



Centres universitaris adscrits a la



Grau en Disseny i Producció de Videojocs

**Creació d'una eina per desenvolupar tenint en compte
discapacitats visuals
MEMÒRIA FINAL**

**Joel Torrent Barrero
Tutor: Rafael González**



Abstract

This project studies the disability known as color vision exploring different solutions inside videogames and other real life situations in order to create a development tool for Unity to develop games considering people with this anomaly easier, from filters that simulate the problems to other front-end functionalities for the build version of the game.

Resum

Aquest projecte analitza la discapacitat coneguda com daltonisme explorant diferents solucions existents tant dins com fora del món dels videojocs, junt amb fluxos de treball i directrius d'accessibilitat en medis interactius establertes amb l'objectiu de crear una eina pel motor Unity que faciliti als desenvolupadors tenir en compte a la part del públic afectada durant el desenvolupament amb funcionalitats de fàcil implementació.

Resumen

Este proyecto analiza la discapacidad conocida como daltonismo explorando distintas soluciones dentro y fuera del mundo del videojuego junto a flujos de trabajo y directrices relacionados con la accesibilidad en medios interactivos con el fin de crear una herramienta para que los desarrolladores puedan producir juegos teniendo en cuenta a las personas afectadas por esta anomalía, con filtros que muestren el problema y soluciones a este para el resultado final de fácil implementación.

Índex

Índex de figures.....	V
Índex de taules.....	IX
1 Introducció.....	1
2 Antecedents de la recerca/Referents.....	5
2.1 The Last of Us Part II.....	5
2.2 Chroma Gun i Color ADD.....	6
2.3 Unity Progrids.....	7
2.4 Celeste.....	8
3 Marc teòric.....	9
3.1 Discapacitats Visuals.....	9
3.1.1 Daltonisme.....	9
3.2 Solucions per a discapacitats visuals.....	15
3.2.1 Solucions fora del món dels videojocs.....	15
3.2.2 Solucions aplicades als videojocs.....	22
3.3 Motor de Joc: Unity.....	27
3.3.1 Què és un motor de joc?.....	27
3.3.2 Breu història de la creació de Unity.....	27
3.3.3 Unity Asset Store.....	28
3.4 Creació d'eines de desenvolupament.....	30
3.4.1 APIs.....	30
3.4.2 Desenvolupament d'eines en el motor de jocs Unity.....	30
3.4.3 Disseny i Documentació de d'eines.....	32
3.5 Programació gràfica.....	34
3.5.1 Introducció a la programació gràfica.....	34
3.5.2 Tècniques d'implementació.....	36

4	Objectius	39
4.1	Objectius Principals	39
4.2	Objectius Secundaris	39
5	Disseny metodològic i cronograma	41
5.1	Fases del projecte	41
5.1.1	Investigació	41
5.1.2	Documentació	41
5.1.3	Cerca de referents	41
5.1.4	Elecció d'eines i llibreries	41
5.1.5	Disseny i Desenvolupament de característiques	42
5.1.6	Desenvolupament d'una escena de prova	42
5.1.7	Desenvolupament d'una demo i testeig	42
5.1.8	Publicació de l'eina	42
5.2	Planificació	43
6	Desenvolupament de l'eina	45
6.1	Filtres en la Scene View	45
6.1.1	Shader de simulació de daltonisme	46
6.1.2	Aplicació del shader a la Scene View	47
6.1.3	Interfície d'usuari en la Scene View	48
6.2	Mode d'Alt Contrast	49
6.2.1	Depth buffer	49
6.2.2	Creació de vores	50
6.2.3	Implementació de shaders de contrast	50
6.2.4	Lògica del mode d'Alt Contrast	50
6.3	Codi visual de colors	53
6.3.1	Disseny del codi visual	53
6.3.2	Implementació d'un lector de color	54

6.3.3	Implementació de shaders.....	56
6.4	Documentació	58
6.5	Creació d'una escena de mostra i demostració	58
6.6	Publicació.....	60
6.7	Testeig	60
6.7.1	Construcció d'un formulari	60
6.7.2	Resultats.....	61
7	Conclusions	63
7.1	Investigació	63
7.2	Objectius	63
7.3	Testeig de l'eina	64
8	Bibliografia/Referències	65
8.1	Referents	65
8.2	Ludografia	72

Índex de figures

Figura 1.1 Característiques d'accessibilitat de Accessible Games Data Base. Font: accessiblegamesdatabase.com.	1
Figura 1.2 Filtre per daltònics en Doom. Font: iD Software.	2
Figura 2.1 Mostra de l'opció High Contrast Display dins la jugabilitat. Font: Naughty Dog.	6
Figura 2.2 Ús del sistema ColorADD en ChromaGun. Font: Pixel Maniacs.	7
Figura 2.3 Interfície d'usuari de l'eina Progrids en el motor de joc Unity. Font: Progrids.	7
Figura 2.4 Text dels desenvolupadors de Celeste informant als jugadors d'una experiència d'usuari diferent a la dissenyada. Font: Extremely OK Games.	8
Figura 3.1 Classificació de daltonisme com a discapacitat. Font: icd.who.int.	9
Figura 3.2 Diagrama del recorregut dels fotons en l'ull humà. Font: Elaboració Pròpia.	10
Figura 3.3 Diagrama de rangs de freqüències dels cons S, M i L respectivament. Font: Hyperphysics.phy-astr.gsu.edu.	11
Figura 3.4 Distribució dels tipus cons en el nervi òptic sense cap deficiència. Font: phaidon.com.	12
Figura 3.5 Representació dels defectes dels cons i espectre de colors de les tipologies de daltonisme principals. Font: Color Universal Design (CUD).	14
Figura 3.6 Matriu de transformació de color d'espai RGB a LMS. Font: Stack Overflow.	15
Figura 3.7 Fórmules de les components $\alpha\beta\gamma$ de l'equació del pla, on W és la respectiva component del color blanc en l'espai LMS i B és la respectiva component del color blau. Font: Viénot, Brettel i Mollon.	16
Figura 3.8 Procés de generació de matrius de transformació lineal pel cas de Protanòpsia. Font: Elaboració pròpia.	17
Figura 3.9 Exemple d'addició de figures del sistema ColorADD. Font: coloradd.net.	18
Figura 3.10 Combinacions possibles del sistema ColorADD. Font: Elaboració pròpia.	19

Figura 3.11 Lents de les ulleres EnChroma per diferents tipologies de daltonisme. Font: eu.enchroma.com.....	19
Figura 3.12 Tests de daltonisme en 3 casos: Visió Normal, Dicromàcia-Deuteranopia i aplicació Chroma, respectivament. Font: UbiComp '14 Adjunct... 21	21
Figura 3.13 Optatron: escàner de text portàtil per persones cegues. Font: Youtube.	22
Figura 3.14 Elements escalables ressaltats en color blanc en Uncharted 4. Font: Game Career Guide.	25
Figura 3.15 Formulari d'Excel de Game Accessibility Guidelines. Font: Game Accessibility Guidelines.	26
Figura 3.16 Unity Versió 1.0 "Next-Gen" Demo. Font: forum.unity.com.....	28
Figura 3.17 Mostra d'un atribut d'editor per amagar una variable pública en l'inspector del motor. Font: Unity.	32
Figura 3.18 Etapes del procés de renderització. Font: tech.metail.com.	35
Figura 3.19 Escena sense post-processat (esquerra) comparada amb post-processat (dreta). Font: Unity.	36
Figura 3.20 Representació del buffer de profunditat (dreta) respecte a una escena (esquerra). Font: GitHub.....	37
Figura 3.21 Problemes de silueta amb normals no suavitzades. Font: GameDevStackExchange.....	38
Figura 3.22 Dues metodologies diferents de detecció de vores aplicades a una imatge original (esquerra). Font: TowardsDataScience.....	38
Figura 6.1 Filtres d'editor. 1-Tricromàcia, 2-Protanopia 3-Deuteranopia, 4-Tritanopia, 5-Acromatòpsia. Font: Elaboració Pròpia	45
Figura 6.2 Funció que retorna la matriu de reducció d'espai de color segons una paraula clau. Font: Elaboració pròpia.	46
Figura 6.3 Fragment shader del shader final que aplica els filtres. Font: Elaboració pròpia.	47
Figura 6.4 Atributs de la classe que aplica el shader anterior. Font: Elaboració pròpia.	47
Figura 6.5 Funció retràctil de l'eina de filtres. Font: Elaboració pròpia.	48
Figura 6.6 Tooltip de l'eina de filtres. Font: Elaboració Pròpia.....	48
Figura 6.7 Mode d'alt contrast implementat en l'eina. Font: Elaboració pròpia....	49

Figura 6.8 Funció de canvi de valors del depth buffer. Font: Elaboració pròpia. .	49
Figura 6.9 Càlcul de la vora a través de la profunditat i normals al voltant d'un píxel. Font: Elaboració pròpia.	50
Figura 6.10 Controlador del mode Alt contrast. Font: elaboració pròpia.....	51
Figura 6.11 Scriptable object per definir Tags i Colors. Font: Elaboració pròpia.	52
Figura 6.12 Inspector de l'objecte amb el seu Renderer i el component que reescriu el color en el mode d'Alt Contrast. Font: Elaboració Pròpia	52
Figura 6.13 Demostració del problema de rotació del sistema ColorAdd. Font: Elaboració Pròpia	53
Figura 6.14 Explicació del nou codi visual de colors. Font: Elaboració Pròpia	54
Figura 6.15 Funcionalitat de lector de color. Font: Elaboració pròpia.....	54
Figura 6.16 Configuració del sistema en Unity. Font: Elaboració pròpia.	55
Figura 6.17 Umbrals definits pels paràmetres del controlador de les figures. Font: Elaboració pròpia.	55
Figura 6.18 Figures corresponents al component Hue més proper. Font: Elaboració pròpia.	56
Figura 6.19 Demostració del shader que pinta la figura sobre la geometria. Font: Elaboració Pròpia.....	56
Figura 6.20 Àrea obtinguda amb els bounds de l'objecte, retallada de forma quadrada en cas del rectangle amb vora discontinua. Font: Elaboració pròpia. ...	57
Figura 6.21 Component que s'utilitza per definir la figura pintada en la geometria. Font: Elaboració pròpia.	57
Figura 6.22 Portada de la documentació de l'eina. Font: Elaboració pròpia.....	58
Figura 6.23 Escena de mostra inclosa en l'eina. Font: Elaboració pròpia.	58
Figura 6.24 Sales que formen la demo interactiva de l'eina. Font: Elaboració pròpia.	59

Índex de taules

Taula 5.1 Cronograma de les fases del projecte. Font: Elaboració Pròpia. 43

1 Introducció

L'accessibilitat en els videojocs no sempre ha estat un tòpic popular. Abans de l'última dècada els jocs que donaven suport a diferents discapacitats eren considerats més seriosos (Miesenberger, Ossmann, Archambault, Searle i Holsinger, 2008). És en els últims anys quan l'accessibilitat en videojocs comença a guanyar popularitat i a ser un tòpic que arriba al públic general, com per exemple la creació del premi Innovation in Accessibility en la gala de The Game Awards al 2020 (Campbell, 2020).

El medi principal de reproducció de videojocs és a través de pantalles (TV o Monitor), mostrant tot un espectre de colors a través dels seus píxels. Sent els colors un factor clau, és aquí on apareix el problema del daltonisme, una condició que arriba a afectar al 8% dels homes i al 0.5% de les dones. Essent una de les discapacitats visuals hereditàries més comuns, sense un tractament efectiu i generant desavantatges realitzant tasques visuals (Simunovic, 2009) és un potencial objecte d'estudi per al desenvolupament de videojocs.

Iniciatives recents com accessiblegamesdatabase.com, publicada el 2020, reuneixen les opcions d'accessibilitat de jocs comercials que els desenvolupadors ofereixen a l'usuari final. Per mesures de daltonisme, la plataforma ofereix les opcions de contrast, color de HUD personalitzable i filtres per daltònics (Figura 1.1).

General ^	Fine Motor ^	Visual ^
<input type="checkbox"/> Save Progress +	<input type="checkbox"/> Change Repeated Taps +	<input checked="" type="checkbox"/> Contrast Settings +
<input type="checkbox"/> Dedicated Accessibility +	<input type="checkbox"/> Change Button Holds To Toggles +	<input type="checkbox"/> Adjustable Interface Text +
<input type="checkbox"/> Tutorials +	<input type="checkbox"/> Quick-Time Events (QTEs) +	<input type="checkbox"/> Narrator +
<input type="checkbox"/> Pause Gameplay +	<input type="checkbox"/> Controller Customization +	<input type="checkbox"/> Screen Magnifier +
<input type="checkbox"/> Difficulty Options +	<input type="checkbox"/> Keyboard Rebinding +	
<input type="checkbox"/> Internet Connection Required +	<input type="checkbox"/> Mouse Rebinding +	Color ^
<input type="checkbox"/> Multiplayer Only +	<input type="checkbox"/> Required Motion Controls +	<input checked="" type="checkbox"/> Customize HUD Color +
Auditory ^	<input type="checkbox"/> Mouse Rebinding +	<input checked="" type="checkbox"/> Colorblind Filters +
<input type="checkbox"/> Volume Controls +		
<input type="checkbox"/> Subtitles +		

Figura 1.1 Característiques d'accessibilitat de Accessible Games Data Base. Font: accessiblegamesdatabase.com.

Tot i que aquesta eina reforça la idea de la importància de l'accessibilitat en els darrers anys, les opcions per daltonisme semblen limitades a 3 caselles on dues es centren en aplicar filtres (ajustament de contrast i filtres per daltònics) i una opció de recolorització d'elements de la interfície d'usuari.

- En relació als filtres per daltònics, tot i ser una opció que es sol poder activar en les opcions del joc, en molts casos s'utilitza de forma inadequada, on enlloc de ressaltar els colors per les persones que pateixen de daltonisme el filtre torna els colors del joc com si el veiés una persona daltònica, produint l'efecte contrari (Figura 1.2).

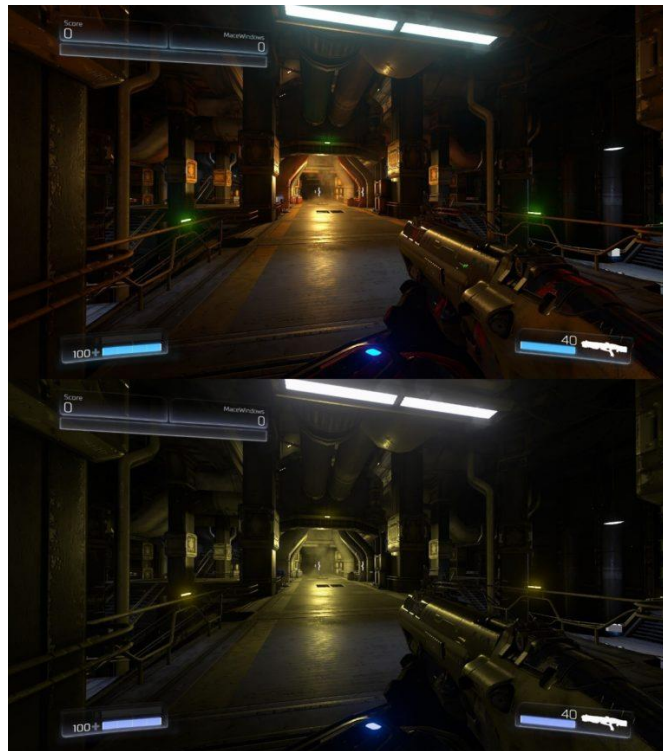


Figura 1.2 Filtre per daltònics en Doom. Font: iD Software.

- Les opcions de contrast son difícils d'analitzar si s'aprofiten correctament, ja que la majoria de jocs tenen aquesta característica lligada al paràmetre de la brillantor de la pantalla. Tot i així hi ha opcions per generar contrast més complexes, com l'analitzada en l'apartat 2.1 The Last of Us Part II.
- La recolorització dels elements de la interfície resta en multiplicar els elements del HUD per el color desitjat o directament canviar la textura. Aquest apartat no sol tenir una funció directa en jugabilitat dins de la simulació.

Les opcions que es solen donar tenen més relació amb un post processament de la

imatge final sense tenir en compte el disseny del joc en si. Jocs on el color és important per avançar en els nivells tenen més complicació al aplicar aquests filtres comentats.

Finalment, observant els jocs mostrats en aquesta plataforma demostren que no existeixen eines unificades per la seva aplicació directa. Cada estudi de desenvolupament implementa la seva versió d'una eina d'accessibilitat visual, però no sembla existir-ne una que reuneixi totes les característiques possibles per donar-li al jugador i menys encara que possibilitin dissenyar el joc tenint en compte problemes que no es puguin solucionar amb aquestes opcions que es plantegen.

De trobar una solució a aquesta problemàtica, neix la iniciativa d'aquest projecte.

2 Antecedents de la recerca/Referents

Per dissenyar l'eina, junt amb el marc teòric, s'han analitzat diferents maneres d'implementar mètodes que utilitzen jocs comercials. Per altra banda també s'analitzen conceptes clau relacionats amb una experiència final, tant per l'usuari final com pel desenvolupador.

2.1 The Last of Us Part II

Essent el guanyador del premi Innovation and Accessibility dels The Game Awards l'any del seu llançament, 2020, The Last of Us Part II (Naughty Dog, 2020) és un dels jocs més complets en opcions d'accessibilitat (Campbell, 2020) no centrades només en l'apartat visual, sinó també en opcions per problemes d'oïda, personalització de subtítols, narrador de pantalla... entre d'altres.

Els elements centrats en l'apartat visuals són els següents:

- Color del HUD: Diferents colors predefinitos per les diferents tipologies de daltonisme que tornen a colorejar els elements de la interfície extra diegètica.
- High contrast display: Aquesta opció (Figura 2.1) ofereix una versió del joc que elimina qualsevol color dels shaders originals per ser colorejat segons un codi de colors personalitzable perquè els elements siguin més fàcils de reconèixer en pantalla i els usuaris amb daltonisme puguin personalitzar aquests colors com necessitin, segons les deficiències en els cons fotoreceptors (Gunther, Neitz i Neitz, 2006). Per arribar al resultat sembla ser que el mode consta de:
 - Dibuixat del buffer de profunditat. En una escala de grisos es mostra quins elements estan més a prop de la càmera virtual que controla el jugador i quins estan més lluny, de blanc a negre respectivament.
 - Ús de contorns en els elements del joc: el joc mostra contorns blancs en els elements de la jugabilitat com l'avatar, companys, enemics, objectes de l'entorn, i també en els elements més propers a la càmera virtual per contrastar-los amb el dibuixat del buffer de profunditat.
 - Codi de colors dels elements en la jugabilitat: per contrastar l'escala de grisos del buffer de profunditat el joc acolorix els diferents tipus d'elements segons un índex de diferents colors personalitzables.



Figura 2.1 Mostra de l'opció High Contrast Display dins la jugabilitat. Font: Naughty Dog.

2.2 Chroma Gun i Color ADD

A partir d'un gameplay penjat a la plataforma Youtube on una persona daltònica intentava jugar al joc de puzzles en primera persona Chroma Gun (Pixel Maniacs, 2015), l'equip de desenvolupament va decidir implementar un sistema per representar els colors amb símbols en una actualització del joc (Pixel Maniacs, 2016), on la deficiència de visualització colors no afecta a com s'interpreta aquest signe. Aquest codi parteix de l'existent ColorADD, documentat en l'apartat 3.2.1.2.

L'ús que se li dóna a aquest codi de colors consta principalment de 2 elements en el joc (Figura 2.2):

- Una imatge per sobre del material que renderitza la geometria indicant el color del mateix, és a dir, informació diegètica localitzada en l'espai tridimensional
- Icones en el HUD, de manera extra diegètica. Per una banda trobem a la cantonada inferior esquerra el color seleccionat diferenciat dels altres per la disposició en pantalla, on es veu un quadre més gran. Finalment tenim una icona a sobre de la pistola de color que indica el mateix color seleccionat, però seguint la posició de la pistola de forma bidimensional.

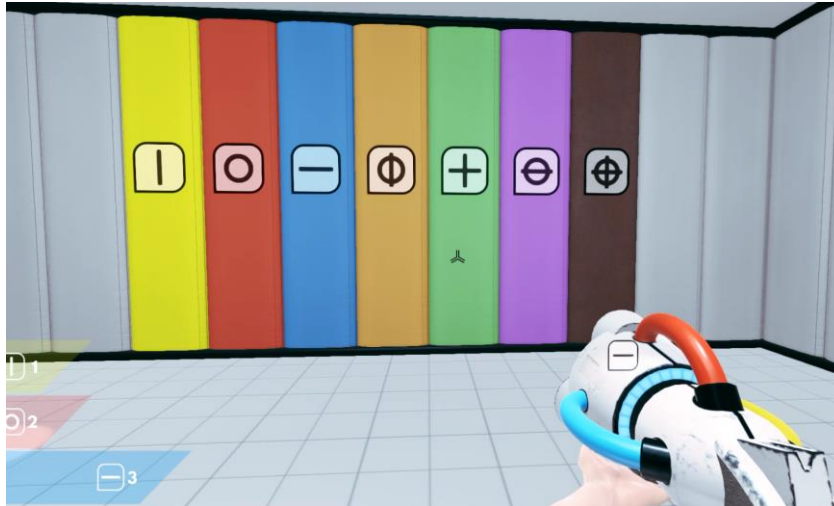


Figura 2.2 Ús del sistema ColorADD en ChromaGun. Font: Pixel Maniacs.

2.3 Unity Progrids

Per pensar en experiència d'usuari pel desenvolupador del joc i per veure quina tipologia d'elements es poden afegir a la interfície d'usuari en la finestra Scene View de Unity, on es permet editar l'entorn tridimensional, trobem que l'eina Progrids implementa un HUD amb botons de fàcil accés com un menú desplegable, mostrats en la següent figura (Figura 2.3) a la cantonada superior esquerra.

Aquests botons pensats pel desenvolupament de l'eina van pensats a activar o desactivar diferents modes com:

- Filtres de les quatre tipologies de daltonisme (Protanòpsia, Deuteranòpsia, Tritanòpsia i acromatòpsia)
- Mode Debug on es mostrin amb contorns possibles problemes de contrast.

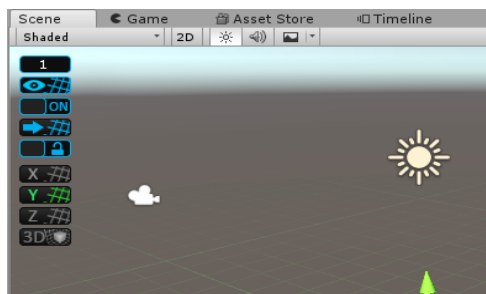


Figura 2.3 Interfície d'usuari de l'eina Progrids en el motor de joc Unity. Font: Progrids.

2.4 Celeste

El joc de plataformes Celeste (Extremely OK Games, 2019) ofereix el mode de joc Assist Mode on la velocitat de tot el joc es pot alentir percentualment per reduir la capacitat de reacció necessària per superar els obstacles, afegir més salts abans de tornar a tocar el terra... Els desenvolupadors informen que el joc ha estat dissenyat sense tenir en compte aquest mode (Figura 2.4), és a dir, que l'experiència d'usuari més significativa o vertadera és la que no fa ús d'aquestes opcions.

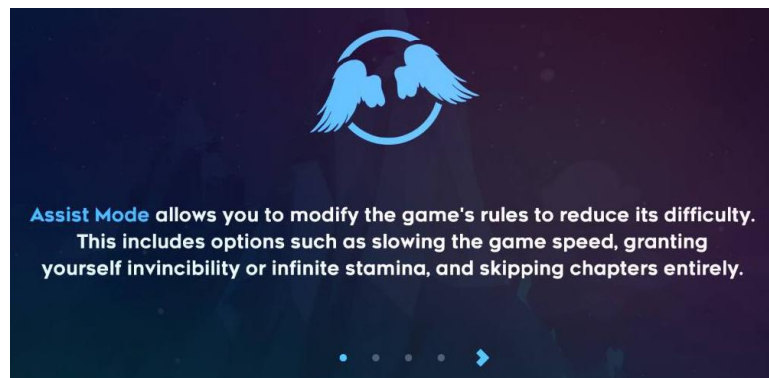


Figura 2.4 Text dels desenvolupadors de Celeste informant als jugadors d'una experiència d'usuari diferent a la dissenyada. Font: Extremely OK Games.

S'ha escollit aquest joc com a referent per el concepte d'oferir al jugador final una versió del joc amb una experiència d'usuari diferent, per complementar-ho amb un disseny de diverses experiències semblants utilitzant metodologies documentades en l'apartat 3.2.2.2 Active Game Accessibility i l'apartat 3.2.2.3 Game Accessibility Guidelines.

3 Marc teòric

La documentació per realitzar l'eina s'ha estructurat en analitzar el daltonisme com a discapacitat, les solucions que s'han trobat tant per al món dels videojocs com a la vida quotidiana, seguit d'apartats que recullen fonaments, altres eines i tècniques per implementar-la.

3.1 Discapacitats Visuals

El terme discapacitat agrupa deterioraments del cos humà, limitacions en la seva activitat per executar tasques i restriccions en la seva participació, és a dir, com s'involucra en una situació de la vida de l'individu, on els factors del entorn també interactuen amb aquests components. La Organització Mundial de la salut proposa el model The International Classification of Functioning, Disability and Health per diferenciar una discapacitat del funcionament entès com el conjunt de funcions, activitats i participació del cos de l'individu (OMS, 2002).

Hi ha diferents tipus de discapacitats segons la part del cos on la funcionalitat es veu afectada, però per la focalització del projecte la recerca es centra en discapacitats visuals, concretament en el tracte del color.

3.1.1 Daltonisme

El daltonisme o visió de color es tracta d'una discapacitat relacionada amb l'habilitat de diferenciar variàncies de color. La majoria de casos de daltonisme son congènits i més predominants en els homes per la seva relació amb el cromosoma X (OMS, 2001).

- ▼ Sensory functions and pain
 - ▼ Seeing and related functions
 - ▼ b210 Seeing functions
 - ▶ b2100 Visual acuity functions
 - b2101 Visual field functions
 - ▼ b2102 Quality of vision
 - b21020 Light sensitivity
 - b21021 Colour vision
 - b21022 Contrast sensitivity
 - b21023 Visual picture quality

Figura 3.1 Classificació de daltonisme com a discapacitat. Font: icd.who.int.

A banda de recents tractaments de teràpies genètiques que de moment només han estat provades en rates i primats no existeix cap manera efectiva de revertir el daltonisme pels éssers humans (Chidlow, Chan, Wood i Casson, 2022).

3.1.1.1 Funcionalitat de l'ull humà

Per entendre la font aquesta discapacitat s'ha d'entendre el comportament de la llum i la seva interacció amb la vista. Per la següent explicació es parteix de la teoria de fotons, entenent la llum com un conjunt de partícules elementals plenes d'energia que es mouen per l'espai (Lewis, 1926).

La teoria de com percebem el color parteix de com la llum arriba als nostres ulls i es reflecteix i refracta. La idea de percebre els colors com una freqüència ve de la part d'aquest procés on la llum, ja refractada pel seu pas per la còrnia i l'iris, passa pel forat de la pupil·la. La llum que entra pel forat té una direcció i es troba més dispersa a les parets del forat per on passa. Fins que arriba al nervi, passant per una segona refracció en el cristal·lí, hi ha un temps de retard entre el recorregut de la llum que és reflectida per aquesta paret i la llum que passa sense reflectir. Finalment, arriba al nervi, constituït en tres porcions sensibles a diferents freqüències (Young, 1801).

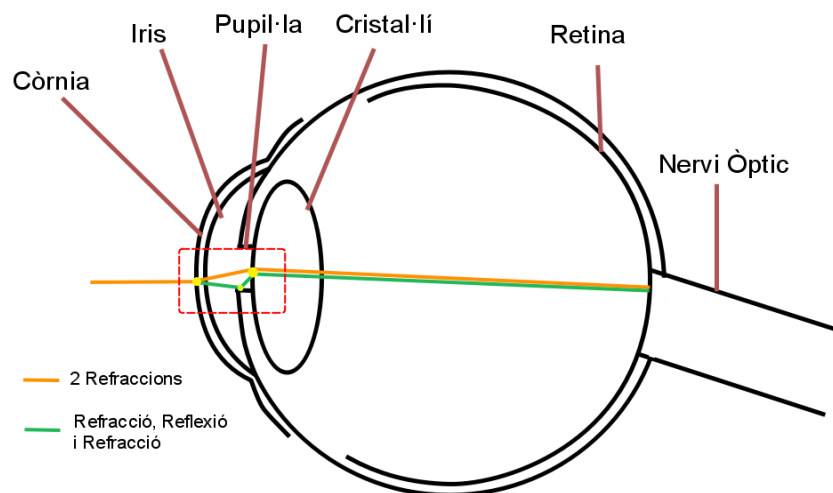


Figura 3.2 Diagrama del recorregut dels fotons en l'ull humà. Font: Elaboració Pròpia.

La hipòtesi de Young (1801) planteja aquestes freqüències amb una correlació directa amb els colors primaris considerats vermell, blau i groc. És una hipòtesi amb la que contrasta una teoria més actual del que coneixem com a cons fotoreceptors que absorbeixen els fotons (Glauber, 1963). Els cons són sensibles a diferents rangs de freqüències, corresponent als cons S, M i L per amplituds d'ona considerades petites, mitjanes o grans, respectivament (Figura 3.3), que engloben el espectre de colors dels primats (Roorda i Neitz, 2000).

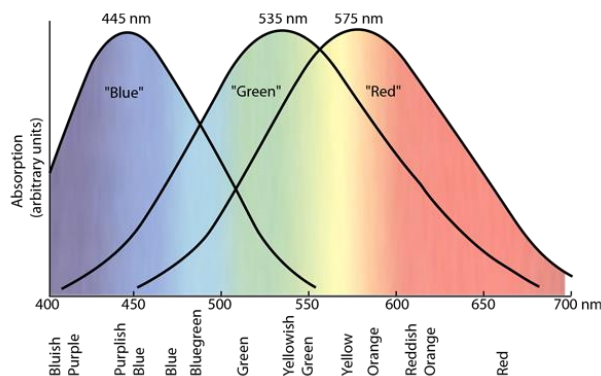


Figura 3.3 Diagrama de rangs de freqüències dels cons S, M i L respectivament.
Font: Hyperphysics.phy-astr.gsu.edu.

La representació dels colors en el sistema nerviós de l'ésser humà es redueix als tres nombres corresponents a la interpretació de la llum dels tres tipus de cons, i aquesta habilitat s'anomena tricromàcia (Lennie, 2000). Com que cada con fotoreceptor només obté la informació de la freqüència amb la que s'absorbeix la llum aquesta tricromàcia de la vista obté la informació del color per comparació dels valors de cada tipus de con (Simunovic, 2009).

Els tipus cons estan irregularment distribuïts (Figura 3.4) on els cons S o blaus representen un 10%, mentre que els cons M i L, verds i vermells respectivament formen una població similar (Lennie, 2000).

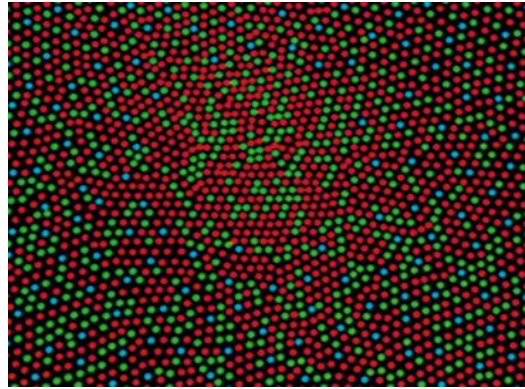


Figura 3.4 Distribució dels tipus cons en el nervi òptic sense cap deficiència. Font: phaidon.com.

Tot i aquesta comparació dels 3 tipus de cons fotoreceptors, en l'ull humà trobem un altre element que contribueix al color percebut i que la seva quantitat distribuïda en la retina és del 95% respecte el 5% dels cons del nervi òptic dels que parla Young (1805), els bastons fotoreceptors (Lamb, 2016). Aquest element, precursor dels cons en l'evolució dels vertebrats, té la funció de proporcionar visió escotòpica, és a dir, és més sensible a la foscor (Warrant, 2015). Com menciona Lamb (2016), l'explicació ve donada per les condicions de llum i supervivència en la vida dels ancestres de l'ésser humà.

Tant els cons com els bastons han evolucionat fins avui dia, proporcionant tant la visió escotòpica com la visió fotòpica o la visió dels colors en condicions diürnes (Warrant, 2015).

3.1.1.2 Deteriorament de l'ull

El daltonisme ve donat per un deteriorament de l'ull humà: és el resultat d'una mutació genètica que afecta a l'expressió de l'espectre complet de colors que els cons fotoreceptors del nervi ocular són capaços d'absorbir (Simunovic, 2009). La mutació genètica que produeix aquest deteriorament funciona de tal manera que reorganitza els gens que codifiquen les opsines, que són les molècules sensibles a la llum que la filtren i modifiquen l'espectre sensible del con fotoreceptor (Zaccardi, 2006) a les membranes de les cèl·lules dels diferents tipus L, M i S (Gunther, Neitz i Neitz, 2006).

3.1.1.3 Tipologies de Daltonisme

Les diferents tipologies de daltonisme es classifiquen principalment per severitat i posteriorment pel tipus de con fotoreceptor afectat. Segons Simunovic (2006) els graus de severitat es classifiquen de la següent manera, de més lleu a més sever:

- Anomalia tricromàtica: la comparació dels valors dels tres cons que utilitza el sistema visual pateix una aberració que accepta valors de colors que una comparació sense aquesta discapacitat no validaria. És la forma de daltonisme menys greu, però aquesta aberració pot variar des de apropar-se a una visió normal a apropar-se al resultat de la següent tipologia, la dicromàcia.
- Dicromàcia: la comparació dels valors dels cons es redueix a la de dos dels tres cons existents per validar el color percebut. La font del problema sembla ser o bé un reemplaçament dels cons d'un tipus per altres o una pèrdua d'un dels tipus de cons.
- Monocromàcia: La distinció dels colors és nul·la a causa d'una greu disfuncionalitat dels cons fotoreceptors i l'únic color percebut és a causa dels bastons que s'utilitzen per la visió nocturna en una retina sense aquest deteriorament.

Posteriorment a la severitat, la tipologia s'aprofunditza en el tipus de con o dels cons afectats. La nomenclatura de cada color segons la seva severitat és la següent:

- Anomalia tricromàtica
 - Protanomalia: anomalia en el con vermell
 - Deuteranomalia: anomalia en el con verd
 - Tritanomalia: anomalia en el con blau
- Dicromàcia
 - Protanòpsia: pèrdua o reemplaçament del con vermell
 - Deuteranòpsia: pèrdua o reemplaçament del con verd
 - Tritanòpsia: pèrdua o reemplaçament del con blau
- Monocromàcia
 - Monocromàcia del con verd
 - Monocromàcia del con vermell
 - Monocromàcia del con blau

- Acromatòpsia: els 3 cons són afectats i el resultat de la visió és donat pels bastons fotoreceptors.

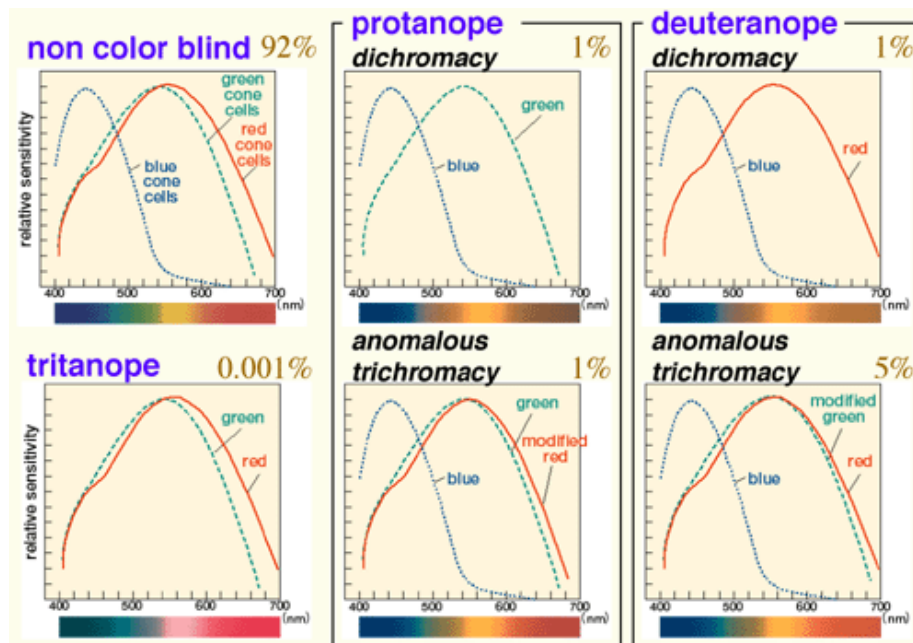


Figura 3.5 Representació dels defectes dels cons i espectre de colors de les tipologies de daltonisme principals. Font: Color Universal Design (CUD).

Finalment, com recopilen Gunther, Neitz i Neitz (2006) les tipologies més comuns són les que tenen a veure amb els colors vermell i verd, o amb els cons L i M (Roorda i Neitz, 2000) que afecten a 1 de cada 12 caucàsics i 1 de cada 230 dones, mentre que les deficiències del con S o del color blau són menys comuns. Aquest fet es contrasta amb el que explica Lennie (2000) sobre la població de cada tipus de con, on el con S o blau és el que es troba en menor quantitat.

3.2 Solucions per a discapacitats visuals

3.2.1 Solucions fora del món dels videojocs

3.2.1.1 Simulació de color

Una primera aproximació per poder dissenyar en base a individus amb diferents tipologies de daltonisme es simular la visió d'aquests. Podem englobar les simulacions en anomalies tricromàtiques junt amb dicromàcies per la seva similitud i les monocromàcies per altre banda.

3.2.1.1.1 Pèrdues parcials o totals dels cons

Per poder aplicar un filtre per simular el tipus d'anomalia, primer s'ha d'obtenir el color en un espai de color que es pugui representar numèricament en un ordinador, en un dels espais de color formals com: RGB format pels colors primaris vermell, verd i blau; CMYK per cian, magenta, groc i negre; i HSV amb tonalitat, saturació i lluminositat (Noor, Mokhtar, Rafiqul i Pramod, 2012).

El següent pas per arribar a la simulació és transformar el color codificat a l'espai de color LMS, per entendre el color com excitacions relatives als tres tipus de cons de l'ull humà indicats en el mateix nom. Aquesta transformació es pot realitzar a través d'una matriu resultant de la modificació colorimètrica de Judd-Vos per arribar a valors tristímuls i la modificació de Smith i Pokorny per arribar als valors LMS (Viénot, Brettel i Mollon, 1999).

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = T_{RGB-LMS} \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 17.8824 & 43.5161 & 4.1194 \\ 3.4557 & 27.1554 & 3.8671 \\ 0.0300 & 0.1843 & 1.4671 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Figura 3.6 Matriu de transformació de color d'espai RGB a LMS. Font: Stack Overflow.

El valor en l'espai LMS es pot modificar linealment per reduir l'espectre de colors d'una persona amb dicromàcia. Segons el tipus la matriu necessària varia segons el con

afectat: una matriu identitat on en la fila de la qual s'anul·la el valor de la identitat i les altres dues dimensions son reemplaçades pels valors dels 3 cons resolent la fórmula del pla $\alpha L + \beta M + \gamma S = 0$ on el con afectat té un valor de 0 (Figura 3.7) i els components $\alpha\beta\gamma$ corresponen als valors de contrast obtinguts d'operacions amb els colors blanc i blau transformats a espai de color LMS (Anbari, Hamidi i Kermanshahani, 2021) mostrades en la següent figura:

$$\alpha = M_W S_B - M_B S_W$$

$$\beta = S_W L_B - S_B L_W$$

$$\gamma = L_W M_B - L_B M_W.$$

Figura 3.7 Fórmules de les components $\alpha\beta\gamma$ de l'equació del pla, on W és la respectiva component del color blanc en l'espai LMS i B és la respectiva component del color blau. Font: Viénot, Brettel i Mollon.

Amb les constants $\alpha\beta\gamma$ obtingudes, resolent l'equació del pla pels diferents tipus de dicromàcies es poden generar les següents matrius de transformació lineal variant la fila de la matriu identitat 3x3 segons el tipus de con afectat (Figura 3.8).

Amb el color transformat linealment es pot aplicar el filtre transformant-lo d'espai LMS a RGM amb la inversa de la matriu utilitzada anteriorment (Figura 3.7) (Viénot, Brettel i Mollon, 1999).

Com que la transformació és lineal, es simula l'anomalia interpolant linealment entre el color original i la dicromàcia on la pèrdua o reemplaçament del con afectat és total.

$$\alpha L + \beta M + \gamma S = 0 \quad \text{LMS White} = \begin{bmatrix} 65.65783 \\ 34.57819 \\ 1.671355 \end{bmatrix} \quad \text{LMS Blue} = \begin{bmatrix} 4.11935 \\ 3.86715 \\ 1.46709 \end{bmatrix}$$

protanopia (pèrdua del con L) => resoldre L

$$L = - \frac{(\beta M + \gamma S)}{\alpha}$$

$$L = - \frac{((S_{white} * L_{blue} - S_{blue} * L_{white}) * M + (L_{white} * M_{blue} - L_{blue} * M_{white}) * S)}{(M_{white} * S_{blue} - M_{blue} * S_{white})}$$

$$L = 2.02344 * M - 2.52581 * S$$

construcció de matriu de transformació

$$T = \begin{array}{ccc|c} & L & M & S \\ \hline & 1 & 0 & 0 & \text{Protanopia (L)} \\ & 0 & 1 & 0 & \text{Deuteranopia (M)} \\ & 0 & 0 & 1 & \text{Tritanopia (S)} \end{array}$$

$$\begin{array}{l} T = \\ \text{Protanopia} \end{array} \begin{array}{ccc|c} 0 & 2.02344 & -2.52581 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \end{array}$$

Figura 3.8 Procés de generació de matrius de transformació lineal pel cas de Protanòpsia. Font: Elaboració pròpia.

3.2.1.1.2 Acromatòpsia

Per simular l'acromatòpsia resultant d'una visió utilitzant només els bastons fotoreceptors enlloc dels cons s'ha de conèixer el seu comportament. de transformar la imatge en blanc i negre, concretament obtenint el valor de luminància. La visió nocturna o escotòpica que menciona Simunovic (2009) de la que s'encarreguen els bastons de la retina es produeix a través de la codificació de la senyal luminància (Oesch i Diamond, 2011).

Mentre que la mitjana dels canals de l'espai de color RGB és la intensitat de la llum d'una llum elèctrica, una suma ponderada de $0.299 * \text{Canal Vermell} + 0.587 * \text{Canal Verd} + 0.114 * \text{Canal Blau}$ retorna la luminància d'una escena (Pierre, Aujol, Bugeau, i Ta, 2015).

3.2.1.2 ColorADD

ColorADD és un sistema universal d'identificació gràfica de colors dissenyat identificant les dificultats dels individus amb daltonisme per crear un vocabulari visual fàcil de comprendre i memoritzar (Neiva, 2017).

El Sistema proposat per Neiva (2017) es basa en un sistema additiu de 3 formes gràfiques (Figura 3.9), cadascuna representant un dels colors vermell, blau i groc, i de forma estesa a sobre d'una figura quadrada negra o blanca esbossada de color negre per representar colors més clars o més foscos respectivament.

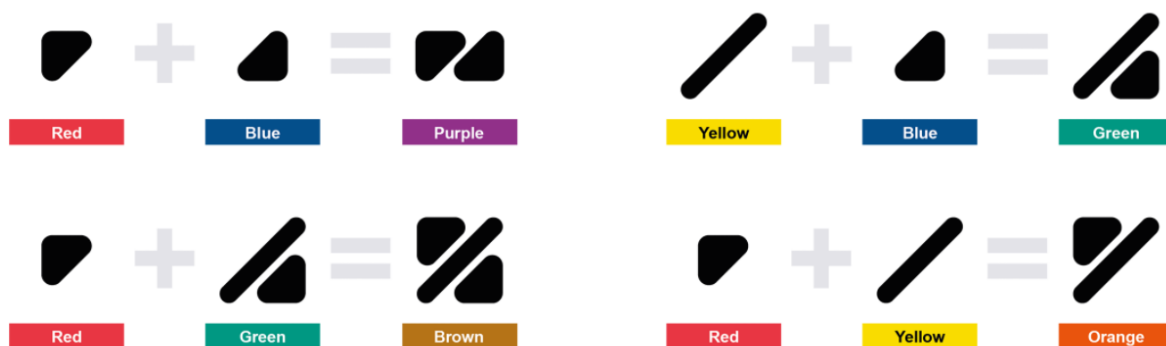


Figura 3.9 Exemple d'addició de figures del sistema ColorADD. Font: coloradd.net.

Aquest sistema utilitza un model de color personalitzat format pels colors vermell, blau i groc amb un paràmetre per enfosquir o aclarir, i difereix amb la consistència establerta en models de colors formals: RGB, CMYK i HSV (Noor, Mokhtar, Rafiqul i Pramod, 2012).

Aquest sistema crea un nombre de possibilitats extens (Figura 3.10) i fàcil de comprimir per explicar la seva funcionalitat (Neiva, 2017). Aquest sistema ha estat provat en escoles, activitats esportives, hospitals, transport públic, roba i altres sectors mostrant un potencial beneficiós per un grup que representa un percentatge significatiu de la població.

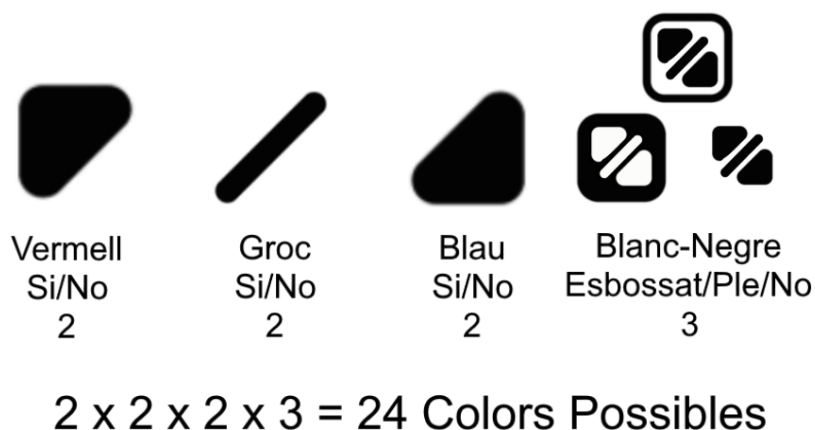


Figura 3.10 Combinacions possibles del sistema ColorADD. Font: Elaboració pròpia.

3.2.1.3 Ulleres per daltònics

Tot i no haver una manera efectiva de tractar el deteriorament de l'ull pel cas del daltonisme, existeixen eines físiques per ajudar amb les persones amb aquesta discapacitat. Una de les opcions per ajudar al dia a dia dels afectats de forma passiva són filtres de colors o lents tintades (Figura 3.11) que recentment han generat interès en les xarxes. Aquestes ulleres es venen sota la marca EnChroma, i amb el filtre que apliquen s'encarreguen de bloquejar part de l'espectre de colors que es troben en els rajos de llum que arriben al nostre ull (Gómez-Robledo, Valero, Huertas, Martínez-Domingo i Hernández-Andrés, 2018).



Figura 3.11 Lents de les ulleres EnChroma per diferents tipologies de daltonisme. Font: eu.enchroma.com.

Tot i que la promesa de la companyia és oferir una ajuda òptica que ajudi a realçar els colors en la vida dels daltònics, recerques com la de Almutairi, Kundart, Muthuramalingam, Hayes i Citek (2017) sostenen que mentre poden aplicar un contrast entre un objecte situat a força profunditat i un objecte més a prop a causa dels artefactes que generen aquests filtres, els resultats no produeixen un efecte significatiu en els subjectes del respectiu experiment. Aquesta informació es contrasta amb una declaració que es mostra en la pàgina web de EnChroma, on exposen que les ulleres poden no funcionar per una tipologia severa dels colors vermell i verd, i l'experiment de Gómez-Robledo, Valero, Huertas, Martínez-Domingo i Hernández-Andrés (2018) acaba desxifrant que aquestes ulleres no tenen cap efecte en persones amb dicromàcia (apartat 3.1.1.3) sinó que serveixen per contrastar els colors en persones amb anomalia tricromàtica.

3.2.1.4 Smart glasses per daltònics

Una altra proposta per ajudar a les persones en daltonisme es basa en les conegudes smart glasses. La tecnologia que ofereixen les smart glasses es basen en interfícies com Heads-Up Display, Head-Mounted Display o Optical Head-Mounted Display que permeten a l'usuari veure a la vegada el món físic i un món digital que pot aportar informació (Lausegger, Spitzer i Ebner, 2017). És aquí on recau el potencial d'aquestes ulleres, en utilitzar Realitat Augmentada per ajudar amb persones amb discapacitats (Tanuwidjaja, Huynh, Koa, Nguyen, Shao, Torbett i Wibel, 2014).

L'ús que se li ha donat a aquesta peça de tecnologia per ajudar als daltònics han estat propostes d'aplicacions que s'encarreguen de forma activa de contrastar els colors que puguin crear una dificultat a l'hora d'identificar-los en activitats diàries (Figura 3.12). Exemples d'aplicacions son: Chroma de Tanuwidjaja, Huynh, Koa, Nguyen, Shao, Torbett i Wibel (2014) i OmniColor de Lausegger, Spitzer i Ebner (2017).

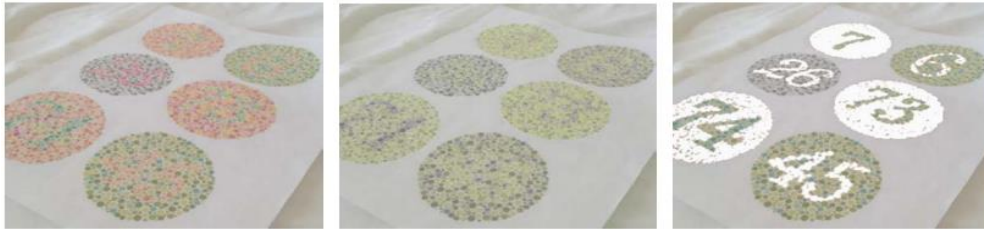


Figura 3.12 Tests de daltonisme en 3 casos: Visió Normal, Dicromàcia-Deuteranopia i aplicació Chroma, respectivament. Font: UbiComp '14 Adjunct.

Tot i que el desenvolupament de les funcionalitats de smart glasses com les conegudes Google Glass segueix creixen gràcies a una gran comunitat (Lausegger, Spitzer i Ebner, 2017) les ulleres generen controvèrsia principalment pel tòpic de la privacitat ja que les ulleres permeten a l'usuari fer fotografies i gravar vídeos amb la seva petita càmera que també utilitza per funcions de realitat augmentada. Ha estat un problema on hi ha hagut agents reguladors i governs de per mig, fet que va resultar en que el producte fos un fracàs comercial al nivell del mercat massiu, resultant també en aturades de aplicacions en desenvolupament per aquestes ulleres (Klein, Sørensen, Freitas, Pedron i Elaluf-Calderwood, 2020). L'estat més recent és que el producte ha estat catalogat com a eina d'ús per empreses, el que deslliga el producte de trobar solucions com les aplicacions esmentades per al gran públic.

3.2.2 Solucions aplicades als videojocs

3.2.2.1 Breu Historia de l'accessibilitat en els videojocs

Com bé expliquen Westin, Hamilton i Ellis (2020) sembla ser que la importància en l'accessibilitat dins dels videojocs té la seva arrel en la dècada de 1970 on usuaris d'ordinadors jugaven a jocs basats en text gràcies a un hardware que permetia donar ordres de forma tàctil junt amb un escàner portàtil per poder llegir el propi text (Figura 3.13).



Figura 3.13 Optatron: escàner de text portàtil per persones cegues. Font: Youtube.

Els mateixos autors estenen que el següent pas per l'accessibilitat en els videojocs arriba a la dècada de 1980, on novel·les comencen a reflectir les barreres que poder crear els videojocs per televisors als usuaris d'aquest tipus de producte posant d'exemple l'alta velocitat de la pilota del joc Pong (Atari, 1972) per a discapacitats amb paràlisi cerebral o síndrome de Down. A més, és en la mateixa dècada on apareixen micròfons amb sintetitzadors que permeten a les persones cegues jugar a un joc de principi a final a través d'ordres de veu, junt amb els sorolls del propi joc i guies d'àudio.

Tot i els avenços donats, no és fins la dècada dels 2000 que els jocs que donaren suport a diferents capacitats eren considerats més seriosos (Miesenberger, Ossmann, Archambault, Searle i Holsinger, 2008). De fet, els avenços durant aquesta dècada son significatius de forma accidental, amb un suport per persones cegues gràcies al traductor de text a veu de sistemes operatius com iOS o la inclusió de subtítols en els jocs del gènere aventura Full Motion Video (Westin, Hamilton i Ellis, 2020).

Aquests fets demostren que tot i haver un interès per desenvolupar tenint en compte a persones discapacitades van enfocats cap al deteriorament total de la funcionalitat afectada: ceguesa en el cas de la vista i sordesa en el cas de l'oïda. Aquestes consideracions contrasten amb els avenços més actuals que tenen en compte deterioraments que no arriben a anul·lar un sentit. És aquí on els videojocs comencen a aplicar solucions per jugadors que no poden diferenciar senyals d'àudio, jugadors amb problemes motors, com per exemple el Xbox Adaptive Controller permetent una gran personalització per englobar la majoria de discapacitats motores (Godineau, 2018), arribant a discapacitats visuals basades en la capacitat de distinció de colors com el daltonisme (Garber, 2013).

Finalment, l'accessibilitat dels videojocs es segueix desenvolupant a dia d'avui, i cada cop surten més iniciatives que ho demostren, com la base de dades mencionada en el capítol introductori, els fluxos de treball i directrius que s'expliquen en els següents subapartats... Un exemple important és el requeriment per part de la Comissió de Comunicacions Federals dels estats units (FCC) per que la comunicació dintre dels videojocs, com els xats de text i xats de veu siguin accessibles (McGuire, 2019). Aquesta demanda també és important per la distinció entre accessibilitat, usabilitat:

- Accessibilitat: mesurada en el rendiment d'assolir objectius.
- Usabilitat: assolida si individus amb discapacitats poden tenir accés a la totalitat del producte.

3.2.2.2 Active Game Accessibility Framework

Quan es parla d'opcions d'accessibilitat en videojocs normalment es parteix de principis d'accessibilitat web o principis disseny universal del model Human Computer Interaction o HCI per a interfícies d'usuari (Miesenberger, Ossmann, Archambault, Searle i Holsinge, 2008) on es té la finalitat de que el producte pugui ser utilitzat pel major nombre de persones possible (Dix, Finlay, Abowd i Beale, 2004). Aquesta focalització en aconseguir que persones amb discapacitats puguin arribar a utilitzar el producte resulta menys eficaç en els videojocs que en la resta de tipologies de software (Pontin, de Lima, de Souza, Agostini i Nogueira, 2017) si no s'aprofunditza, ja que es parteix d'un model interactiu que requereix per part del jugador, a més de realitzar un comportament, executar-lo amb certa habilitat, i és aquí on trobem que un

gran nombre de tipologies de discapacitats posen una barrera al disseny del joc: cada tipus de discapacitat pot afectar de diferent manera a l'execució del comportament esperat.

Per entendre com dissenyar segons diferents tipologies de discapacitats la metodologia de desenvolupament Active Games Accessibility (Miesenberger, Ossmann, Archambault, Searle i Holsinge, 2008). Aquest sistema de treball parteix de dos principis:

- Fer que els desenvolupadors aprenguin les necessitats de les persones amb les discapacitats indicades per poder dissenyar tenint-ho amb compte.
- Permetre als desenvolupadors crear característiques que els usuaris finals puguin activar per jugar al joc.

Tot i que la metodologia s'orienta principalment a les interfícies d'usuari no deixa de banda la configuració proactiva dels elements del joc per adaptar-ho als usuaris finals. Com es planteja en el "Assist mode" de Celeste (2018) l'enfocament del disseny i configuració dels elements poden estar adaptat posteriorment per facilitar que més persones puguin arribar a jugar al joc, amb la compensació de perdre l'experiència original.

La configuració de joc de la qual parla el Active Game Accessibility Framework convergeix amb aquesta idea d'experiència significativa. També s'observa el punt de vista de treballar la dificultat des del desenvolupament del producte per fer-lo més accessible. Aquestes idees de dificultat i jugabilitat dissenyada estan relacionades amb l'experiència d'usuari, i aquesta normalment s'avalua en conjunt amb l'accessibilitat del joc (Pontin, de Lima, de Souza, Agostini i Nogueira, 2017).

Per reforçar aquesta idea en relació a la temàtica de la recerca s'ha de tenir en compte els colors del joc, ja que aquests junt amb el conjunt les mecàniques, objectes del nivell... contribueixen al model conceptual dels jugadors sobre la jugabilitat (Aslam i Brown, 2020) tant com que pot capturar l'atenció d'aquests i guiar la seva presa de decisions.

Exemples de color com element fonamental de la jugabilitat es poden trobar en jocs de plataformes com Uncharted 4 (Naughty Dog, 2019) on els elements interactuables en l'escalada són ressaltats en un color diferent al propi entorn (Figura 3.14) i ajuden al jugador a crear el seu camí a seguir, de forma que se li dona un objectiu i un propòsit, de forma que l'usuari té més interès en arribar a llocs amb un camí que ha estat planificat (Woods, Rudd, Robertson i Davids, 2020).



Figura 3.14 Elements escalables ressaltats en color blanc en Uncharted 4. Font: Game Career Guide.

El color també té un rol molt important en el disseny d'interfícies, on s'ha revelat que el blau i el taronja son els colors preferits de la població (Napoli i Chiasson, 2018), colors relacionats directament amb el les tipologies de daltonisme que afecten als cons S i M respectivament. Amb aquest fet, entrant en psicologia del color per persones amb daltonisme, les emocions provocades pels colors venen per dues parts no exclusives, fons biològic i un fons de l'entorn on es troba l'individu (Elliot i Maier, 2014). És en aquest entorn d'aprenentatge on els videojocs contribuir a formar associacions entre colors i emocions.

3.2.2.3 Game Accessibility Guidelines

La creació del set de directrius denominades Game Accessibility Guidelines neix d'una evolució dels estàndards d'accessibilitat en videojocs plantejats per la BBC i d'una manca de focalització en videojocs del Web Content Accessibility Guidelines 2.0 de la World Wide Web Consortium (Westin, Ku, Dupire i Hamilton, 2018).

Un altre enfocament que justifica la necessitat d'un nou grup de normes específiques pels videojocs parteix del propòsit de l'aplicació, on l'accessibilitat en videojocs es

diferència de l'accessibilitat del software en general per l'objectiu que té el programa d'entretenir a l'usuari. Mentre que l'Active Game Accessibility de l'apartat anterior es una metodologia de desenvolupament, les guidelines serveixen com un forma popular de valorar l'accessibilitat del joc (Pontin, de Lima, de Souza, Agostini i Nogueira, 2017) com mostra la Figura 3.15.

Les directrius proposades pels autors es classifiquen tant per dificultat d'implementació com per tipologia de discapacitat. Els motius de la classificació resten en un equilibri de quanta gent pot satisfer, el seu impacte i el seu cost de desenvolupament dintre de la producció del joc (Westin, Ku, Dupire i Hamilton, 2018).

	A	B	C	D
1	Guideline	Relevant to mechanic?	Implemented?	Notes
2		(no" automatically greys out row)		
3	Motor			
4	(Control / mobility)			
5				
6	Basic			
7	Allow controls to be remapped / reconfigured			
8	Ensure controls are as simple as possible, or provide a simpler alternative			
9	Ensure that all areas of the user interface can be accessed using the same input method as the gameplay			
10	Include an option to adjust the sensitivity of controls			
11	Ensure interactive elements / virtual controls are large and well spaced, particularly on small or touch screens			
12				
13	Intermediate			
14	Support more than one input device			
15	Make interactive elements that require accuracy (eg. cursor/touch controlled menu options) stationary			
	Ensure that multiple simultaneous actions (eg. click/drag or swipe) are			

Figura 3.15 Formulari d'Excel de Game Accessibility Guidelines. Font: Game Accessibility Guidelines.

Tot i que existen aquestes directrius, també hi ha d'altres com les proposades per la International Game Developers Association o IGDA o per Medialt. Finalment per arribar a avaluacions més específiques es recorre al les proves tradicionals amb varis usuaris (Pontin, de Lima, de Souza, Agostini i Nogueira, 2017).

3.3 Motor de Joc: Unity

3.3.1 Què és un motor de joc?

Un motor de joc és el conjunt de components de software que permeten crear un joc, des de la inclusió d'assets visuals a la lògica del món virtual i les seves mecàniques. Normalment es compon d'una suite d'eines i un component en runtime (Gregory, 2018).

El terme de motor de joc neix en la dècada de 1990 amb videojocs llançats al mercat pensats pel modding, amb una personalització lliure a través de llenguatges de programació.

3.3.2 Breu història de la creació de Unity

El popular motor de jocs neix al 2002 d'una conversa entre dos programadors, Nicholas Francis i Joachim Ante, en un fòrum de desenvolupament on un d'ells buscava una manera d'implementar un sistema de shaders per el motor de joc que estava desenvolupant (Haas, 2014). De la conversa va resultar una col·laboració entre els dos i poc després es va afegir un tercer soci, David Helgason, el qual es torna el CEO de la companyia amb la qual es funda el projecte, Over the Edge Entertainment.

Per fer popular el motor, trobar errors, es va desenvolupar un joc amb ell llançat el 2005 abans de llançar l'eina. La intenció darrere el desenvolupament va ser crear una eina enfocada als problemes dels creadors de videojocs. El motor es va acabar llançant el juny del mateix any en la World Wide Apple Inc's Conference, exclusivament per Mac OS X (Hussain, Shakeel, Hussain, Uddin i Ghouri, 2020). La compatibilitat amb més plataformes, com Windows Vista, es va donar amb el llançament de la versió 2.0 al 2007, i per això va ser necessari donar suport a Microsoft DirectX (Haas, 2014).

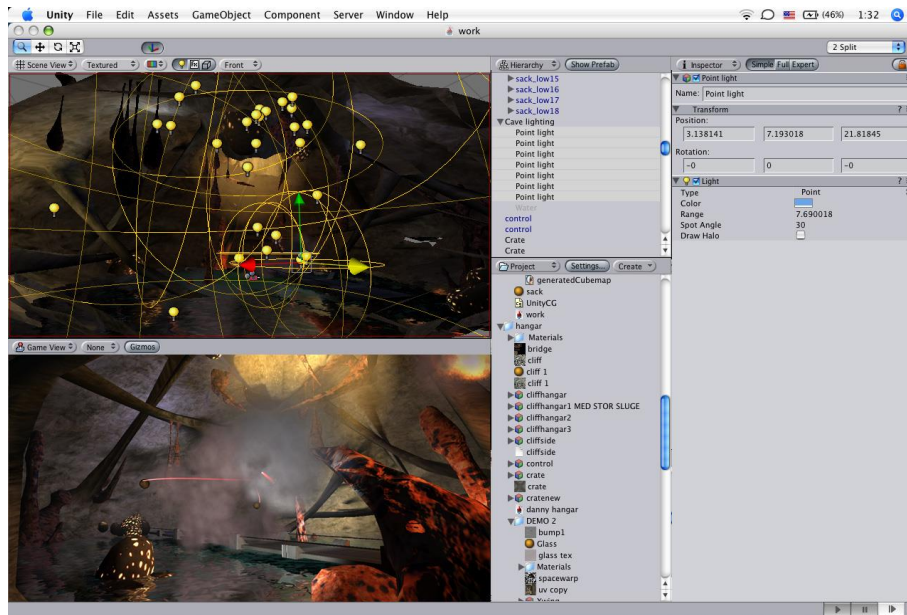


Figura 3.16 Unity Versió 1.0 “Next-Gen” Demo. Font: forum.unity.com.

El nombre de plataformes compatibles ha augmentat amb els anys, tant en que al 2010, amb el llançament de la versió 3.0 es va tornar el motor de joc més popular a nivell d'ensenyament i el més utilitzat en plataformes mòbils.

Tot i haver-se desenvolupat a principis de la dècada dels 2000, el panorama de la creació i publicació de motors de desenvolupament al públic prové de pocs anys enrere, gràcies al llançament del joc Unreal (Epic Games, 1998) el qual oferia, entre grans avenços tècnics respecte els jocs de l'època, un editor de nivells i un nucli del motor de forma extensible en el llenguatge C++ (Herwig i Paar, 2002).

Unity ha estat utilitzat per altres indústries fora dels videojocs com el cine, arquitectura, automoció, construcció i altres enginyeries (Hussain, Shakeel, Hussain, Uddin i Ghouri, 2020).

3.3.3 Unity Asset Store

A finals de 2010, l'empresa responsable del motor de joc anuncia i publica una botiga online anomenada Unity Asset Store dissenyada per ser inclosa dins del mateix ecosistema de desenvolupament per una ràpida implementació dels arxius publicats (Haas, 2014).

Tot i que les dades de Haas (2014) afirmen que existeixen desenvolupadors que poden viure només de vendre paquets d'arxius en aquesta plataforma, actualment és més difícil i controversial ((Hussain, Shakeel, Hussain, Uddin i Ghouri, 2020) per l'increment de jocs del tipus Asset Flip, jocs creats pràcticament en la seva totalitat d'assets (Packard, i Phillips, 2019).

3.4 Creació d'eines de desenvolupament

Per millorar l'experiència de desenvolupament dins d'un software a vegades és necessari l'existència d'alguna eina per automatitzar processos i millorar la metodologia de treball (Tadres, 2015).

Com menciona Pratama (2018) en un exemple de desenvolupament d'aplicacions de realitat virtual, la producció de contingut gràcies a les eines publicades es torna més fàcil, però tot i així la majoria d'eines son especialitzades i deixen poc marge de funcionalitat, sent molt grans i difícils de personalitzar per un ús específic (Blow, 2004). Per aquest fet algunes empreses decideixen crear les seves pròpies eines de desenvolupament.

Finalment, a banda de millorar un procés de creació, l'absència d'eines del motor de joc que forma part del model d'estudi de videojocs pot arribar a influenciar la capacitat d'innovació dels treballadors a causa de les limitacions tècniques en aquest procés (Schmalz, 2015).

Per motius de desenvolupament de la part pràctica d'aquest projecte, aquest apartat s'aprofunditza en el tipus de software escollit, Unity com a motor de joc.

3.4.1 APIs

Una Interfície de programació d'aplicacions o API és un programa que exposa serveis o informació d'un software oferint recursos que poden utilitzar altres aplicacions sense haver de treballar de manera subjacent en el software d'on es volen utilitzar les funcionalitats. D'aquesta manera, les APIs faciliten tasques de programació de forma modular per reutilitzar codi (Meng, Steinhardt i Schubert, 2017).

3.4.2 Desenvolupament d'eines en el motor de jocs Unity

Recentment hi ha una tendència de fer que els motors de jocs siguin lleugers en termes de característiques que inclouen que porten perquè després l'usuari pugui afegir diferents extensions o plugins per expandir les capacitats del motor. Fins i tot motors comercials com Unreal Engine o Unity donen suport a aquesta opció d'afegir plugins (Park i Baek, 2020).

Quan un projecte creix en complexitat o en quantitat, més tasques involucren repetició i més possibilitat d'error, i això és perquè l'eina original, en aquest cas el motor de joc Unity, no satisfà totes les necessitats (Tadres, 2015).

Els desenvolupadors de Unity faciliten una API sobre l'editor de scripting per modificar el comportament de l'editor de Unity mitjançant events, millorar la metodologia amb una GUI personalitzada i automatitzar tasques repetitives accedint a les funcionalitats principals de l'editor (Tadres, 2015).

3.4.2.1 Carpetes Especials

Mentre que a l'hora de afegir nous arxius o assets en un projecte Unity deixa incloure-ls a qualsevol carpeta, el motor es reserva el nom de varies carpetes per diferents propòsits. En cas de desenvolupar eines, la carpeta "Editor" s'encarrega d'afegir funcionalitats per l'editor que no es trobaran en les builds compilades (Unity, 2021).

3.4.2.2 Editor Scripting API

L'Editor Scripting API de Unity junt amb el coneixement d'ús del Unity Package Manager afavoreix el desenvolupament amb eines de forma modular.

La flexibilitat que ofereix Unity amb aquesta API per crear eines en C# comunicant-se amb altres parts del motor permet crear punts d'inflexió en la forma de desenvolupar dins del motor. Grans exemples que aprofiten aquesta API són les mateixes eines opcionals de Unity Timeline, aprofitant la Immediate Mode GUI i manipulació de dades de Scriptable Objects; o Shader Graph, utilitzant la part de UI Toolkit i generació de codi de shader (Kok, 2021).

Per utilitzar les funcionalitats d'aquesta API d'editor s'ha d'utilitzar l'assembly UnityEditor, i no pot ser referenciada per codi en runtime compilat pels jugadors finals (Unity, 2021). Aquest assembly dona accés a les classes que implementen funcionalitats de desenvolupament o perquè nous scripts heretin de classes preparades per sobreescriure les funcionalitats base.

3.4.2.3 Atributs

Plataformes de desenvolupament com .NET donen suport a atributs, els quals son anotacions entre claudàtors associades a elements d'un programa, com classes i membres. Aquests atributs s'encarreguen de customitzar la metadata per al seu fàcil accés en runtime (Ghezzi i Monga, 2002).

Unity, a banda dels atributs definits per les llibreries .NET, també ofereix una serie d'atributs en referència a la seva API, tant per la part de motor com la de l'editor (Unity, 2021), per exemple canviar la forma de com es dibuixen les propietats en la finestra inspector (Kok, 2021).

```
[HideInInspector]  
public float strength;
```

Figura 3.17 Mostra d'un atribut d'editor per amagar una variable pública en l'inspector del motor. Font: Unity.

3.4.3 Disseny i Documentació de d'eines

Una eina desenvolupada amb una API passa a ser una extensió de les eines que ofereix la mateixa interfície. Quan s'ofereix una documentació d'aquesta extensió els desenvolupadors esperen que aquesta estigui integrada en l'entorn de desenvolupament (Meng, Steinhardt i Schubert, 2017).

A raó de que una eina o extensió es desenvolupa amb una API com a referència es poden seguir un set d'heurístiques de disseny i documentació com les que recull Bloch (2006):

- Una API hauria de ser fàcil d'utilitzar i difícil d'utilitzar malament.
- Una API hauria de documentar-se per si mateixa.
- Per començar a dissenyar primer cal recollir els requisits.
- Esbossar la documentació de la API de forma curta, normalment una pàgina.
- El codi mostrat ha de ser exemplar.
- Evitar límits que puguin generar obsolescència.
- Importància de nomenclatures per a una bona consistència.
- Rere el dubte, descartar una funcionalitat.

- No sobre explicar la implementació del codi.
- Documentar cada element de la API exportada.
- Minimitzar l'accessibilitat d'encapsulació.

Amb aquestes directrius s'apunta a aconseguir una bona API, amb l'objectiu de que el seu ús no impliqui la creació de codi addicional per la seva funcionalitat esperada (Henning, 2009).

3.5 Programació gràfica

3.5.1 Introducció a la programació gràfica

En el món dels ordinadors, el procés de crear una imatge es coneix com a renderització. En el cas dels videojocs, aquesta imatge es construeix a través de models 2D o 3D i informació d'una escena com la posició i orientació de la càmera o les llums situades en un nivell (Halladay, 2019).

3.5.1.1 Geometria

Els models 2D o 3D defineixen la informació que l'ordinador traduirà en formes en pantalla. La informació necessària per definir una forma es compon de primitives: vèrtexs, com a punts en l'espai 3D; vores, com a línies que ajunten 2 vèrtexs; i cares, formades per 3 o més vores.

Els vèrtexs, tot i representar punts en l'espai 3D local també poden consistir en propietats addicionals, com una propietat de color, la seva direcció de la normal, coordenades de textura i el seu pesat per al sistema d'animacions (Luna, 2006).

Per codificar els vèrtexs s'utilitza un vertex buffer, que defineix una llista amb les propietats dels vèrtexs. Per codificar les cares s'utilitza un index buffer, que defineix una llista amb cadenes de 3 índexs de vèrtex formant una cara.

3.5.1.2 Càmera virtual

La càmera d'un joc especifica quina part del món es pot veure per renderitzar la respectiva imatge en 2D. Per renderitzar els models del món en perspectiva, el volum de la càmera ha de ser un frustum definit pels plans de contenció i un angle de visió (Luna, 2006).

3.5.1.3 Procés de Renderització

El procés de renderització conforma les operacions necessàries per transformar la informació de l'espai en una imatge en dues dimensions. Com recull Luna (2006), aquestes operacions s'han d'encarregar de transformar les coordenades locals dels vèrtexs a coordenades de projecció de la càmera virtual. Amb les transformacions donades s'aplica el procés conegut com clipping per descartar la informació

innecessària fora del volum que conté la càmera. Al final del procés de renderització es rasteritza la imatge final, es a dir es construeix la imatge amb els píxels obtinguts.

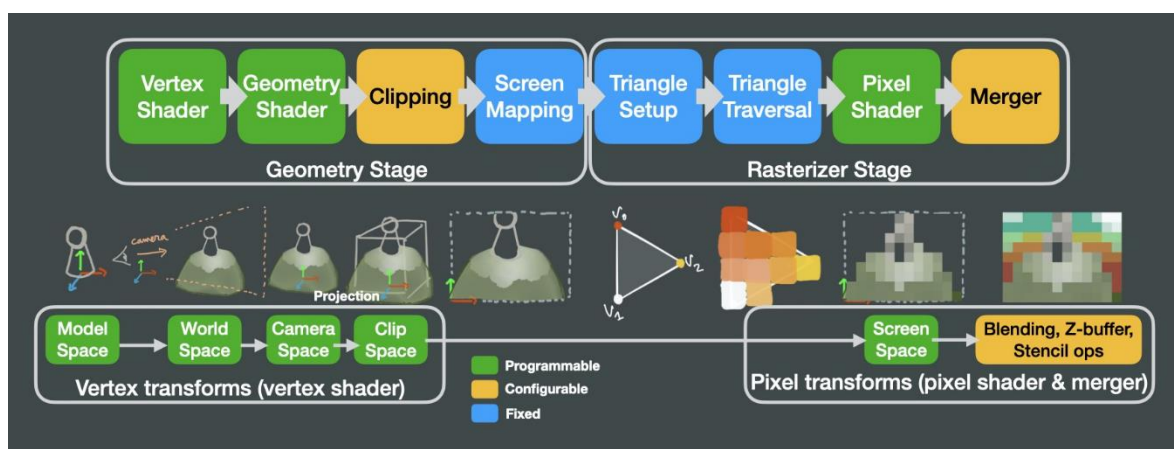


Figura 3.18 Etapes del procés de renderització. Font: tech.metall.com.

3.5.1.4 Shaders

Un shader és un codi que defineix les operacions matemàtiques de materials per a la seva il·luminació en l'escena en el procés de renderització (He, Foley, Hofstee i Fatahaliam, 2017). El codi resultant dona informació i instruccions a la GPU per realitzar aquesta tasca enlloc de la CPU ja que aquesta no és capaç de realitzar aquests processos amb una freqüència suficient.

Halladay (2019) defineix els principals tipus de shaders com:

- **Vertex shader:** el primer shader en el procés de renderització i l'encarregat de realitzar les transformacions per portar coordenades locals a coordenades de projecció, mitjançant les matrius de model, vista i projecció.
- **Fragment shader:** definint un fragment com a estructura de les dades necessàries per dibuixar un píxel en pantalla, el fragment shader s'encarrega de generar la informació de color del píxel resultant. Les dades d'aquesta estructura arriben interpolades respecte els extrems que corresponen als vèrtexs del model 3D.

Existeixen altres tipus de shaders que es poden aplicar al procés de renderització, com el tessellation shader, per augmentar o reduir la densitat de cares del model 3D (Tariq, 2009); o el geometry shader, encarregat de generar, modificar o descartar geometries primitives (Mukundan, 2022).

3.5.2 Tècniques d'implementació

Hi ha un gran nombre de tècniques utilitzades per arribar infinits resultat. Per les dimensions d'aquesta recerca s'expliquen les tècniques rellevants per el desenvolupament de l'eina com a objectiu.

3.5.2.1 Efectes de Post-processing

Un efecte de post-processament és un efecte visual aplicat a una cada píxel d'una imatge renderitzada (Doppioslash, 2017). La tècnica utilitzada per copiar tots el píxels d'una imatge original a una imatge com a destinació s'anomena Blit. Amb aquesta imatge copiada es pot crear l'efecte visual manipulant-la.

Els efectes de post-processing es troben freqüentment com a part de cada motor de videojocs, i poden arribar a aconseguir fer la imatge més realista (Figura 3.19) com amb efectes de bloom, tone mapping, lens flare... o accentuar una estilització marcant les vores dels objectes, imitant efectes de còmics... fins a arribar a ajudar a la part mecànica del joc ressaltant zones o objectes en colors més contrastats (Magdics, Sauvaget, García i Sbert, 2013)

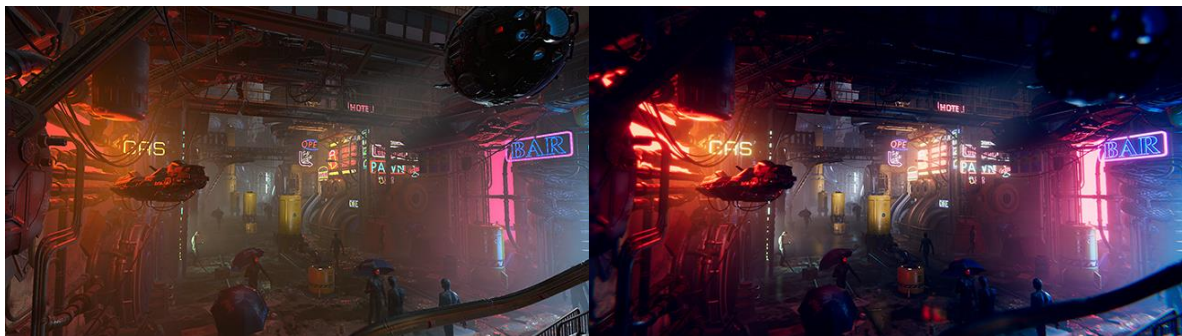


Figura 3.19 Escena sense post-processat (esquerra) comparada amb post-processat (dreta). Font: Unity.

3.5.2.2 Depth Buffer

El Depth Buffer o buffer de profunditat és un dels algorismes de computació de visibilitat (Greene, Kass i Miller, 1993) que s'encarrega d'emmagatzemar en una imatge el valor de profunditat que correspon al respectiu píxel dibuixat en pantalla (Lapidous i Jiao, 1999). Aquesta profunditat funciona de forma efectiva per geometries opaques mentre que no tracta bé les transparències.

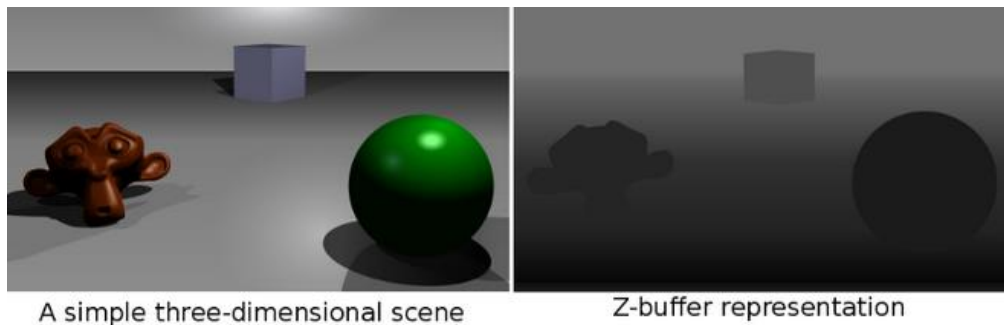


Figura 3.20 Representació del buffer de profunditat (dreta) respecte a una escena (esquerra). Font: GitHub.

Al funcionar per geometries opaques, el seu ús principal en aquests casos és aplicar un test de profunditat per renderitzar de forma ordenada amb un algorisme que descarta els píxels o fragments amb més profunditat que la dibuixada en el píxel del buffer (Theoharis, Papaioannou i Karabassi, 2001). Aquest buffer es pot utilitzar per aplicar altres funcionalitats, com la generació d'ombres a través de shadow mapping o trobar una aproximació al tractament de transparències a través de depth peeling (Everitt, 2001).

Finalment, el valor dibuixat en el depth buffer correspon a la divisió de les components z i w de la posició en coordenades de projecció o perspectiva (Persson, 2007).

3.5.2.3 Detecció de vores

Una de les tècniques utilitzades per millorar la llegibilitat d'una escena és la creació de vores sobre els objectes. Aquest procediment es sol utilitzar quan l'objectiu estètic d'un projecte no és el fotorealisme (Card i Mitchell, 2002).

Per crear aquest element visual existeixen diferents tècniques, basades en la creació de quads extrems que funcionin com a silueta, de forma que la geometria base ha de partir d'unes normals suavitzades perquè funcioni correctament (Figura 3.21), o en algorismes de post-processat d'imatge.

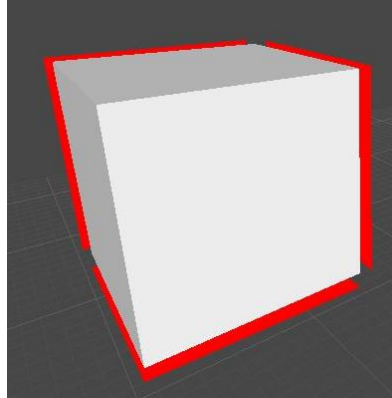


Figura 3.21 Problemes de silueta amb normals no suavitzades. Font: GameDevStackExchange.

El funcionament de la segona opció, els algorismes de post-processament d'imatge, es basa en la lectura i detecció de vores, definides com discontinuïtats en les propietats de l'espai físic, fotomètric i geomètric (Ziou i Tabbone, 1998). Tot i així hi ha diverses maneres de trobar aquestes vores (Figura 3.22), normalment agrupades en les metodologies de degradats o d'operacions lapacianes (Shrivakshan i Chandrasekar, 2012). Aquests dos es diferencien per la forma de validar la derivada de la imatge, segons si es compara el valor amb el màxim i el mínim trobat, el que implica una iteració constant, o es compara amb un valor t definit com a travessa zero.



Figura 3.22 Dues metodologies diferents de detecció de vores aplicades a una imatge original (esquerra). Font: TowardsDataScience.

Per trobar la derivada de la imatge existeixen diferents operadors. Una aproximació simple és l'operador Roberts Cross, on es sumen les diferències en els valors de les dues diagonals que envolten un punt o píxel en la pantalla.

4 Objectius

Els objectius del treball s'han estructurat com a principals, essent un requeriment essencial i secundaris, com apartats ideals però no fonamentals per un correcte funcionament.

4.1 Objectius Principals

L'objectiu central del projecte consta en realitzar una eina o plugin pel motor Unity per facilitar als desenvolupadors la creació del joc en funció dels usuaris que pateixin daltonisme.

Aquesta eina es planteja amb les següents característiques:

- Opcions pels desenvolupadors
 - Filtre de l'editor de l'escena amb les diferents tipologies de daltonisme.
- Opcions per l'usuari final (facilitar la implementació als desenvolupadors)
 - Opció de mode per daltònics: Shaders especials per afegir llenguatge gràfic de representació de colors
 - Opció per traduir el color del píxel on jugador estigui apuntant en una figura d'un codi visual de colors, en la interfície de joc sense interferir en el disseny artístic del projecte desenvolupat.
 - Opció de mode d'alt contrast inspirat en l'apartat 2.1.

4.2 Objectius Secundaris

Com a elements extres a la publicació de l'eina es plantegen els següents afegits:

- Documentació de l'eina.
- Mode Debug per trobar elements que puguin crear conflictes per les persones amb daltonisme.
- Creació d'una escena de mostra amb totes les funcionalitats de l'eina
- Creació d'una demo executable que funcioni com a tutorial per el sistema desenvolupat.
- Testejar i recollir feedback amb una mostra de gent amb discapacitat visual
- Expandir les funcionalitats de l'eina a més discapacitats visuals

5 Disseny metodològic i cronograma

Aquest capítol recull la planificació de cada pas realitzat en el treball d'investigació i implementació pràctica.

5.1 Fases del projecte

5.1.1 Investigació

Per a realitzar la investigació s'ha estructurat la informació segons els diferents focus dels quals parteix el projecte per poder realitzar un disseny i implementació de l'eina de manera de que el resultat sigui efectiu pels usuaris finals.

5.1.2 Documentació

Durant tot el transcurs del projecte es realitzarà la documentació a mode d'investigació i de memòria de desenvolupament de la seva respectiva part pràctica, incloent els següents apartats partint d'uns objectius marcats.

5.1.3 Cerca de referents

La cerca i selecció de referents pel projecte ha tingut en compte l'efectivitat i completesa de les implementacions en els propis projectes. Per una banda es troben referents amb característiques de solucions a problemes per la discapacitat que es volen replicar i per altre banda exemples relacionats amb l'experiència d'usuari, tant per la persona que faci ús de l'eina desenvolupada posteriorment o de l'usuari que jugui a un joc desenvolupat amb l'eina en qüestió.

5.1.4 Elecció d'eines i lliberies

L'elecció de les eines s'ha donat per la seva popularitat i facilitat d'ús. S'ha escollit Unity com a motor de joc on implementar l'eina i el llenguatge de programació C# per estar lligat amb el llenguatge d'implementació que utilitza el motor.

Per l'apartat de programació gràfica s'han escollit els llenguatges de shaders Shaderlab, com a forma de transformar paràmetres de l'inspector en variables de la targeta de vídeo, i HLSL per escriure el codi en qüestió.

5.1.5 Disseny i Desenvolupament de característiques

Basat en la investigació recopilada en el marc teòric i els referents analitzats es dissenyarà l'eina per a la seva implementació.

Els components clau del seu disseny venen del funcionament i deteriorament de la discapacitat investigada en l'apartat 3.1.1 Daltonisme seguint les metodologies Active Game Accessibility i Game Accessibility Guidelines dels apartats 3.2.2.2 i 3.2.2.3 respectivament. Els components analitzats en els referents també influeixen el disseny segons la seva efectivitat analitzada amb les metodologies anteriors.

Havent dissenyat prèviament les característiques desitjades, en aquest apartat s'implementen en el motor de joc escollit, Unity.

5.1.6 Desenvolupament d'una escena de prova

El desenvolupament de la escena de prova plantejada en el projecte inclou totes les característiques implementades per poder realitzar posteriorment una demostració de vídeo del seu comportament.

5.1.7 Desenvolupament d'una demo i testeig

Partint del propi disseny de l'eina, s'implementa tangencialment a l'eina un programa que funcioni de demo o tutorial per l'usuari final del videojoc desenvolupat perquè entengui el funcionament de forma curta. Aquest tutorial funciona com un recorregut lineal amb entrades de veu que ajudin a agilitzar el procés d'aprenentatge.

5.1.8 Publicació de l'eina

L'eina desenvolupada té la finalitat de ser publicada a la plataforma GitHub per les seves opcions de control de versions amb el codi font i oferir una descàrrega gratuïta.

5.2 Planificació

La següent taula (Taula 5.1) recull la planificació prèvia al desenvolupament del treball respecte les seves fases plantejades.

Fase	Novembre	Desembre	Gener	Febrer	Març	Abril	Maig	Juny
Documentació								
Anàlisi de Referents								
Investigació Sobre Daltonisme								
Investigació Sobre Solucions								
Investigació Sobre Implementació Tècnica								
Disseny de l'eina								
Desenvolupament de l'eina								
Testeig								
Conclusions								

Taula 5.1 Cronograma de les fases del projecte. Font: Elaboració Pròpia.

6 Desenvolupament de l'eina

L'eina s'ha dissenyat com el conjunt de 3 diferents implementacions en el desenvolupament d'un videojoc: filtres de simulació en l'editor de món, un mode alt contrast inspirat en l'anàlisi de l'apartat 2.1 i un conjunt de funcions que utilitzen un codi visual de colors inspirat en la documentació de l'apartat 3.2.1.2.

El mode alt contrast i el sistema del codi visual de colors són excloents entre si, així queden com diferents opcions per contribuir a la metodologia Active Game Accessibility aconseguint directrius de Game Accessibility Guidelines.

6.1 Filtres en la Scene View

La primera funcionalitat de l'eina desenvolupada és el conjunt de filtres que simulen les diferents tipologies de daltonisme, explicades en l'apartat 3.1.1.3, perquè un desenvolupador amb tricromàcia puguin detectar problemes de contrast d'una vista deteriorada pel daltonisme.

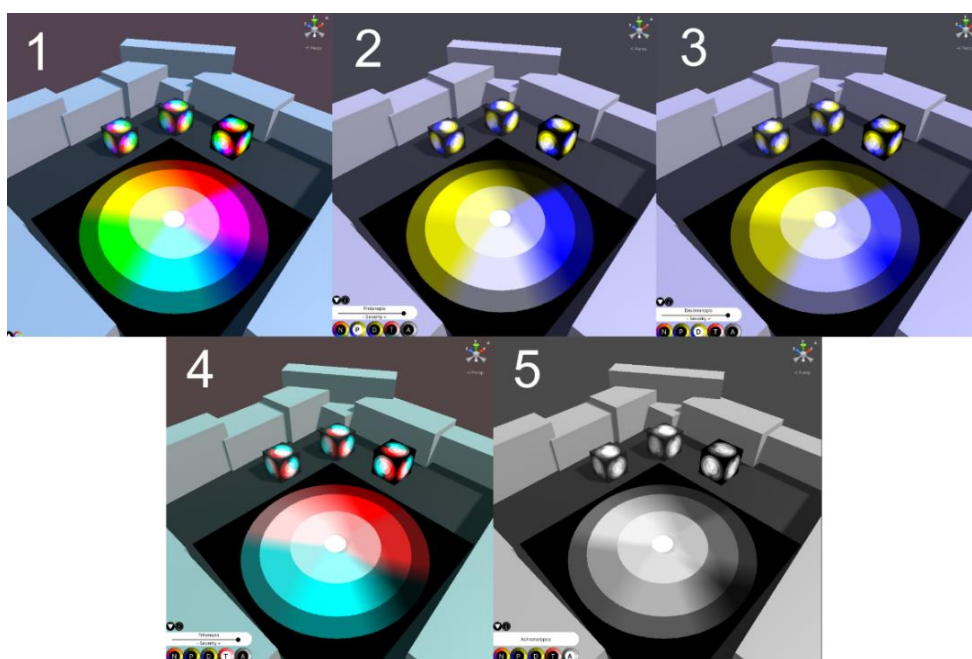


Figura 6.1 Filtres d'editor. 1-Tricromàcia, 2-Protanopia 3-Deuteranopia, 4-Tritanopia, 5-Acromatòpsia. Font: Elaboració Pròpia

L'aplicació dels filtres en la finestra Scene View de l'editor de Unity es compon de 3 passos:

- Creació del shader que redueix l'espai de color.
- Aplicació del shader com a post processat.

- Creació d'una eina que coordina una interfície d'usuari amb el material del shader anterior.

6.1.1 Shader de simulació de daltonisme

La simulació dels diferents tipus s'ha realitzat amb un únic shader, concretament del tipus uber-shader, el qual compila el codi segons unes paraules clau. Aquestes paraules clau corresponen a cada tipus de daltonisme.

El procés en el fragment shader consta de transformar el color en espai de color RGB en LMS, aplicar la reducció de l'espai amb una matriu i finalment tornar-lo a transformar en RGB per retornar-lo com el color del píxel final.

Per cada dicromàcia es defineix la matriu que redueix el espai de color amb els valors resultants de resoldre l'equació del pla.

```
float3x3 GetColorBlindnessLinearTransformation(){
#ifdef _CVTYPE_PROTANOPIA
    float3x3 transformationMatrix = { 0.0, 2.02344, -2.52581,
                                     0.0, 1.0, 0.0,
                                     0.0, 0.0, 1.0
    };
    return transformationMatrix;
#elif _CVTYPE_DEUTERANOPIA
    float3x3 transformationMatrix = { 1.0, 0.0, 0.0,
                                     0.494207, 0.0, 1.24827,
                                     0.0, 0.0, 1.0
    };
    return transformationMatrix;
#elif _CVTYPE_TRITANOPIA
    float3x3 transformationMatrix = { 1.0, 0.0, 0.0,
                                     0.0, 1.0, 0.0,
                                     -0.0122449, 0.07203435, 0.0
    };
    return transformationMatrix;
#else
    float3x3 transformationMatrix = { 1.0, 0.0, 0.0,
                                     0.0, 1.0, 0.0,
                                     0.0, 0.0, 1.0
    };
    return transformationMatrix;
#endif
}
```

Figura 6.2 Funció que retorna la matriu de reducció d'espai de color segons una paraula clau. Font: Elaboració pròpia.

Per simular les anomalies tricromàtiques enlloc de les dicromàcies, s'ha realitzat una interpolació lineal entre el color base i el color resultant de la dicromàcia. En el cas de la monocromàcia, el fragment shader retorna directament la luminància del color base.

```

float4 frag (interpolators i) : SV_Target {
    float4 col = tex2D(_MainTex, i.uv);

#ifdef CVTYPE_ACHROMATOPSIA
    float l_luminance = col.r * 0.299 + col.g * 0.587 + col.b * 0.114;
    return float4 (l_luminance.xxx, 1);
#endif

    float R = col.r;
    float G = col.g;
    float B = col.b;

    // Manual matrix multiplication
    float L = (17.8824 * R) + (43.5161 * G) + (4.11935 * B);
    float M = (3.45565 * R) + (27.1554 * G) + (3.86714 * B);
    float S = (0.0299566 * R) + (0.184309 * G) + (1.46709 * B);

    float3x3 anomalyTransformationMatrix = GetColorBlindnessLinearTransformation();

    float3 anomalyColorLMS = mul(anomalyTransformationMatrix, float3(L,M,S));

    // Manual matrix multiplication
    R = (0.0809444479 * anomalyColorLMS.x) + (-0.130504409 * anomalyColorLMS.y) + (0.116721066 * anomalyColorLMS.z);
    G = (-0.0102485335 * anomalyColorLMS.x) + (0.0540193266 * anomalyColorLMS.y) + (-0.113614708 * anomalyColorLMS.z);
    B = (-0.000365296938 * anomalyColorLMS.x) + (-0.00412161469 * anomalyColorLMS.y) + (0.093511405 * anomalyColorLMS.z);

    float3 anomalyColorRGB = lerp(col.rgb, float3(R,G,B), _Intensity);

    return float4(anomalyColorRGB, 1);
}
ENDCG

```

Figura 6.3 Fragment shader del shader final que aplica els filtres. Font: Elaboració pròpia.

6.1.2 Aplicació del shader a la Scene View

Per aplicar el filtre a la finestra d'escena, la classe que utilitza la funció `OnRenderImage` per cridar dins seu la funció `Graphics.Blit`, explicada en l'apartat 2.5.2.1 ha de tenir 2 atributs que modifiquin la seva metadata, un que permet executar el codi en mode d'edició i un que permet aplicar filtres post-processat en la càmera de la Scene View.

```

[ExecuteInEditMode]
[ImageEffectAllowedInSceneView]
public class ApplyFilter : MonoBehaviour {

```

Figura 6.4 Atributs de la classe que aplica el shader anterior. Font: Elaboració pròpia.

Aquest component, anomenat `ApplyFilter`, és creat per una classe que hereta de la classe `Editor`, també afegida en la carpeta especial del mateix nom i s'afegeix a la llista de components de la càmera principal de la escena. D'aquesta manera el component es destrueix i es torna a afegir a la càmera principal de la escena per ser l'últim filtre a aplicar cada cop que es l'script és compilat. Perquè no molesti al desenvolupador, quan es crea el component s'assigna el seu membre `hideFlags` a `hideInInspector`, de manera que no es visualitza en la llista de components de la càmera.

6.1.3 Interfície d'usuari en la Scene View

Per crear una bona experiència d'usuari, s'ha creat una interfície per alterar els paràmetres de tipologia i severitat d'anomalia ampliant el codi de la SceneView de Unity amb una funció pròpia. Aquesta funció utilitza diverses funcionalitats de la Editor Scripting API, mencionada en l'apartat 2.4.2 per crear funcionalitats de botons, sliders i la lògica que permet connectar-ho amb el material, carregat dins d'una de les carpetes especials, Resources. Amb les imatges carregades dins d'aquesta mateixa carpeta s'ha canviat l'aspecte dels botons per defecte que ofereix Unity amb les classes GUIContent i GUIStyle.

Per optimitzar l'espai o per si l'usuari final no vol veure ampliada l'eina en tot moment s'ha creat un botó que permet amagar o mostrar la selecció dels filtres.

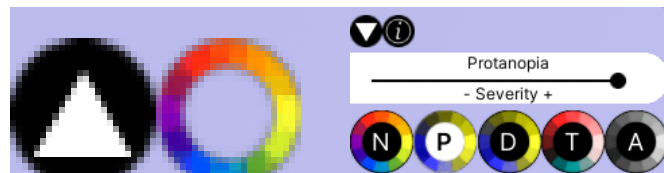


Figura 6.5 Funció retràctil de l'eina de filtres. Font: Elaboració pròpia.

Com a element de polit s'ha creat un botó d'informació extra que apareix si l'usuari deixa el ratolí a sobre durant uns instants, amb una funció Tooltip.

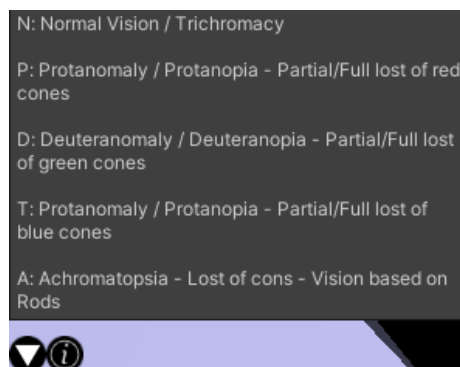


Figura 6.6 Tooltip de l'eina de filtres. Font: Elaboració Pròpia.

Per connectar l'eina amb la càmera de l'escena la lògica elimina i torna a afegir un component invisible de forma que sempre queda com l'últim filtre de post-processat a aplicar. Per tornar-lo invisible, s'ha utilitzat el paràmetre `hideFlags` del component que aplica el `Graphics.Blit`.

6.2 Mode d'Alt Contrast

La segona funcionalitat que implementa l'eina consta d'una rèplica del High Contrast Mode analitzat en l'apartat 2.1.

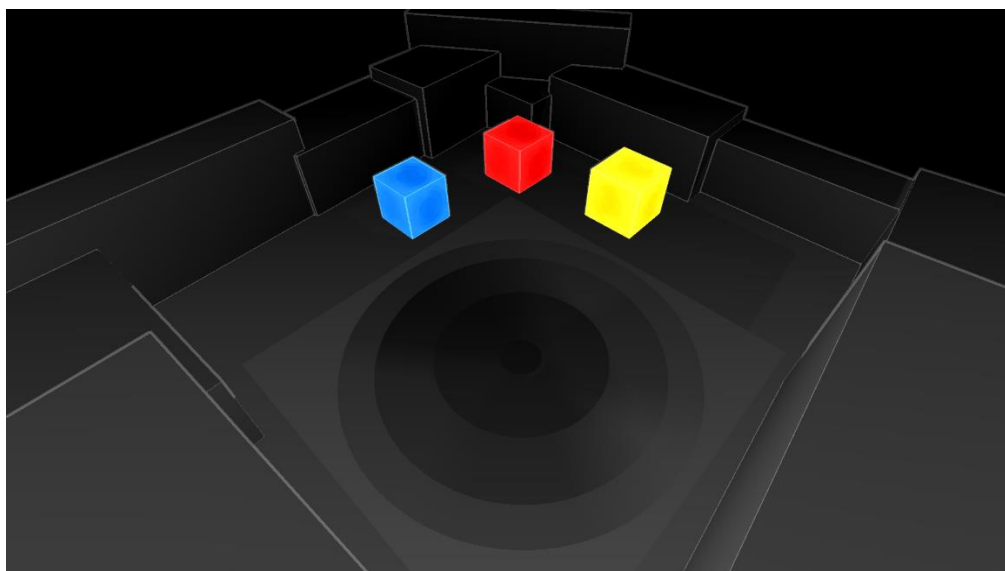


Figura 6.7 Mode d'alt contrast implementat en l'eina. Font: Elaboració pròpia.

6.2.1 Depth buffer

Per dibuixar el depth buffer amb valors que es puguin alterar per millorar el contrast d'un joc en específic s'ha realitzat un canvi de valors en aquest. Partint d'un valor entre 0 i 1 que realitza el gradient depenent dels valors Near i Far Plane de la càmera, s'ha transformat a un valor lineal, en unitats de Unity, per després realitzar un Inverse Lerp per aplicar els valors definits.

```
float CustomDepthValue(float4 screenPos){
    // Get Depth 01
    float l_depth = screenPos.z/screenPos.w;

    // Linear Distance Transform
    float l_linearDepth = LinearEyeDepth(l_depth);

    // Remapping
    return 1 - saturate((l_linearDepth-_HCZStart)/(_HCZEnd-_HCZStart));
}
```

Figura 6.8 Funció de canvi de valors del depth buffer. Font: Elaboració pròpia.

6.2.2 Creació de vores

Utilitzant la funció Roberts Cross de l'apartat 2.5.2.3 per calcular la derivada d'una imatge, s'ha realitzat la detecció de vores com el màxim resultant de les imatges que defineix la càmera de l'escena a través de la configuració de l'eina , profunditat i normals.

```
// Depth
float l_depthDifference0 = abs(l_depthTR - l_depthBL);
float l_depthDifference1 = abs(l_depthTL - l_depthBR);
float l_depthEdge = (l_depthDifference0 + l_depthDifference1) * 10 > _DepthThreshold;

// Normals
float3 l_normalDifference0 = l_normalsTR - l_normalsBL;
float3 l_normalDifference1 = l_normalsTL - l_normalsBR;
float l_normalEdge = (length(l_normalDifference0) + length(l_normalDifference1)) > _NormalThreshold;

float l_edge = saturate(l_depthEdge + l_normalEdge);
```

Figura 6.9 Càlcul de la vora a través de la profunditat i normals al voltant d'un píxel.
Font: Elaboració pròpia.

6.2.3 Implementació de shaders de contrast

Per arribar al resultat visual resultant s'ha proposat l'aplicació de shaders especials que interpolen linealment entre el color original i el color del mode d'alt contrast.

Per facilitar els shaders als usuaris finals l'eina ofereix 3 shaders d'exemple, un skybox, un standard surface, que utilitza el surface shader de Unity similar al material predeterminat, i un Unlit basat en vertex i fragment shader.

Les funcions que apliquen aquest aspecte visual s'engloben en un fitxer d'extensió .cginc perquè qualsevol usuari pugui aplicar-ho als seus shaders personalitzats. S'han descrit els passos a replicar per arribar a aquest resultat en la documentació (Annex 2).

6.2.4 Lògica del mode d'Alt Contrast

Per alterar els valors d'aquest mode segons les preferències de l'usuari final s'ha creat un component controlador que es pot incloure en un GameObject. Per activar o desactivar el mode, es pot modificar el membre m_highContrast del singleton que inclou el component.

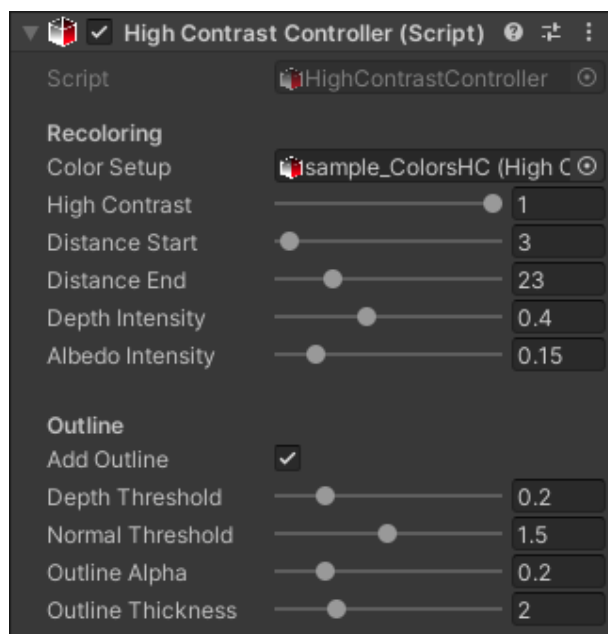


Figura 6.10 Controlador del mode Alt contrast. Font: elaboració pròpia.

Els paràmetres que poden variar son:

- Distance Start/End: distància en unitats de Unity per canviar els valors del degradat que forma el depth buffer.
- Depth Intensity: quantitat de color del depth buffer que s'aplica sobre el fons negre.
- Albedo Intensity: intensitat amb la que enfosqueix el color resultant segons la luminància del color base original.
- Outline: Activar o desactivar. S'ha proposat com paràmetre opcional per si el joc que l'usuari desenvolupa ja inclou un sistema de detecció de vores.
- Depth/Normal Threshold: a partir de quina quantitat de profunditat o de normal es detecta una vora.
- Outline Alpha i Thickness: aspecte final d'opacitat i mida de les vores pintades.

Un paràmetre que també pot variar és el Color Setup. És un paràmetre opcional ja que l'eina proposa un sistema de canvi de colors. Els shaders especials tenen un paràmetre de color que representen el seu aspecte quan el mode d'Alt contrast està activat. Si s'afegeix una configuració en aquest espai, la qual es crea amb un scriptable object per definir una llista de tags i colors que reescriuran el color d'alt contrast final.

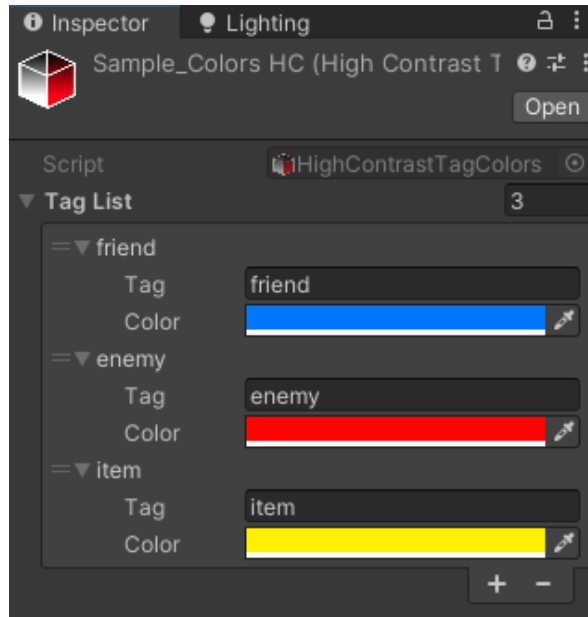


Figura 6.11 Scriptable object per definir Tags i Colors. Font: Elaboració pròpia.

Finalment, perquè tot aquest sistema funcioni, s'ha d'incloure un material que utilitzi un d'aquests shaders especials i un component creat, Set High Contrast Color By Tag, en el mateix objecte que dona informació de quin tag s'utilitza per reescriure el color.

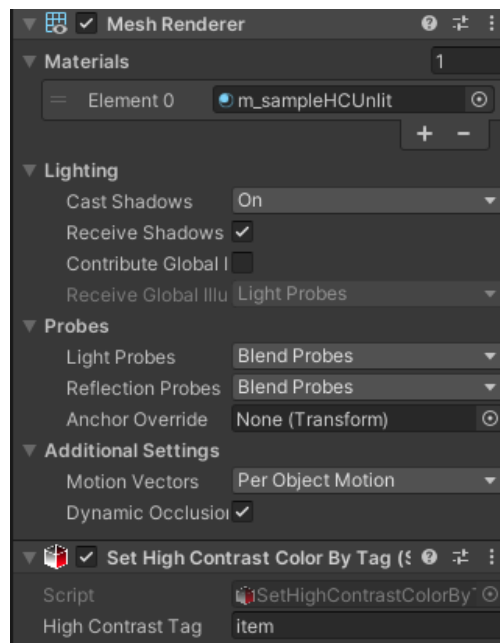


Figura 6.12 Inspector de l'objecte amb el seu Renderer i el component que reescriu el color en el mode d'Alt Contrast. Font: Elaboració Pròpia

6.3 Codi visual de colors

Tal i com s'ha explicat en l'apartat 2.2.1.2, ja existeix un codi visual per definir un color, l'anomenat ColorAdd. No obstant s'han detectat 2 problemes amb aquest sistema per aplicar-lo a un videojoc:

- Si el símbol es gira, complica la seva comprensió ja que dues de les seves figures son triangles i son simètrics (Figura 6.13).
- El sistema està basat en un espai de color no habitual, definit pels colors vermell, blau i groc amb un paràmetre de valor blanc o negre.

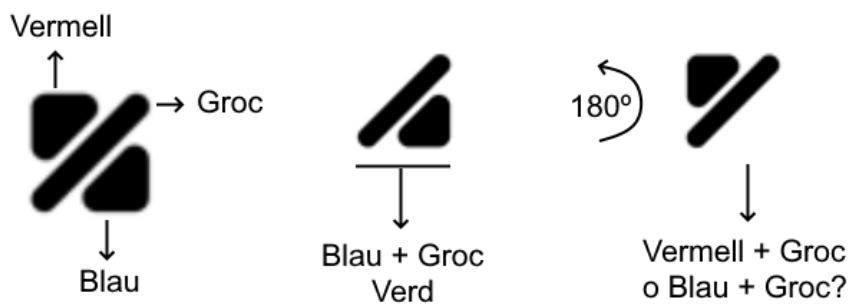


Figura 6.13 Demostració del problema de rotació del sistema ColorAdd. Font: Elaboració Pròpia

6.3.1 Disseny del codi visual

Per crear un sistema més consistent per videojocs, ja que el problema de les rotacions es pot donar si una d'aquestes figures es situa al terra, s'ha utilitzat la forma d'un triangle. Aquest sistema es basa en la roda de colors RGB de forma corresponent a les puntes d'aquest triangle (Figura 6.14).

Finalment per mostrar les components, aquest cop amb els color vermell, verd i blau, s'utilitza una figura diferent per cadascuna: un cercle pel color vermell, un triangle pel color verd i un quadrat pel color blau. Per crear més combinacions de colors es pinta un triangle invertit al centre del triangle de fons: si no hi ha el color no té peculiaritats, si és un triangle blanc amb vora negra és un color clar i si és un triangle de color negre és un color fosc.

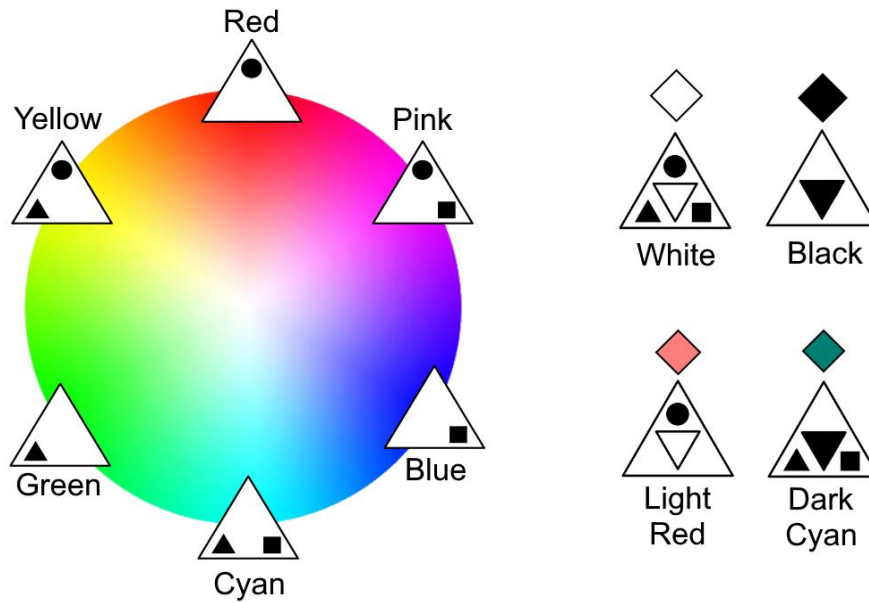


Figura 6.14 Explicació del nou codi visual de colors. Font: Elaboració Pròpia

6.3.2 Implementació d'un lector de color

La primera funcionalitat de l'eina en implementar el codi visual de colors és un sistema per llegir un píxel de la pantalla i traduir-lo a una figura d'aquest sistema.

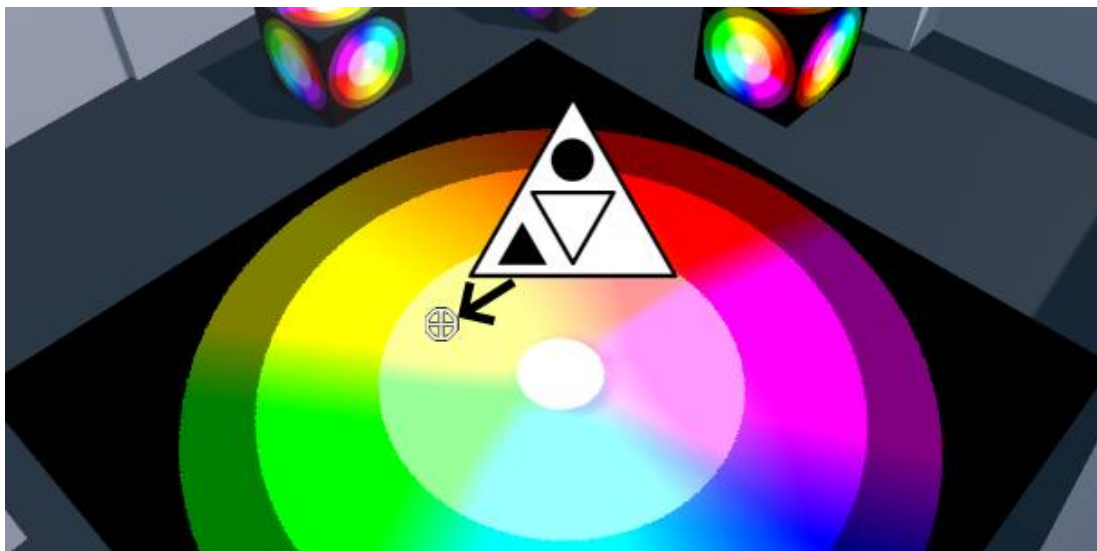


Figura 6.15 Funcionalitat de lector de color. Font: Elaboració pròpia.

Per utilitzar el sistema dissenyat l'eina ofereix una creació automàtica de les textures que defineixen la figura final a través d'un Scriptable Object i un component controlador amb les funcionalitats d'un singleton, tot gràcies a funcions estàtiques amb

el atribut MenuItem per cridar-les des del menú de l'editor. El component controlador definirà les textures de els figures de forma global en tots els shaders.

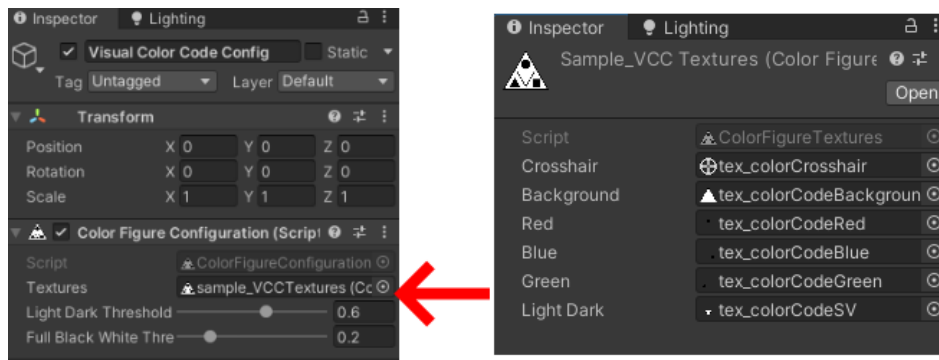


Figura 6.16 Configuració del sistema en Unity. Font: Elaboració pròpia.

El component controlador te 2 paràmetres que defineixen els límits per considerar si un color és clar o fosc i per considerar que el color és totalment blanc o negre (Figura 6.17).

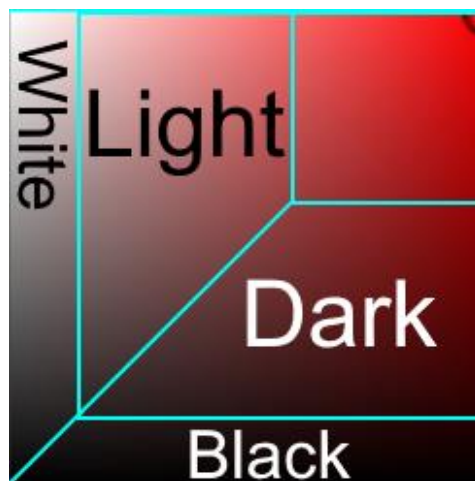


Figura 6.17 Umbrals definits pels paràmetres del controlador de les figures. Font: Elaboració pròpia.

L'eina també ofereix la creació automàtica de la retina per llegir el color. Per obtenir el color del píxel on es situa el component Transform d'un objecte es crea una Texture2D amb mida 1x1 píxels amb les coordenades que seleccionen el píxel corresponent de la Render Texture obtinguda pel primer component ApplyFilter, descrit en l'apartat 6.1.2, de la càmera principal, en aquest cas abans d'aplicar els filtres de l'editor. Amb la Texture2D creada, s'utilitza la funció ReadPixels per obtenir el color amb el que es realitzen les operacions per traduir-lo en una figura.

Les operacions per obtenir la figura es basen en activar i desactivar les figures que representen els colors primaris segons la component Hue més propera al segment que les defineix del color en sistema HSV (Figura 6.18).



Figura 6.18 Figures corresponents al component Hue més proper. Font: Elaboració pròpia.

6.3.3 Implementació de shaders

La segona implementació d'aquest sistema és un sistema per pintar la figura d'un color definit manualment sobre la geometria (Figura 6.18). La figura es pinta dintre dels límits de la geometria 3D segons l'àrea que ocupa en pantalla l'objecte.

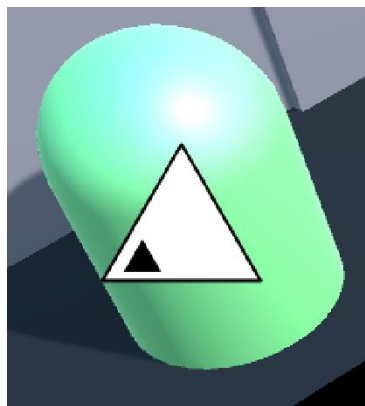


Figura 6.19 Demostració del shader que pinta la figura sobre la geometria. Font: Elaboració Pròpia.

Per trobar l'àrea que ocupa l'objecte en pantalla s'han utilitzat els bounds del component Renderer. El component Renderer d'un objecte en 3 dimensions defineix els seus bounds amb un cub que conté la geometria. Les coordenades resultants s'obtenen de projectar cadascun dels 8 punts que formen el cub a coordenades de pantalla, utilitzant la matriu de transformació de coordenades de món a coordenades de pantalla. L'àrea s'ha calculat com el quadrat definit pels màxims i mínims d'aquestes projeccions. Finalment, aquesta àrea en 2D serveix per calcular les noves coordenades de textura per aplicar l'addició de figures.

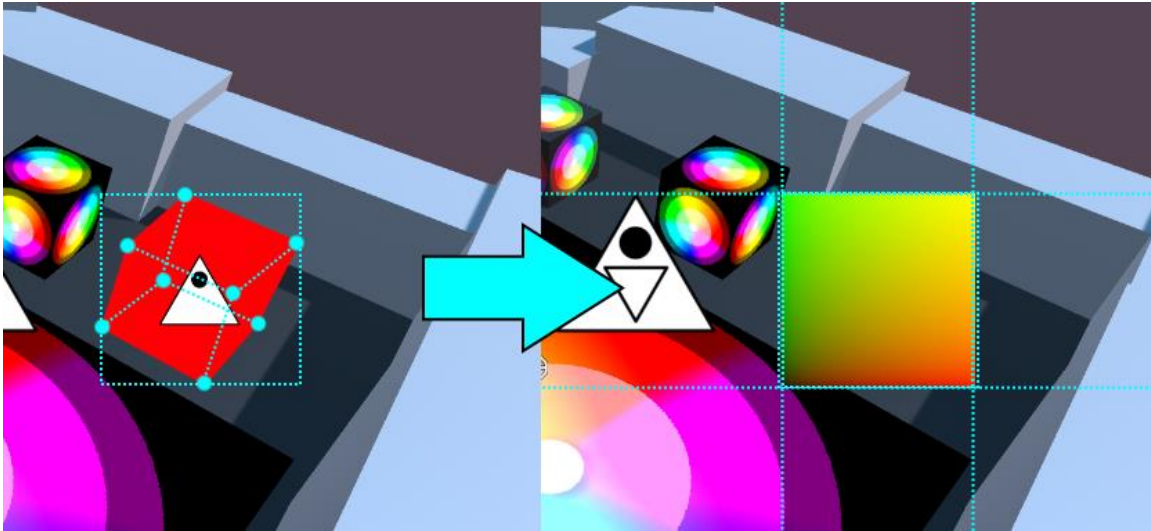


Figura 6.20 Àrea obtinguda amb els bounds de l'objecte, retallada de forma quadrada en cas del rectangle amb vora discontinua. Font: Elaboració pròpia.

El Renderer ha d'utilitzar un material definit per un shader especial. L'eina ofereix un shader surface i un shader unlit per utilitzar aquesta funcionalitat. En cas que l'usuari vulgui crear el seu propi shader, l'eina ofereix funcions per facilitar-ho, com els shaders d'alt contrast mencionats en l'apartat 6.2.3.

Per passar l'àrea i coordenades de l'objecte en pantalla s'utilitza un component Show Color Bounds, que s'ha d'incloure en l'objecte que tingui el component renderer. Aquest component permet definir manualment el color, la mida de la figura i desviar el sistema de coordenades de l'objecte si els bounds no corresponen amb la geometria (Figura 6.21).

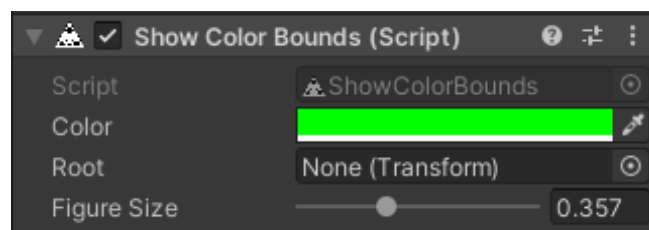


Figura 6.21 Component que s'utilitza per definir la figura pintada en la geometria. Font: Elaboració pròpia.

6.4 Documentació

La documentació de l'eina, trobada a l'Annex 2, s'ha realitzat en un document de text en anglès, exportat en PDF i inclòs dintre de les carpetes que importa l'arxiu amb extensió .unitypackage.

Seguint les directrius documentades en l'apartat 3.4.3, el document recull les funcionalitats i tutorials pas per pas per implementar les solucions que l'usuari escolleixi, amb exemples de codi incloure, indicacions de preparació de la escena, etc.

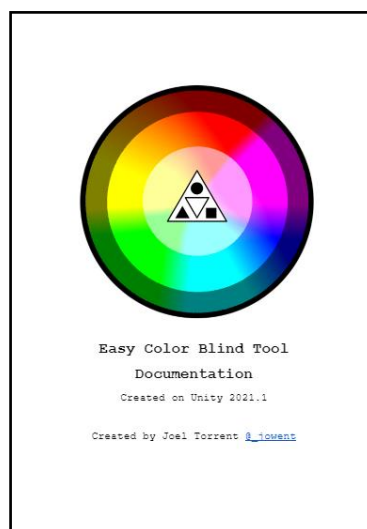


Figura 6.22 Portada de la documentació de l'eina. Font: Elaboració pròpia

6.5 Creació d'una escena de mostra i demostració

Perquè l'usuari final pugui provar totes les funcionalitats de forma ràpida, l'eina inclou una escena de mostra amb les dues implementacions per un joc final: mode d'alt contrast i codi visual de colors, mostrant tots els shaders que l'eina ofereix junt amb el lector de píxels mencionat en l'apartat 6.3.2.

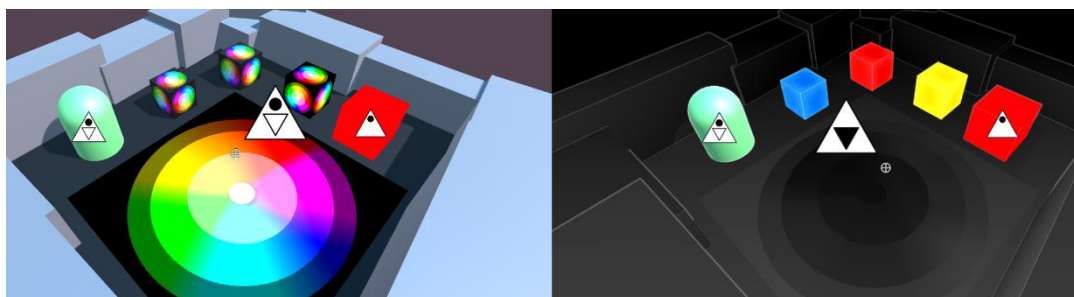


Figura 6.23 Escena de mostra inclosa en l'eina. Font: Elaboració pròpia.

Amb totes les parts de l'eina desenvolupades s'ha decidit fer una demo interactiva per navegador, trobada en l'Annex 5, ensenyant les funcionalitats i validant que el codi visual de colors s'entén. Aquesta demo es compon de 4 sales:

- Escena de filtres de simulacions: tot i reproduir-se fora de l'editor de Unity s'ha decidit mostrar els filtres ja que representen un terç de l'eina.
- Escena del mode d'alt contrast: una opció que activa i desactiva el mode amb una configuració prèvia.
- Escena del codi visual de colors: una sala dividida en dues parts marcades per una línia discontinua en el terra que al creuar-la s'activa la simulació d'acromatòpsia. En les parets hi ha elements d'explicació d'aquest codi de colors
- Laberint del codi visual de colors: per validar que el codi s'ha entès s'ha generat un petit laberint que el jugador haurà de resoldre seguint una indicació d'un color en text i creuant la respectiva porta que tingui la figura amb el codi de colors.

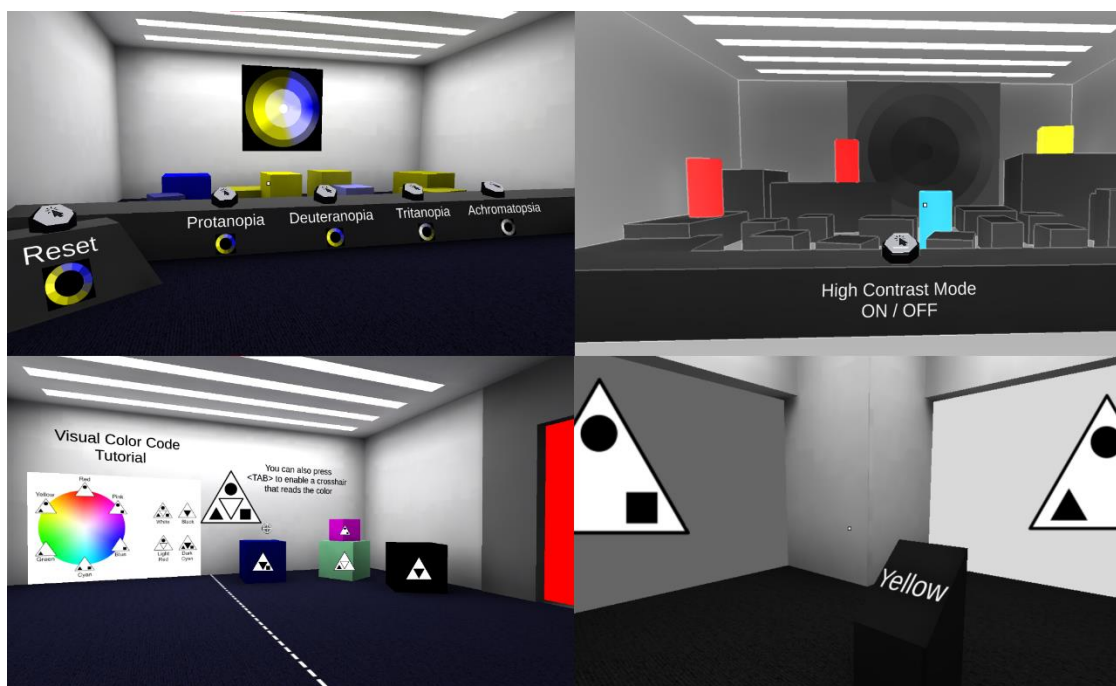


Figura 6.24 Sales que formen la demo interactiva de l'eina. Font: Elaboració pròpia.

6.6 Publicació

S'ha decidit publicar la demo interactiva de l'eina en la plataforma itch.io perquè permet crear una mateixa pàgina que permet provar una build en navegador, adjuntar informació de text com una descripció i adjuntar arxius com l'exportació en extensió .unitypackage de l'eina final.

Així doncs, l'eina s'ha publicat en la mateixa pàgina de la demo però també s'ha publicat en la plataforma github.com ja que està orientada al control de versions per futures actualitzacions i millores. S'ha descartat publicar l'eina en la Unity Asset Store de Unity perquè s'ha enfocat a una publicació gratuïta i la investigació de l'apartat 3.3.3 no ha reforçat que pugui tenir una bona difusió en aquesta plataforma.

6.7 Testeig

Per realitzar un testeig de l'eina de forma que pugui arribar al major nombre de persones possibles s'ha utilitzat la demo interactiva de l'apartat 6.5 com a eix principal per construir un formulari de Google Forms.

6.7.1 Construcció d'un formulari

El formulari ha estat compost de les següents preguntes:

- Ets una persona amb daltonisme?
- Com d'útils creus que pot ser aquesta eina per desenvolupar videojocs?
- Quina de les 3 implementacions (filtres, mode d'alt contrast, codi visual de colors) creus que és la millor?
- Has arribat al final del laberint?
- Has entès el codi visual de colors?
- Com de fàcil/difícil d'entendre t'ha semblat el codi visual de colors?
- Consideraràs utilitzar l'eina en futurs projectes?
- Apartat de comentaris opcionals.

El formulari s'ha difós per les plataformes itch.io (la mateixa pàgina de la demo interactiva), twitter.com i reddit.com, concretament en les comunitats de daltonisme i desenvolupament de videojocs.

6.7.2 Resultats

S'ha considerat part del testeig el formulari proposat i el feedback qualitatiu en les comunitats on s'ha difós la demo interactiva. Aquests resultats es troben el l'Annex 5.

Les dades obtingudes del formulari han estat les següents:

- Nombre de participació: 12 persones.
- Un 16,7% (2 persones) eren daltòniques.
- La majoria dels participants (58,3%) consideren que l'eina pot resultar molt útil pel desenvolupament de videojocs (5 sobre 5 d'utilitat) encara que el percentatge restant ha puntuat la utilitat per sobre del punt mig d'utilitat (3 sobre 5).
- Hi ha un empat entre les persones que han considerat la part de simulació de filtres i les persones que han considerat el mode d'alt contrast com l'apartat més útil de l'eina (41,7% en els dos casos).
- De les 12 persones només 1 no ha arribat al final del laberint descrit en l'apartat 6.5. És el mateix cas per les persones que no han entès el codi visual de colors en la seva totalitat.
- Un 25% dels participants ha considerat el codi de colors molt fàcil d'aprendre (1 sobre 5 en dificultat), un 33 relativament fàcil (2 sobre 5) i un 41,7% difícil (4 sobre 5)
- La meitat de les persones han considerat utilitzar l'eina en un futur, sent un 66% considerats desenvolupadors de videojocs.
- Les dues persones daltòniques han superat el laberint, han entès el codi visual de colors completament i han puntuat l'eina amb un 5 sobre 5 d'utilitat en el desenvolupament de videojocs.

Per la part de feedback qualitatiu (recollit de l'últim apartat opcional del formulari, la comunitat de reddit r/colorblind i comentaris de la publicació a twitter) ha estat generalment positiu. Les dades a destacar han estat les següents:

- Nombre de comentaris: 7.
- Dues persones del formulari no daltòniques, reforcen que el codi visual de colors pot arribar a ser confós o innecessari, substituint la figura per un text indicant el color.

- Dues persones daltòniques, una del formulari i un usuari de twitter, consideren interessant treure el color fora de l'equació per navegar per un entorn com el laberint mencionat.

7 Conclusions

En aquest capítol es realitza una retrospectiva de la recerca, l'eina acabada i el seu testeig per separat.

7.1 Investigació

Aquest treball ha partit de solucionar un problema, i la investigació ha estat crucial per platejar les diverses solucions proposades per l'eina resultant. Gràcies a conèixer les arrels d'una discapacitat i com s'han desenvolupat diferents propostes per millorar l'experiència dels daltònics, tant en casos quotidians com en casos específics de videojocs s'ha pogut dissenyar tenint un objectiu clar perquè l'eina desenvolupada pugui ser útil.

Una de les parts a destacar de la recerca és l'aplicació dels filtres per simular una visió de persona daltònica ja que, a priori, pot semblar que es pot arribar amb un simple post processat, però arribar a les fórmules matemàtiques per assolir el filtre implementat en l'eina transformant l'espai de color i reduint-lo en la seva forma que simula els valors dels cons de l'ull humà ha reforçat la complexitat del problema original.

En la mateixa investigació s'han pogut contrastar diferents referents, fet que ha portat a optimitzar la qualitat dels resultats, com dissenyar un nou codi visual de colors per una falta de coherència amb un espai de colors estandarditzat, o evitar publicar en la Unity Asset Store amb una aproximació actual del seu funcionament on la publicació pot resultar fútil, per l'alt nombre d'altres paquets d'assets que poden ofuscar-la, més encara quan s'ha decidit que l'eina és gratuïta.

7.2 Objectius

Amb l'eina acabada, es pot afirmar que s'han assolit tots els objectius principals. No obstant s'ha considerat eliminar dos objectius secundaris per la seva importància o rellevància amb la investigació. És el cas del mode Debug proposat per trobar de forma automàtica discrepàncies en els diferents casos de daltonisme, ja que canviant el filtre amb un botó ja es pot observar aquesta diferència, i el cas d'ampliar les

funcionalitats de l'eina per altres discapacitats visuals, que s'ha considerat fora dels marges d'aquest treball.

7.3 Testeig de l'eina

El testeig de l'eina no ha tingut la difusió esperada. Tot i que el temps de recollir resultats ha estat reduït (dues setmanes) només s'ha pogut arribar a obtenir 12 respostes al formulari.

Les hipòtesis de la causa d'aquest baix nombre de participació són les següents:

- Com el producte final és una eina, el públic objectiu queda reduït a desenvolupadors de videojocs que utilitzin Unity o persones afectades directament pel daltonisme interessades en el desenvolupament de videojocs.
- El procés de prova de 2 passos (demo i formulari) ha fet que part dels usuaris que han jugat la demo hagin obviat respondre a les preguntes.

Tot i el baix nombre de participació en el testeig no s'ha donat cap cas d'una persona daltònica que consideri un mal disseny de l'eina desenvolupada en aquest treball. Tot i així, per una correcta validació i futures investigacions, s'ha d'obtenir un nombre més significatiu de participants en aquest procés.

8 Bibliografia/Referències

8.1 Referents

- Almutairi, N., Kundart, J., Muthuramaligam, N., Hayes, J., & Citek, K. (2017). Assessment of Enchroma Filter for Correcting Color Vision Deficiency [Tèsis] Pacific University, College of Optometry.
- Anbari, S., Hamidi, H. R., & Kermanshahani, S. (2021). *Color-blindness simulation for red-green and blue-yellow ambiguity. Journal of Computational Methods in Sciences and Engineering*, 1–12. doi:10.3233/jcm-215023
- Aslam, H., & Brown, J. A. (2020). *Affordance Theory in Game Design: A Guide Toward Understanding Players. Synthesis Lectures on Games and Computational Intelligence*, 4(1), 1–111. doi:10.2200/s00977ed1v01y201912gc
- Bloch, J. (2006, October). How to design a good API and why it matters. In Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications (pp. 506-507)
- Blow, J. (2004). Game Development Harder Than You Think. Queue February 2004, 29-37. Recuperat de: <https://queue.acm.org/detail.cfm?id=971590>
- Campbell, K. (2020). The Game Awards Now Has An Accessibility (AKA The Last Of Us 2) Category. THEGAMER. Recuperat de: <https://www.thegamer.com/game-awards-accessibility-category-last-of-us-2/>
- Card, D., & Mitchell, J. L. (2002). Non-photorealistic rendering with pixel and vertex shaders. Direct3D ShaderX: vertex and pixel shader tips and tricks, 319-333.
- Chidlow, G., Chan, W. O., Wood, J. P., & Casson, R. J. (2022). Differential Effects of Experimental Retinal Detachment on S-and M/L-Cones in Rats. *Molecular Neurobiology*, 59(1), 117-136.

- Dix, A., Finlay, J., Abowd, G. & Beale, R. (2004). *Human Computer Interaction* (3a edició) Prentice-Hall.
- Doppioslash, C. (2017). Post-Processing Effects. *Physically Based Shader Development for Unity 2017*, 121–135. doi:10.1007/978-1-4842-3309-2_10
- Elliot, A. J., & Maier, M. A. (2014). *Color Psychology: Effects of Perceiving Color on Psychological Functioning in Humans. Annual Review of Psychology*, 65(1), 95–120. doi:10.1146/annurev-psych-010213-115035
- Everitt, C. (2001). Interactive order-independent transparency. White paper, nVIDIA, 2(6), 7.
- Gang, S.-M., Choi, H.-W., Kim, D.-R., & Choung, Y.-J. (2016). *A Study on the Construction of the Unity 3D Engine Based on the WebGIS System for the Hydrological and Water Hazard Information Display. Procedia Engineering*, 154, 138–145. doi:10.1016/j.proeng.2016.07.431
- Garber, L. (2013). *Game Accessibility: Enabling Everyone to Play. Computer*, 46(6), 14–18. doi:10.1109/mc.2013.206
- Glauber, R.J. (1963). The Quantum Theory of Optical Coherence. *Physical Review*, Volume 130 Number 6, 2529-2539.
- Godineau, C. (2018). The new Xbox adaptive controller, another step towards digital inclusion?. *Masters of Media, New Media & Digital Culture M.A.*, University of Amsterdam.
- Gómez-Robledo, L., Valero, E., M. Huertas, R. Martínez-Domingo, M. A., & Hernández-Andrés, J. (2018). Do EnChroma glasses improve color vision for colorblind subjects?. *Opt. Express* 26. 28693-28703.
- Greene, N., Kass, M., & Miller, G. (1993). *Hierarchical Z-buffer visibility. Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93*. doi:10.1145/166117.166147
- Gregory, J. (2018). *Game engine architecture*. AK Peters/CRC Press.

- Ghezzi, C., & Monga, M. (2002). Fostering component evolution with C# attributes. Proceedings of the International Workshop on Principles of Software Evolution - IWPSE '02. doi:10.1145/512035.512041
- Gunther, K., Neitz, J., & Neitz, M. (2006). A novel mutation in the short-wavelength-sensitive cone pigment gene associated with a tritan color vision defect. *Visual Neuroscience* 23, 403-409. DOI: 10.1017/S0952523806233169
- Haas, J. K. (2014). A History of the Unity Game Engine. Recuperat de <https://digitalcommons.wpi.edu/iqp-all/3207>
- Halladay, K. (2019). Practical Shader Development. doi:10.1007/978-1-4842-4457-9
- Hardin, B (2016). Colorblind accessibility in video games – is the industry heading in the right direction?. *The Gamer's Experience*. Recuperat de: <https://www.gamersexperience.com/colorblind-accessibility-in-video-games-is-the-industry-heading-in-the-right-direction/>
- He, Y., Foley, T., Hofstee, T., Long, H., & Fatahalian, K. (2017). Shader components. *ACM Transactions on Graphics*, 36(4), 1–11. doi:10.1145/3072959.3073648
- Herwig, A., & Paar, P. (2002). Game Engines: Tools or Landscape Visualization and Planning?. *Trends in GIS and Virtualization in Environmental Planning and Design*. Wichmann.
- Henning, M. (2009). *API design matters*. *Communications of the ACM*, 52(5), 46. doi:10.1145/1506409.1506424
- Hussain, A., Shakeel, H., Hussain, F., Uddin, N., & Ghouri, T. L. (2020). Unity Game Development Engine: A Technical Survey. *University of Sindh Journal of Information and Communication Technology* Vol 4, Issue 2.
- Klein, A., Sørensen, C., Freitas, A. S., de Pedron, C. D., & Elaluf-Calderwood, S. (2020). *Understanding controversies in digital platform innovation processes: The Google Glass case*. *Technological Forecasting and Social Change*, 152, 119883. doi:10.1016/j.techfore.2019.119888

- Kok, B. (2021). *Beginning Unity Editor Scripting*. Apress.
- Lamb, T.D. (2016). Why rods and cones?. *Eye* 30, 179-185. DOI: 0950-222X/16
- Lapidous, E., & Jiao, G. (1999). Optimal depth buffer for low-cost graphics hardware. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware - HWWS '99*. doi:10.1145/311534.311579
- Lausegger, G., Spitzer, M., & Ebner, M. (2017). OmniColor - A Smart Glasses App to Support Colorblind People. *iJim* Vol. 11. N°5. DOI: <https://doi.org/10.3991/ijim.v11i5.6922>
- Lewis, G. (1926). The Conservation of Photons. *Nature* 118, 874-875. DOI: <https://doi.org/10.1038/118874a0>
- Luna, F. (2006). *Introduction to 3D Game Programming with DirectX 9.0 c: A Shader Approach: A Shader Approach*. Wordware Publishing, Inc..
- Magdics, M., Sauvaget, C., García, R. J., & Sbert, M. (2013). Post-processing NPR effects for video games. *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry - VRCAI '13*. doi:10.1145/2534329.2534348
- Meng, M., Steinhardt, S., & Schubert, A. (2017). *Application Programming Interface Documentation: What Do Software Developers Want? Journal of Technical Writing and Communication*, 48(3), 295–330. doi:10.1177/0047281617721853
- McGuire, D (2019). Federal Communications Commission Rules Require in Video Games to Be Accessible and Usable. *The National Law Review*
- Miesenberger, K., Ossmann, R., Archambault, D., Searle, G., & Holsinger, A. (2008). More Than Just a Game: Accessibility in Computer Games. *Lecture Notes in Computer Science*, 247-260. DOI: 10.1007/978-3-540-89350-9_18
- Mukundan, R. (2022). The Geometry Shader. In *3D Mesh Processing and Character Animation* (pp. 73-89). Springer, Cham.

- Napoli, D., & Chiasson, S. (2018). Exploring the Impact of Colour-Blindness on Computer Game Performance. SIGCHI Conference on Human factors in Computing Systems (CHI) Late Breaking Work, 2018.
- Neiva, M. (2017). ColorADD: Color Identification System for Color-Blind People. *Injuries and Health Problems in Football*, 303–314. doi:10.1007/978-3-662-53924-8_27
- Noor, I., Mokhtar, H., Rafiqul, K., & Pramod, M. (2012). Understanding Color Models: A Review. *ARPN Journal of Science and Technology Vol 2*, 265–275.
- Oesch, N. W., & Diamond, J. S. (2011). *Ribbon synapses compute temporal contrast and encode luminance in retinal rod bipolar cells. Nature Neuroscience*, 14(12), 1555–1561. doi:10.1038/nn.2945
- Packard, B., & Phillips, J. (2019, April). An examination of the ethics of asset flips. In 34th Annual Conference of The Pennsylvania Association of Computer and Information Science Educators (p. 47).
- Park, H., & Baek, N. (2020). Developing an Open-Source Lightweight Game Engine with DNN Support. *Electronics*, 9(9), 1421. doi:10.3390/electronics9091421
- Persson, E. (2007). Depth in-depth. technical report, ATI Technologies, Inc.
- Pierre, F., Aujol, J.-F., Bugeau, A., & Ta, V.-T. (2015). *Luminance-Hue Specification in the RGB Space. Scale Space and Variational Methods in Computer Vision*, 413–424. doi:10.1007/978-3-319-18461-6_33
- Pixel Maniacs (2016). Fanlove and Colorblind Mode. Pixel Maniacs. Recuperat de: <https://pixel-maniacs.com/blog/2016-07-08-fanlove-and-colorblind-mode>
- Pratama, L. A., & Dossick, C. S. (2018). Workflow in Virtual Reality Tool Development for AEC Industry. *Advances in Informatics and Computing in Civil and Construction Engineering*, 297–306. doi:10.1007/978-3-030-00220-6_36

- Pontin, R., de Lima, A., de Souza, F., Agostini, L., & Nogueira, E. A. (2017). Game Accessibility Evaluation Methods: A Literature Survey. UAHCI 2017, Part I, LNCS 10277 182-102
- Roorda, A. i Neitz, J (2000). Chromatic Topography of The Retina. Journal of the Optical Society of America A 17(3):495-496. DOI: 10.1364/JOSAA.17.000495
- Shrivakshan, G. T., & Chandrasekar, C. (2012). A comparison of various edge detection techniques used in image processing. International Journal of Computer Science Issues (IJCSI), 9(5), 269.
- Schmalz, M. (2015). Limitation to Innovation in the North American Console Video Game Industry 2001-2013: A Critical Analysis [Tèsis] The University of Western Ontario. Recuperat de: <https://ir.lib.uwo.ca/etd/3393/>
- Simunovic, M. (2009). Color vision deficiency. Nature, Eye 24, 747-755. DOI: 10.1038/eye.2009.251
- Tadres, A (2015). Extending Unity with Editor Scripting: Put Unity to use for your video games by creating your own custom tools with editor scripting.
- Tanuwidjaja, E., Huynh, D., Koa, K., Nguyen, C., Shao, C., Torbett, P., & Weibel, N. (2014). *Chroma. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*. doi:10.1145/2632048.2632091
- Tariq, S. (2009). D3D11 Tessellation. In Game Developers Conference. Session: Advanced Visual Effects with Direct3D for PC.
- Theoharis, T., Papaioannou, G., & Karabassi, E. A. (2001). The magic of the z-buffer: A survey.
- Unity (2021) Unity User Manual 2021.3 (LTS). Recuperat de: <https://docs.unity3d.com/Manual/UnityManual.html>

- Viénot, F., Brettel, H., & Mollon, J. D. (1999). *Digital video colourmaps for checking the legibility of displays by dichromats*. *Color Research & Application*, 24(4), 243–252. doi:10.1002/(sici)1520-6378(199908)24:4<243::aid-col5>3.0.co;2-3
- Warrant, E. J. (2015). Photoreceptor Evolution: Ancient “Cones” Turn Out to Be Rods. *Current Biology*, 25(4), R148–R151. doi:10.1016/j.cub.2015.01.005
- Westin, T., Hamilton, I., & Ellis, B. (2020). *Game Accessibility from: The Digital Gaming Handbook*. CRC Press. DOI: 10.1201/9780429274596
- Westin, T., Ku, J. J., Dupire, J., & Hamilton, I. (2018). *Game Accessibility Guidelines and WCAG 2.0 – A Gap Analysis*. *Computers Helping People with Special Needs*, 270–279. doi:10.1007/978-3-319-94277-3_43
- Woods, C.T. Rudd, J. Robertson, S. Davids, K. (2020). *Wayfinding: How Ecological Perspectives of Navigating Dynamic Environments Can Enrich Our Understanding of the Learner and the Learning Process in Sport*. *Sports Medicine - Open* 6, Article number: 51.
- Young, T. (1801). *The Bakerian Lecture. On the Theory of Light and Colours*. By Thomas Young, M. D. F. R. S. Professor of Natural Philosophy. in the Royal Institution. Dr. Young's , &c. 12-48.
- Zaccardi, G. (2006). Color discrimination in the red range with only one long-wavelength sensitive opsin. *Journal of Experimental Biology*, 209(10), 1944–1955. doi:10.1242/jeb.02207
- Ziou, D., & Tabbone, S. (1998). *Edge detection techniques-an overview*. *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, 8, 537-559.

8.2 Ludografia

Atari (1972). Pong [videojoc] Atari. Nova York, Estats Units.

iD Software (2016). Doom [videojoc] Bethesda Softworks. Rockville, Estats Units.

Naughty Dog (2020). The Last of Us Part II [videojoc] Sony Interactive Entertainment.

Pixel Maniacs (2015). ChromaGun [videojoc] Pixel Maniacs. Alemanya.

Extremely OK Games (2019). Celeste [videojoc] Extremely OK Games. Canadà.

Epic Games (1998). Unreal [videojoc] GT Interactive. Nova York, Estats Units.



Centres universitaris adscrits a la



Grau en Disseny i Producció de Videojocs

Creació d'una eina per desenvolupar tenint en compte discapacitats visuals ANNEXES

**Joel Torrent Barrero
Tutor: Rafael González**



Annex 1: Codi Font

Inclou l'eina (EasyColorBlindV0.1.unitypackage) i el projecte on s'ha desenvolupat.

Ruta del directori: Torrent_Barrero_Daltonisme_TFG/Annexos/1_Codi Font

Annex 2: Documentació

Inclou la documentació necessària per entendre, fer funcionar de forma correcta l'eina i crear shaders personalitzats amb les funcions facilitades.

Ruta del directori: Torrent_Barrero_Daltonisme_TFG/Annexos/2_Documentació

Annex 3: Vídeo explicatiu

Inclou un vídeo exemplificant la instal·lació i ús de l'eina en un projecte buit de Unity.

Ruta del directori: Torrent_Barrero_Daltonisme_TFG/Annexos/3_Demostració_Vídeo

Annex 4: Resultats de testeig

Inclou els resultats del formulari utilitzat pel testeig de l'eina i captures de pantalla de comentaris en les plataformes de publicació.

Ruta del directori: Torrent_Barrero_Daltonisme_TFG/Annexos/4_Resultats_Testeing

Annex 5: Enllaços de publicació

Inclou els enllaços de publicació de l'eina i de la demo interactiva per navegador.

Ruta del directori: Torrent_Barrero_Daltonisme_TFG/Annexos/5_Enllaços_de_publicació

Annex 6: Programari i llibreries utilitzades

Visual Studio Code: Editor de codi font desenvolupat per Microsoft. Versió 1.67.2.

Obtingut de: <https://code.visualstudio.com/>

C# for Visual Studio Code: Extensió del Llenguatge de programació utilitzat per programar la lògica dels components en Unity, per l'editor de codi utilitzat. Versió

1.24.0. Obtingut de: <https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp>.

Unity: Motor de joc on s'ha desenvolupat l'eina. Versió: 2021.1.15f1. Obtingut de: <https://unity3d.com/get-unity/download/archive>

HLSL: Llenguatge de programació utilitzat per escriure els shaders necessaris. Inclòs en Unity.

Unity UI: Conjunt d'elements necessaris pels apartats relacionats amb la interfície d'usuari de l'eina desenvolupada. Versió 1.0.0. Obtingut de: <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/ugui.html?q=unity%20ui#installing-unity-ui>

