de la presentación de introducción al curso

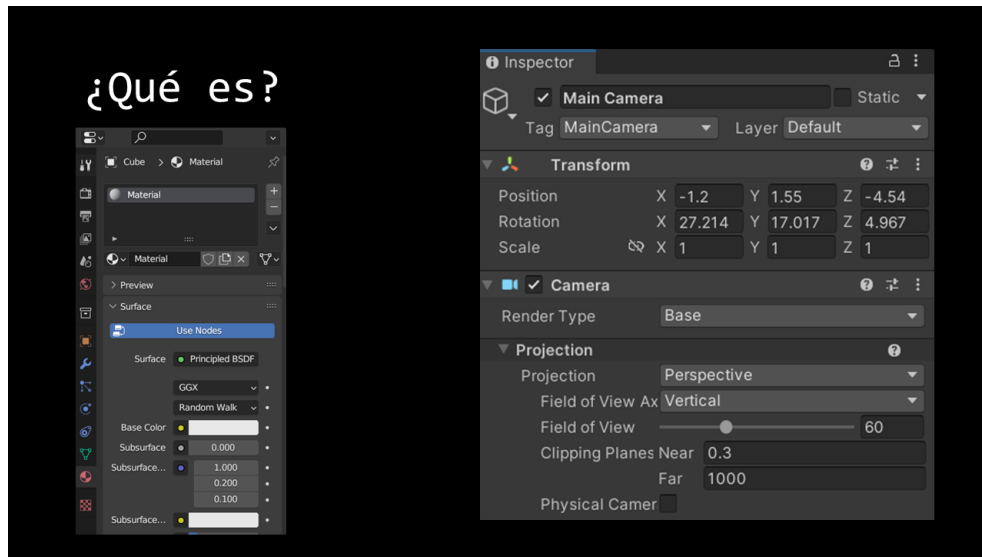


Figura A.4: Diapositiva 4 de la presentación de introducción al curso



Figura A.5: Diapositiva 5 de la presentación de introducción al curso

¿Por qué lo estamos viendo?

El arte técnico permite a los estudios tener:

- Mejor productividad
- Acabados profesionales
- Capacidades únicas

Figura A.6: Diapositiva 6 de la presentación de introducción al curso

¿Quién es el usuario final?

El equipo de desarrollo

Artistas

Programadores

Figura A.7: Diapositiva 7 de la presentación de introducción al curso



Figura A.8: Diapositiva 8 de la presentación de introducción al curso

A.2. Presentación de efectos de *post-processing*

A.2.1. Enlace a repositorio de demostración usado para impartir el tema de creación de efectos de *post-processing*

Enlace al repositorio de demostración usado durante la presentación (Unity 2022.3.3f1):

<https://github.com/JmDeleon2000/SRP-post-processing>

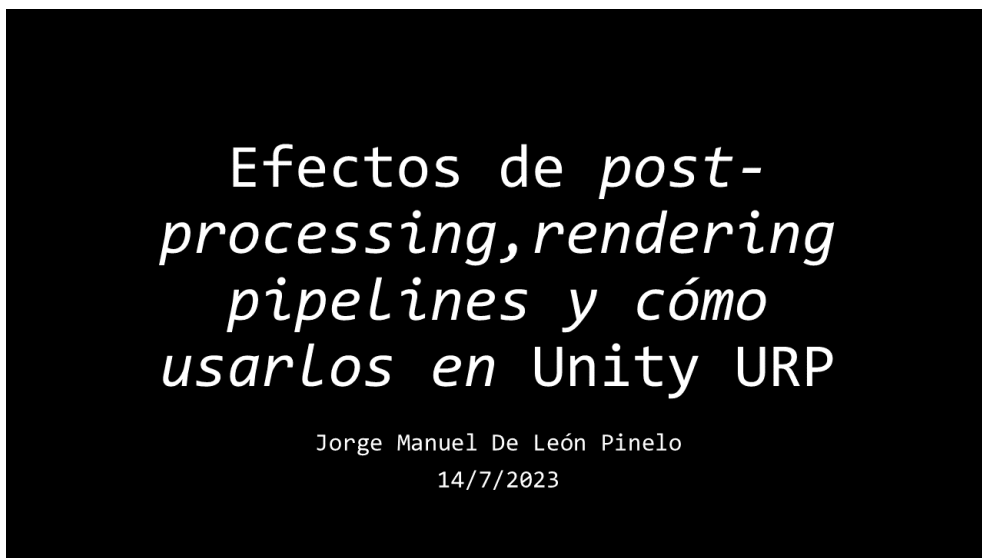


Figura A.9: Diapositiva 1 de la presentación de introducción al curso

¿Qué es post-processing?

- ¿Qué información tenemos disponible tan tarde en el proceso de renderizado? ¿Proceso de renderizado?

Figura A.10: Diapositiva 2 de la presentación de introducción al curso

Rendering pipeline

Fases necesarias para transformar el mundo del juego a píxeles en pantalla.

Tenemos información diferente en cada fase*

*depende de la pipeline que usemos

```

graph TD
    A[Vertex Specification] --> B[Vertex Shader]
    B --> C[Tessellation]
    C --> D[Geometry Shader]
    D --> E[Vertex Post-Processing]
    E --> F[Primitive Assembly]
    F --> G[Rasterization]
    G --> H[Fragment Shader]
    H --> I[Per-Sample Operations]
  
```

Figura A.11: Diapositiva 3 de la presentación de introducción al curso

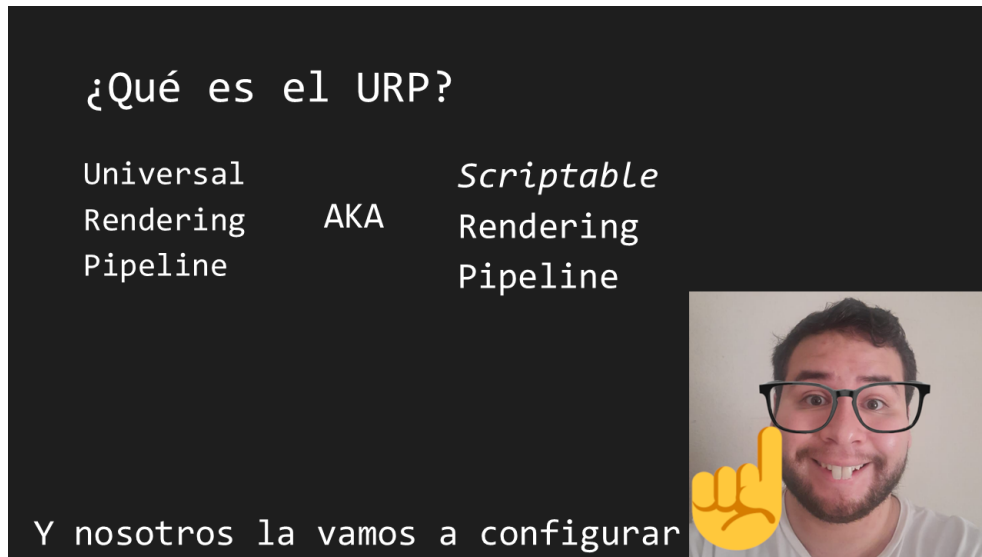


Figura A.12: Diapositiva 4 de la presentación de introducción al curso



Figura A.13: Diapositiva 5 de la presentación de introducción al curso



Figura A.14: Diapositiva 6 de la presentación de introducción al curso

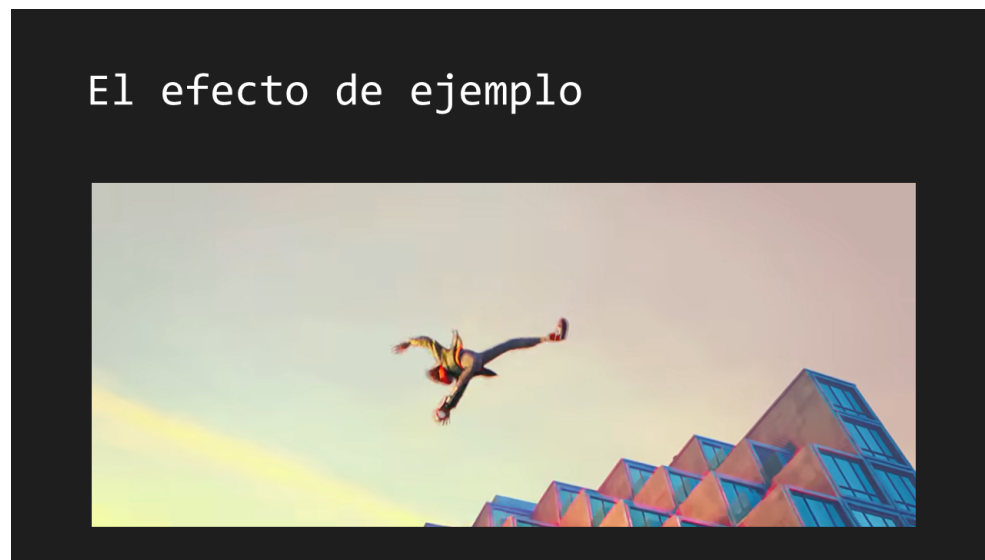


Figura A.15: Diapositiva 7 de la presentación de introducción al curso

Componente de volumen

```
[VolumeComponentMenuForRenderPipeline("Custom/Spider-verse/3D Glasses focus", typeof(UniversalRenderPipeline))]
@ Unity Script | 2 references
public class RedBlueFocusComponent : VolumeComponent, IPostProcessComponent
{
    [Header("Into the Spider-Verse focus")]
    public ClampedFloatParameter intensity = new ClampedFloatParameter(0.5f, 0, 1, true);
    public FloatParameter Thickness = new FloatParameter(1, true);
    public ClampedFloatParameter DepthThreshold = new ClampedFloatParameter(0.5f, 0, 1, true);
    public FloatParameter DepthThickness = new FloatParameter(1, true);
    public Vector2Parameter Focus = new Vector2Parameter(new Vector2());
    public ColorParameter RightTint = new ColorParameter(new Color());
    public ColorParameter LeftTint = new ColorParameter(new Color());

    0 references
    public bool IsActive()
    {
        return true;
    }

    0 references
    public bool IsTileCompatible()
    {
        return false;
    }
}
```

Define los parámetros y cómo se muestran en el inspector

Figura A.16: Diapositiva 8 de la presentación de introducción al curso

Scriptable Render Feature

```
public class FocusRenderFeature : ScriptableRenderFeature
{
    [SerializeField]
    private Shader focus_shader;
    private Material focus_material;

    private FocusRenderPass custom_pass;

    0 references
    public override void AddRenderPasses(ScriptableRenderer renderer, ref RenderingData renderingData)
    => renderer.EnqueuePass(custom_pass);

    //Inicializar el pase y los materiales que necesita
    0 references
    public override void Create()
    {
        //Crea un material que se borrará cuando se active y se llame al renderer feature
        focus_material = CoreUtils.CreateEngineMaterial(focus_shader);

        //Crear el pase y darle el material
        custom_pass = new FocusRenderPass(focus_material);
    }

    //BORRAR el material, si no se leea a DISCO, ni siquiera a memoria
    0 references
    protected override void Dispose(bool disposing) => CoreUtils.Destroy(focus_material);

    0 references
    public override void SetupRenderPasses(ScriptableRenderer renderer, in RenderingData renderingData)
    {
        if (renderingData.cameraData.cameraType == CameraType.Game) //evitar hacer el efecto de post processing en el editor de escena
        {
            custom_pass.ConfigureInput(ScriptableRenderPassInput.Depth);
            custom_pass.ConfigureInput(ScriptableRenderPassInput.Color);
            custom_pass.SetTarget(renderer.cameraColorTargetHandle, renderer.cameraDepthTargetHandle);
        }
    }
}
```

Inicializa y orquesta varios pases.

*Aquí sólo es uno

Figura A.17: Diapositiva 9 de la presentación de introducción al curso

Scriptable Render Pass

```
RedBlueFocusComponent volume_config;
0 references
public override void Execute(ScriptableRenderContext context, ref RenderingData renderingData)
{
    VolumeStack stack = VolumeManager.Instance.stack;
    volume_config = stack.GetComponent<RedBlueFocusComponent>();

    //Sirve para definir las acciones que hará unity con la librería de gráficas.
    //Puede que decida hacer las cosas en diferente orden o aprovechar a paralelizar lo que pueda.
    CommandBuffer cmd_buff = CommandBufferPool.Get();

    //Esto controla como lo que corre acá se muestra en el frame debugger.
    //Obviamente se loguea lo que le mandamos a hacer al command buffer
    using (new ProfilingScope(cmd_buff,
        new ProfilingSampler("Custom Post Process Effects/Spider-Man Focus")))
    {
        if (volume_config)
        {
            focus_material.SetFloat("_Intensity", volume_config.intensity.value);
            focus_material.SetFloat("_Thickness", volume_config.Thickness.value);
            focus_material.SetFloat("_DepthThickness", volume_config.DepthThickness.value);
            focus_material.SetFloat("_DepthThreshold", volume_config.DepthThreshold.value);
            focus_material.SetVector("_FocusPosition", volume_config.Focus.value);
            focus_material.SetColor("_RightTint", volume_config.RightTint.value);
            focus_material.SetColor("_LeftTint", volume_config.LeftTint.value);
        }
        Blitter.BlitCameraTexture(cmd_buff, CameraColorTargetHandle, CameraColorTargetHandle, focus_material, 0);
    }
    context.ExecuteCommandBuffer(cmd_buff);
    cmd_buff.Clear();
}
```

Define lo que corre en la GPU (Qué y cuándo)

Recolecta y aplica la configuración

Figura A.18: Diapositiva 10 de la presentación de introducción al curso

Shaders!

Escribir shaders se trata sobre entender funciones y transformarlas

[Desmos!](#)




Figura A.19: Diapositiva 11 de la presentación de introducción al curso

HLSL

High Level Shader Language Esencialmente C, pero orientado a GPUs.
Igual, pero diferente a GLSL.
(pero de Microsoft)

Algunas features:

 a la documentación

```
col = col.zyxw;
```

```
float4 _RightTint;  
float4 _LeftTint;  
float2 _FocusPosition;
```

Returns the partial derivative of the specified value with respect to the screen-space x-coordinate.

```
ret ddx(x)
```

Figura A.20: Diapositiva 12 de la presentación de introducción al curso

El Shader:

Ver demo:

- Mostrar las UVs de la pantalla
- Macros de las depth textures
- Mostrar la depth textura
- Mostrar Sobel (Ejercicio, outlines!)
- URP is weird

Figura A.21: Diapositiva 13 de la presentación de introducción al curso

¿Realmente tenemos el efecto deseado?



Figura A.22: Diapositiva 14 de la presentación de introducción al curso

Referencias

- <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.1/manual/containers/create-custom-renderer-feature-1.html>
- <https://www.youtube.com/watch?v=9fa4uFm1eCE>
- <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.1/manual/integration-with-post-processing.html>
- <https://docs.unity3d.com/2020.1/Documentation/Manual/SL-DepthTextures.html>
- <https://www.youtube.com/watch?v=RMT6DcaMxcE>
- <https://github.com/Unity-Technologies/Graphics/blob/master/Packages/com.unity.render-pipelines.universal/Documentation~/renderer-features/how-to-fullscreen-blit.md>
- <https://www.youtube.com/watch?v=yUVrtPCsCb0>
- [High-level shader language \(HLSL\) - Win32 apps | Microsoft Learn](https://docs.microsoft.com/en-us/windows/win32/direct3dshader/high-level-shader-language-hlsl)
- https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview

Figura A.23: Diapositiva 15 de la presentación de introducción al curso

A.3. Presentación de creación de *shaders*

A.3.1. Enlace a repositorio de demostración usado para impartir el tema de creación de *shaders*

Enlace al repositorio de demostración usado durante la presentación(Unity 2022.3.3f1):

<https://github.com/JmDeleon2000/Shader-sandbox>

A petición popular: *Shaders*

Jorge De León, 8/9/2023

Figura A.24: Diapositiva 1 de la presentación de introducción al curso

¿Qué es un *Shader*?

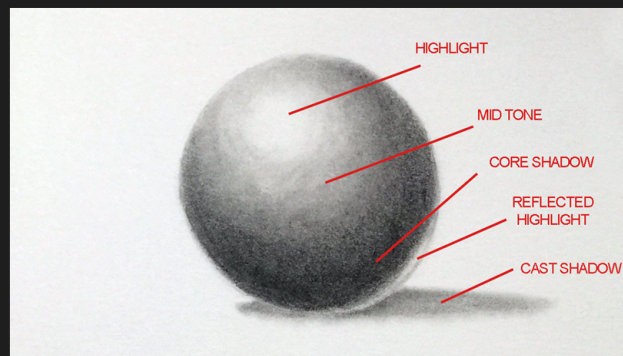


Foto de: <https://thevirtualinstructor.com/shading-techniques-basics.html>

Figura A.25: Diapositiva 2 de la presentación de introducción al curso

Fuera de la etimología:

Es un programa que corre en la GPU. ¿Para hacer qué?

- Compute *Shaders*
- *Shaders, shaders*
- Teselación
- *RayTracing*
- *PostProcessing*
- Farmeo de Bitcoin
- Etc...

(El uso de la
palabra ya no
tiene sentido)

Figura A.26: Diapositiva 3 de la presentación de introducción al curso

El “*Shader, shader*”

Las cosas que más
comúnmente se programan

*pintamos fragmentos,
no pixeles.

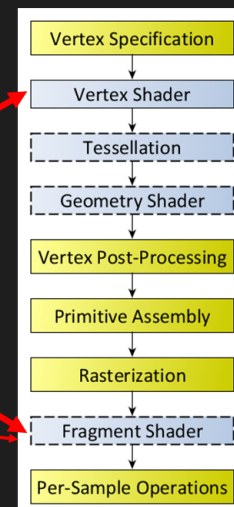



Figura A.27: Diapositiva 4 de la presentación de introducción al curso

El "Shader, shader"



knits angrily

Hay por lo menos un fragmento por pixel, pero por cosas como *anti aliasing* puede que se tomen varias muestras para un mismo pixel. (Para nuestros propósitos: una distinción sin diferencia)



```

graph TD
    A[Vertex Specification] --> B[Vertex Shader]
    B --> C[Tessellation]
    C --> D[Geometry Shader]
    D --> E[Vertex Post-Processing]
    E --> F[Primitive Assembly]
    F --> G[Rasterization]
    G --> H[Fragment Shader]
    H --> I[Per-Sample Operations]
  
```

Figura A.28: Diapositiva 5 de la presentación de introducción al curso

El Shader, shader:

Parte 1 (*shader* de vértices):

- Espacio de objeto → Espacio de mundo*
- Asociar UVs a vértices
- Cualquier otra cosa que convenga

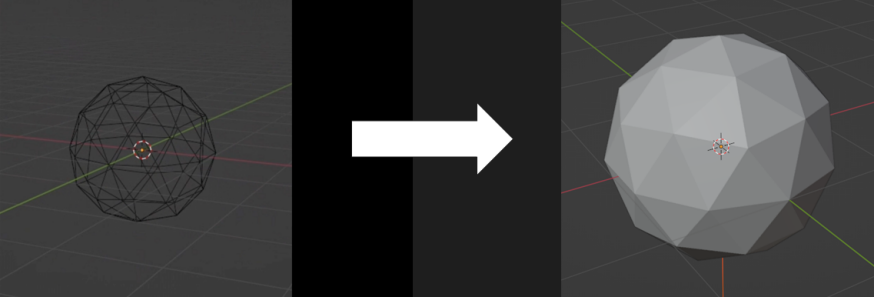


Figura A.29: Diapositiva 6 de la presentación de introducción al curso

El Shader, shader:

Parte 2 (*shader* de fragmentos):

- Definir qué color va dónde
- Cualquier cosa que convenga

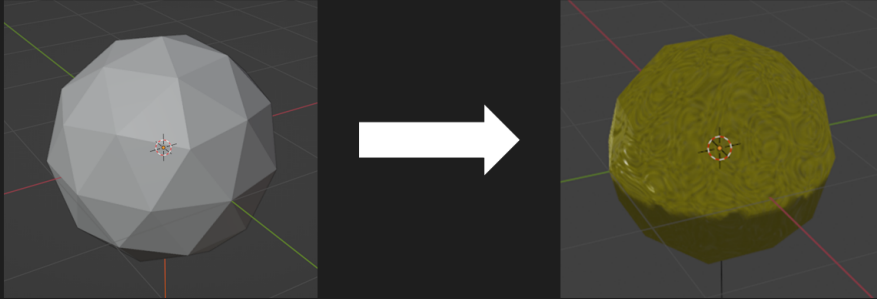
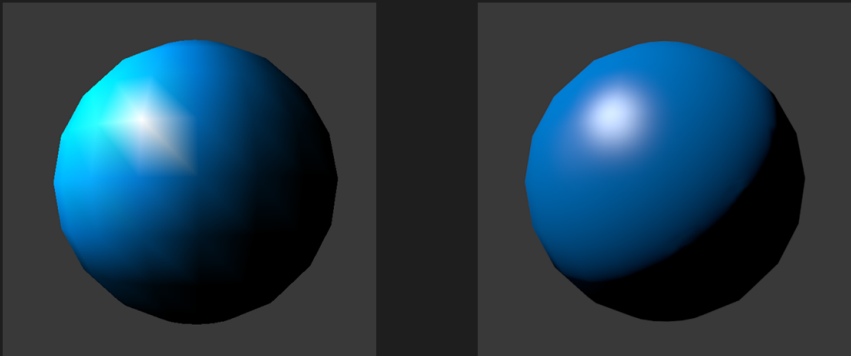


Figura A.30: Diapositiva 7 de la presentación de introducción al curso

“Cualquier otra cosa que convenga”
Pero ¿dónde conviene más?



¿Cuál es la diferencia entre estas imágenes?

Figura A.31: Diapositiva 8 de la presentación de introducción al curso

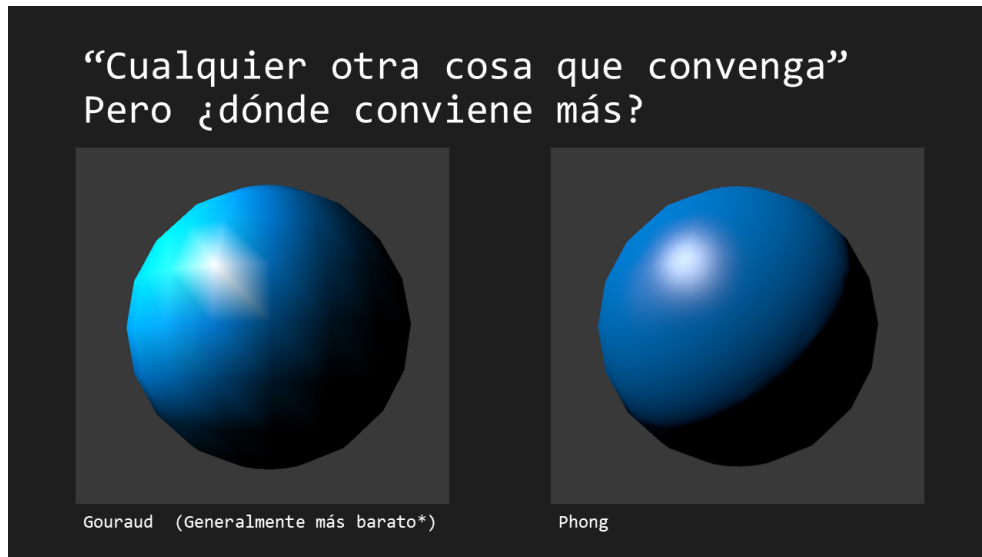


Figura A.32: Diapositiva 9 de la presentación de introducción al curso

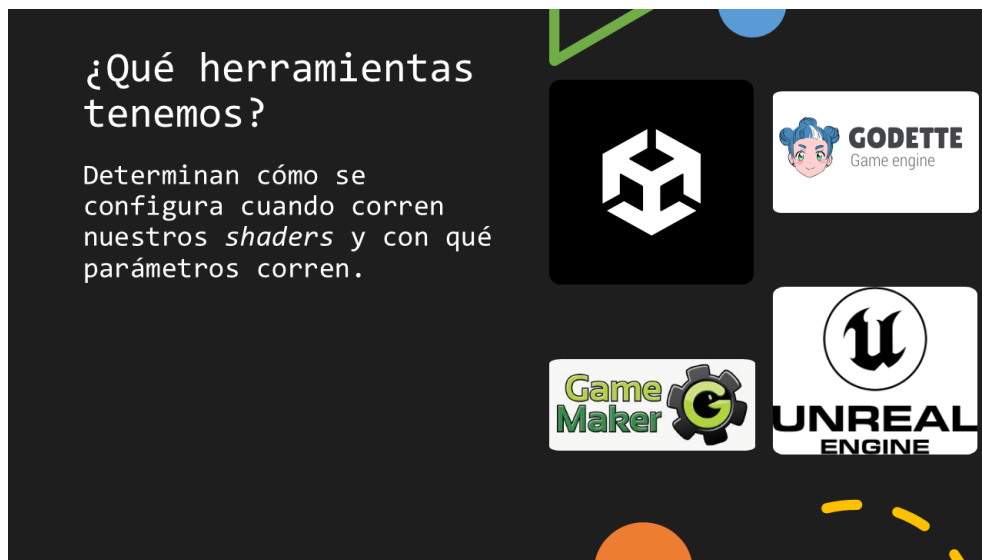


Figura A.33: Diapositiva 10 de la presentación de introducción al curso



Figura A.34: Diapositiva 11 de la presentación de introducción al curso



Figura A.35: Diapositiva 12 de la presentación de introducción al curso

¿Qué herramientas tenemos?

- Vectores
- Funciones
- Matrices
- Espacios



Figura A.36: Diapositiva 13 de la presentación de introducción al curso

Preguntas importantes:

¿Qué es un color?

¿Qué son las normales?

¿Qué son las rotaciones?

¿Qué es una convolución?

¿Qué son las UVs?

¿Qué son las imágenes?

Figura A.37: Diapositiva 14 de la presentación de introducción al curso

Preguntas importantes:

El color es un vector

RGBA → XYZW

¿Qué son las normales?

¿Qué son las rotaciones?

¿Qué es una convolución?

¿Qué son las UVs?

¿Qué son las imágenes?

Figura A.38: Diapositiva 15 de la presentación de introducción al curso

Preguntas importantes:

El color es un vector

RGBA → XYZW

Las normales son un vector

(Vector normal)

¿Qué son las rotaciones?

¿Qué es una convolución?

¿Qué son las UVs?

¿Qué son las imágenes?

Figura A.39: Diapositiva 16 de la presentación de introducción al curso

Preguntas importantes:

El color es un vector RGBA → XYZW	Las normales son un vector (Vector normal)
Las rotaciones son matrices ↻ → $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	¿Qué es una convolución?
¿Qué son las UVs?	¿Qué son las imágenes?

Figura A.40: Diapositiva 17 de la presentación de introducción al curso

Preguntas importantes:

El color es un vector RGBA → XYZW	Las normales son un vector (Vector normal)
Las rotaciones son matrices ↻ → $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	Las convoluciones son matrices (Y loops)
¿Qué son las UVs?	¿Qué son las imágenes?

Figura A.41: Diapositiva 18 de la presentación de introducción al curso

Preguntas importantes:

El color es un vector RGBA → XYZW	Las normales son un vector (Vector normal)
Las rotaciones son matrices ↻ → $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	Las convoluciones son matrices (Y loops)
Las UVs son vectores XY	¿Qué son las imágenes?

Figura A.42: Diapositiva 19 de la presentación de introducción al curso

Preguntas importantes:

El color es un vector RGBA → XYZW	Las normales son un vector (Vector normal)
Las rotaciones son matrices ↻ → $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	Las convoluciones son matrices (Y loops)
Las UVs son vectores XY	Las imágenes son matrices de color (vectores) (Las podemos leer con UVs)

Figura A.43: Diapositiva 20 de la presentación de introducción al curso

La Cheat Sheet:

El color es un vector

RGBA → XYZW

Las normales son un vector

(Vector normal)

Las rotaciones son matrices


$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Las convoluciones son matrices

(Y loops)

Las UVs son vectores

XY

Las imágenes son matrices de color (vectores)

(Las podemos leer con UVs)

*No está ni cerca de estar completa

Figura A.44: Diapositiva 21 de la presentación de introducción al curso

Sí, es Álgebra Lineal, pero en la GPU

Pros:

Mucha computación
paralela
Hardware especializado

Cons:

Bottlenecks
No branching



Figura A.45: Diapositiva 22 de la presentación de introducción al curso

HLSL

High Level Shader Language Esencialmente C, pero orientado a GPUs.
Igual, pero diferente a GLSL.
(pero de Microsoft)

Algunas features:

🔗 a la documentación

La aritmética es por componente.

```
col = col.zyxw;
```

```
float4 _RightTint;
```

```
float4 _LeftTint;
```

```
float2 _FocusPosition;
```

Figura A.46: Diapositiva 23 de la presentación de introducción al curso

Antes de ver código:
¿Cómo debuggeamos?

No existe print()

Ni siquiera nos sale segfault

Figura A.47: Diapositiva 24 de la presentación de introducción al curso

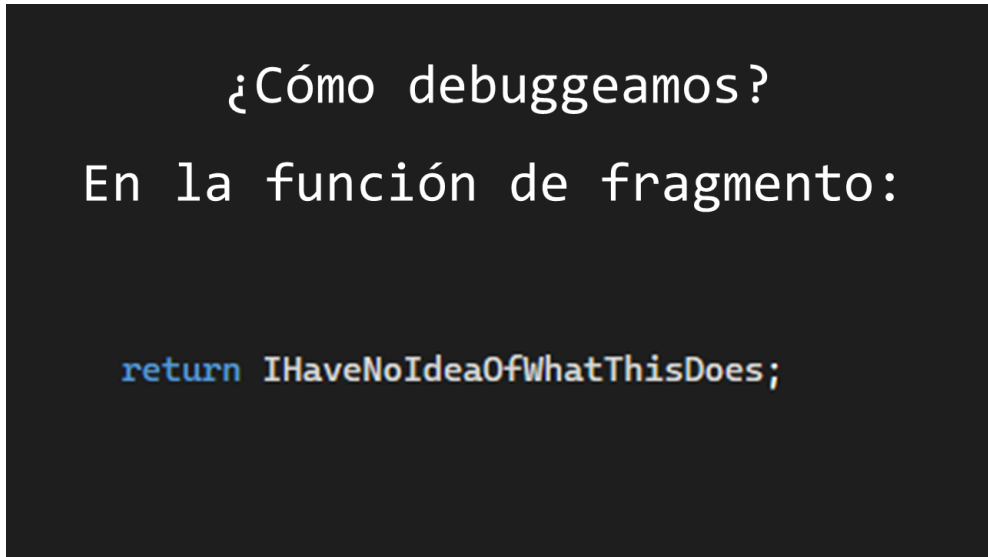


Figura A.48: Diapositiva 25 de la presentación de introducción al curso

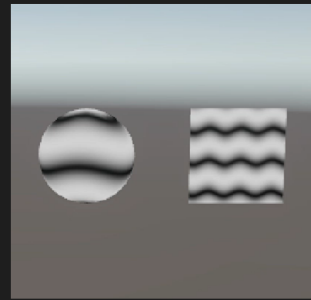


Figura A.49: Diapositiva 26 de la presentación de introducción al curso

Algunas funciones útiles:

```
sin(x) y cos(x)
abs(x)
step(x, max)
smoothstep(min, max, x)
pow(x, y)
ceil(x)
floor(x)
frac(x)
reflect(x)
lerp(x, y, t)
saturate(x)
noise(x)
```

[Para más info, ver la referencia de HLSL](#)

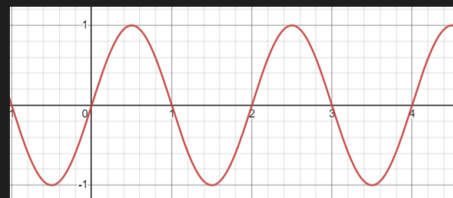


```
float4 offset = (sin((i.uv.x + _Time.a) * 3 * pi * 2)
+ 0.5 + 0.5) * 0.05;
return sin((offset + i.uv.y + _Time.y) * 3 * pi * 2)
* 0.5 + 0.5;
```

Figura A.50: Diapositiva 27 de la presentación de introducción al curso

$\text{pow}(x, y)$

$\sin(x \pi)$



$\sin(x \pi)^{32}$

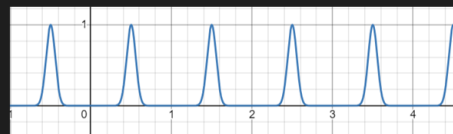


Figura A.51: Diapositiva 28 de la presentación de introducción al curso

El secreto:

Considerar el dominio y el contradominio de las funciones.

$$\cos(\mathbb{R}) \Rightarrow [-1, 1]$$

Colores son $[0, 1]$

Figura A.52: Diapositiva 29 de la presentación de introducción al curso

Ejercicio:

$$f(\cos(\mathbb{R})) \Rightarrow [0, 1]$$

$$f(x) := ?$$

Figura A.53: Diapositiva 30 de la presentación de introducción al curso

Uniforms

Información UNIFORME para todos los hilos.

(Constantes que tienen el mismo valor para todo vértice y fragmento)

¿Cómo se usan? Depende de la versión
En Unity:

```
Properties
{
    _MainTex ("Foo Texture", 2D) = "white" {}
    _Foo ("Bar parameter", Range(-1, 100)) = 0
}
```

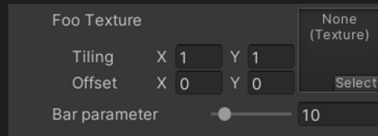


Figura A.54: Diapositiva 31 de la presentación de introducción al curso

SEMANTICS

Nos permiten tener nombres únicos para nuestras funciones y variables, al mismo tiempo que manejamos qué entre y sale de nuestros *shaders*.

```
struct VertexInput
{
    float4 vertex : POSITION;
    float2 uv : TEXCOORD0;
    float3 normal : NORMAL;
};
```



Este NO es

```
#pragma fragment frag
fixed4 frag (FragInput i) : SV_Target
{
```



Este sí

Figura A.55: Diapositiva 32 de la presentación de introducción al curso

Swizzling



```
return col. rgba;
```



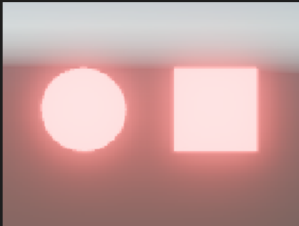
```
return col. bgra;
```

Figura A.56: Diapositiva 33 de la presentación de introducción al curso

Emisión de luz

*Muy específico a Unity URP:

Si es mayor a 1, no solo es el color, también brilla.



```
return float4(10, 1, 1, 1);
```

Si no queremos que pase, podemos usar saturate(x)

Figura A.57: Diapositiva 34 de la presentación de introducción al curso

Normal

```
return float4(i.normal, 1);
```



Figura A.58: Diapositiva 35 de la presentación de introducción al curso

Profundidad

ZWrite

On

Off

ZTest

Less

LEqual

Equal

GEqual

Greater

Always

NotEqual

Default

La única ilustración que sirve para esto:



Figura A.59: Diapositiva 36 de la presentación de introducción al curso

Blending y transparencia

```
Blend SrcAlpha OneMinusSrcAlpha // Traditional transparency
Blend One OneMinusSrcAlpha // Premultiplied transparency
Blend One One // Additive
Blend OneMinusDstColor One // Soft additive
Blend DstColor Zero // Multiplicative
Blend DstColor SrcColor // 2x multiplicative
```


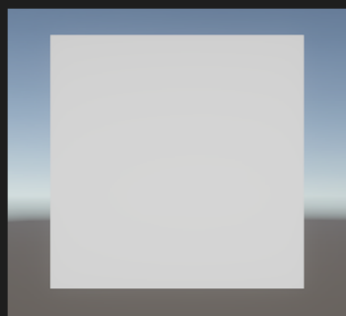
Configurar esto en Unity es arcano, molesto y cosa de calle. Buena cheat sheet 
[Unity - Manual: ShaderLab command: Blend \(unity3d.com\)](#)

Figura A.60: Diapositiva 37 de la presentación de introducción al curso

Blending y transparencia



Blend OneMinusDstColor One ~



Blend One One ~

Figura A.61: Diapositiva 38 de la presentación de introducción al curso



Figura A.62: Diapositiva 39 de la presentación de introducción al curso

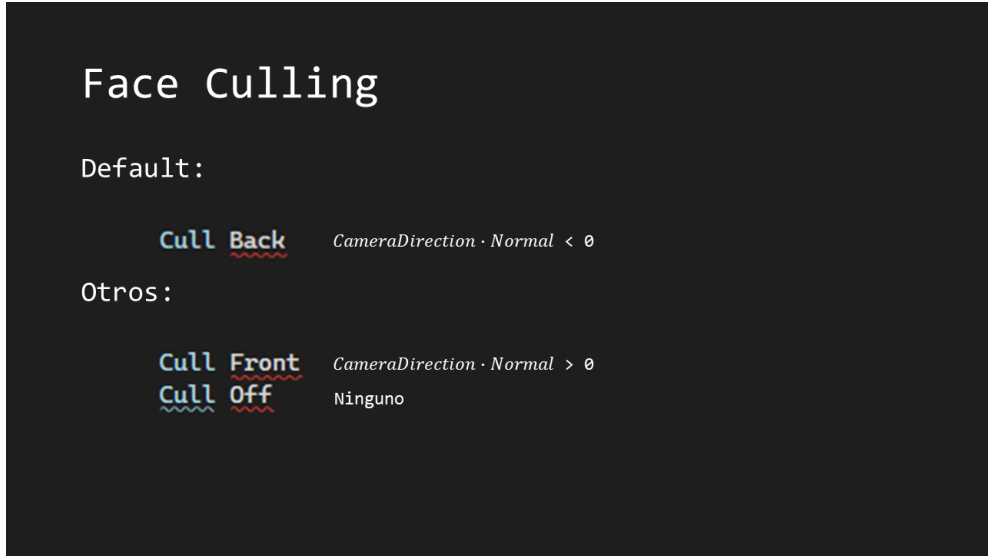


Figura A.63: Diapositiva 40 de la presentación de introducción al curso



Figura A.64: Diapositiva 41 de la presentación de introducción al curso

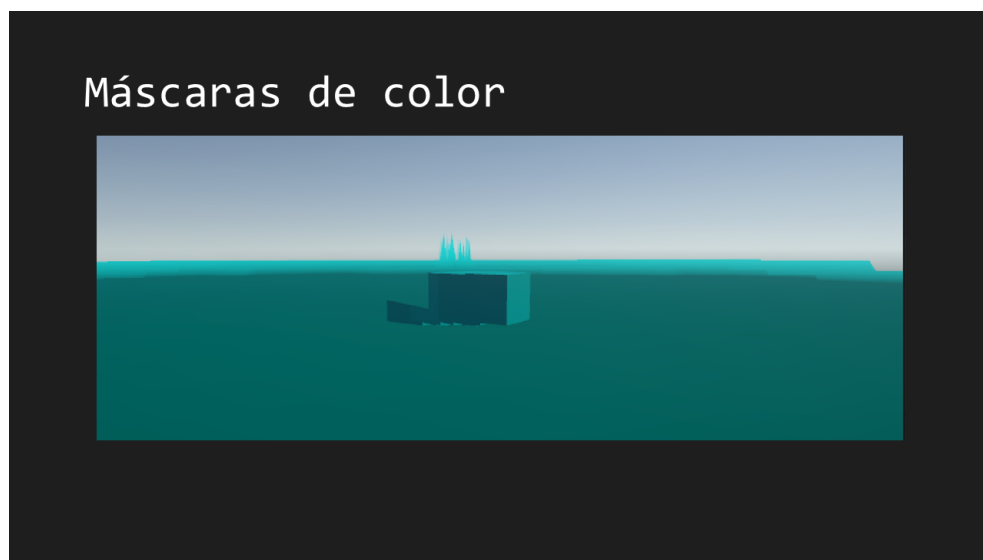


Figura A.65: Diapositiva 42 de la presentación de introducción al curso

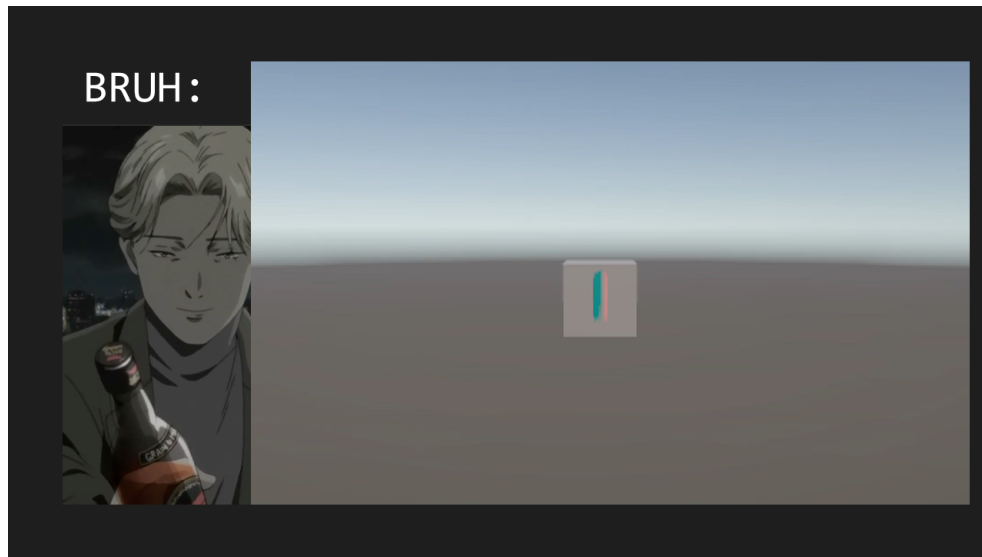


Figura A.66: Diapositiva 43 de la presentación de introducción al curso



Figura A.67: Diapositiva 44 de la presentación de introducción al curso

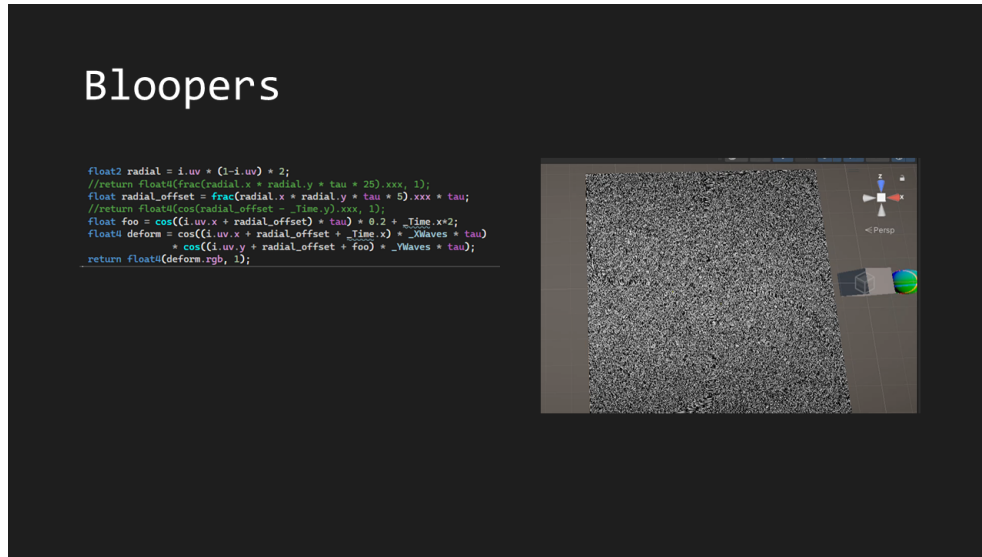


Figura A.68: Diapositiva 45 de la presentación de introducción al curso

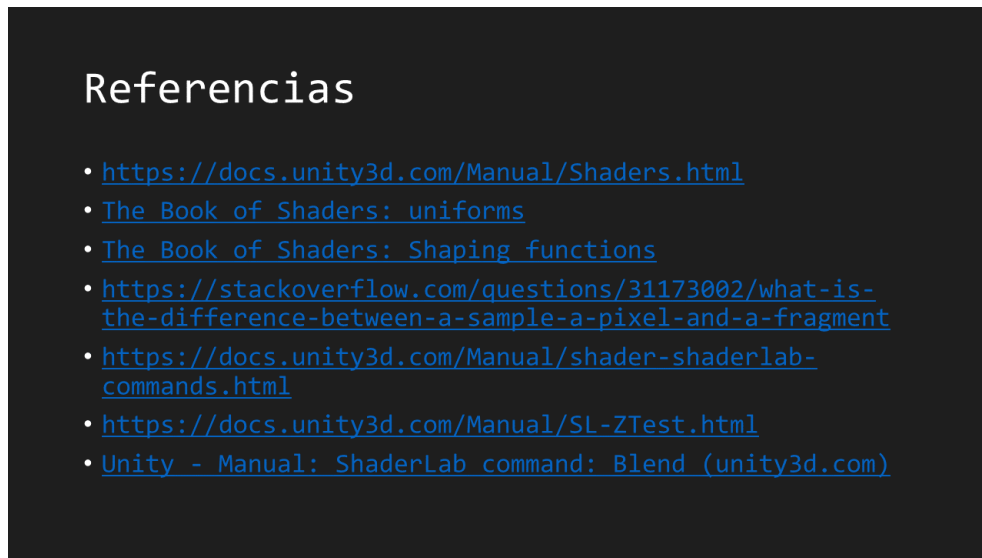


Figura A.69: Diapositiva 46 de la presentación de introducción al curso

A.3.2. Listado de funciones de HLSL cubiertas en la presentación para el tema de creación de *shaders*

Funciones:

1. $\sin(x)$ y $\cos(x)$
2. $\text{abs}(x)$
3. $\text{step}(x, \text{max})$

-
4. `smoothstep(min, max, x)`
 5. `pow(x, y)`
 6. `ceil(x)`
 7. `floor(x)`
 8. `frac(x)`
 9. `reflect(x)`
 10. `lerp(x, y, t)`
 11. `saturate(x)`

A.3.3. Listado de técnicas cubiertas en la presentación para el tema de creación de *shaders*

Técnicas:

1. *Swizzling*
2. Emisión de luz
3. Uso de normales
4. Deformación de modelos
5. Uso de UVs y texturas en aplicaciones que no son colorear un objeto.

A.3.4. Listado de configuraciones cubiertas en la presentación para el tema de creación de *shaders*

Configuraciones específicas a Unity:

1. Uniforms e interfaz de Unity para parametrizarlas
2. Semánticas
3. Manejo del *buffer* de profundidad
4. Transparencia y mezclado
5. *Face Culling*
6. Máscaras de color

Bitácora de ejecución de metodología

Actividad	Lunes	Martes	Miércoles	Jueves	Viernes
Preparación de material tema Post-Processing	2023-06-26	2023-06-27	2023-06-28	2023-06-29	2023-06-30
	Lunes	Martes	Miércoles	Jueves	Viernes
Preparación de material tema Post-Processing	2023-07-03	2023-07-04	2023-07-05	2023-07-06	2023-07-07
	Lunes	Martes	Miércoles	Jueves	Viernes
Preparación de material tema Post-Processing	2023-07-10	2023-07-11	2023-07-12	2023-07-13	2023-07-14
Preparación de material tema Post-Processing					
Presentación material tema Post-Processing					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-07-17	2023-07-18	2023-07-19	2023-07-20	2023-07-21
Recopilación de retroalimentación de material sobre Post-Processing					
Preparación de material tema Shaders					
	Lunes	Martes	Miércoles	Jueves	Viernes
Preparación de material tema Shaders	2023-07-24	2023-07-25	2023-07-26	2023-07-27	2023-07-28
	Lunes	Martes	Miércoles	Jueves	Viernes
Preparación de material tema Shaders	2023-07-31	2023-08-01	2023-08-02	2023-08-03	2023-08-04
	Lunes	Martes	Miércoles	Jueves	Viernes
Preparación de material tema Shaders	2023-08-07	2023-08-08	2023-08-09	2023-08-10	2023-08-11
	Lunes	Martes	Miércoles	Jueves	Viernes
Presentación material tema Shaders	2023-08-14	2023-08-15	2023-08-16	2023-08-17	2023-08-18
Recopilación de retroalimentación de material sobre Shaders					

Figura B.1: Bitácora de ejecución (continuado)

Actividad	Lunes	Martes	Miércoles	Jueves	Viernes
Definición de próximo tema a desarrollar	2023-08-21	2023-08-22	2023-08-23	2023-08-24	2023-08-25
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-08-27	2023-08-28	2023-08-29	2023-08-30	2023-08-31
Preparación de material tema Desarrollo de efectos visuales avanzados					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-09-03	2023-09-04	2023-09-05	2023-09-06	2023-09-07
Preparación de material tema Desarrollo de efectos visuales avanzados					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-09-11	2023-09-12	2023-09-13	2023-09-14	2023-09-15
Preparación de material tema Desarrollo de efectos visuales avanzados					
Redacción de informe (Temas de shader y post-processing)					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-09-18	2023-09-19	2023-09-20	2023-09-21	2023-09-22
Presentación de material tema Desarrollo de efectos visuales avanzados					
Redacción de informe (Temas de shader y post-processing)					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-09-18	2023-09-19	2023-09-20	2023-09-21	2023-09-22
Redacción de informe (Temas de shader y post-processing)					
Recopilación de retroalimentación de material sobre Desarrollo de efectos visuales avanzados					

Figura B.2: Bitácora de ejecución (continuado)

Actividad	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-09-25	2023-09-26	2023-09-27	2023-09-28	2023-09-29
Redacción de informe (Tema de efectos visuales avanzados)					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-10-02	2023-10-03	2023-10-04	2023-10-05	2023-10-06
Redacción de informe					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-10-09	2023-10-10	2023-10-11	2023-10-12	2023-10-13
Redacción de informe					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-10-16	2023-10-17	2023-10-18	2023-10-19	2023-10-20
Redacción de informe					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-10-23	2023-10-24	2023-10-25	2023-10-26	2023-10-27
Redacción de informe					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-10-30	2023-10-31	2023-11-01	2023-11-02	2023-11-03
Redacción de informe					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-11-06	2023-11-07	2023-11-08	2023-11-09	2023-11-10
Revisión de la primer versión del informe					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-11-13	2023-11-14	2023-11-15	2023-11-16	2023-11-17
Revisión de la primer versión del informe					
Corrección de informe en base a las correcciones recibidas					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-11-20	2023-11-21	2023-11-22	2023-11-23	2023-11-24
Corrección de informe en base a las correcciones recibidas					
	Lunes	Martes	Miércoles	Jueves	Viernes
	2023-11-27	2023-11-28	2023-11-29	2023-11-30	2023-12-01
Corrección de informe en base a las correcciones recibidas					
Preparación de presentación a terna					
	Lunes				
	2023-12-04				
Presentación a terna					

Figura B.3: Bitácora de ejecución.