



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΤΕΧΝΩΝ ΗΧΟΥ ΚΑΙ ΕΙΚΟΝΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Φοιτητής:

Νικόλαος Γκάκιος ΤΧ2015028

Τίτλος:

“Δημιουργία τρισδιάστατου βιντεοπαιχνιδιού με τίτλο ‘Davy
Jones’ Locker’”

Επιβλέποντες:

Βαλιανάτος Θωμάς

Κομιανός Βασίλης

Κέρκυρα Φεβρουάριος 2023

“Δημιουργία τρισδιάστατου βιντεοπαιχνιδιού με τίτλο ‘Davy
Jones’ Locker’”

Νικόλαος Γκάκιος

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία προβάλλει τη δημιουργία ενός βιντεοπαιχνιδιού, συγκεκριμένα ενός τρισδιάστατου, με πρώτου προσώπου κάμερα, με γρίφους και μηχανισμούς εξερεύνησης, από το πρωταρχικό στάδιο του μέχρι και την πλήρη ολοκλήρωση του. Αρχικά θα επεξηγηθούν ορισμοί όσο αναφορά το είδος του βιντεοπαιχνιδιού και τη σύνδεση των στοιχείων με τον πραγματικό κόσμο, καθώς και την ένωση φανταστικών μη ρεαλιστικών στοιχείων μέσα σε αυτόν. Έπειτα, παρατίθεται η μεθοδολογία και τα τεχνικά εργαλεία που χρησιμοποιήθηκαν για την πλήρη υλοποίηση του έργου, κάποιες ιστορικές πληροφορίες, οι λόγοι που χρησιμοποιήθηκαν, οι τρόποι που χρησιμοποιήθηκαν, προσωπικές εφαρμογές και δυσκολίες που ανέκυψαν στην πορεία, καθώς και τρόπους επίλυσης αυτών των δυσκολιών. Τέλος, στον επίλογο θα αναφερθούν πιθανές προεκτάσεις του έργου και γνώσεις που αποκτήθηκαν στην πορεία της υλοποίησης του.

ABSTRACT

The current thesis projects the creation of a video game, specifically a 3D, with first person camera, full of puzzles and exploration mechanisms, from the start to its full completion. At first, definitions will be explained as to the genre of the video game's connection with the real world, as well as the connection of fantasy and unrealistic features within it. Then, the methodology and the technical tools used will be listen and observed, some historical information, the reasons used, the ways used, personal applications and difficulties encountered along the way, as well as ways to solve those difficulties. In the end, the epilogue will be full of possible extensions of the project and the knowledge gained along the way to the creation of it.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ, 4

ABSTRACT, 4

ΠΕΡΙΕΧΟΜΕΝΑ, 5

ΕΙΣΑΓΩΓΗ, 7

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ, 8

1. Βιντεοπαιχνίδια

1.1 Τι είναι ένα βιντεοπαιχνίδι - ηλεκτρονικό παιχνίδι; και ποια είδη υπάρχουν;

1.2 Ο ρόλος της κάμερας και του χώρου.

1.3 Η τέχνη του σχεδιασμού παιχνιδιών, της οπτικοποίησης ιδεών και του σχεδιασμού χαρακτήρων.

1.4 Ο σχεδιασμός του “Davy Jones’ Locker”.

ΚΥΡΙΟ ΜΕΡΟΣ, 19

2. Τα είδη των γραφικών, μοντελισμός και υφές.

2.1 Τα είδη των γραφικών και τα γραφικά στο “Davy Jones’ Locker”.

2.2 Μοντελισμός και πολύγωνα.

2.3 Υφές - Textures.

2.4 Σχεδίαση της διεπαφής και εμπειρίας χρήστη.

3. Μηχανές παιχνιδιών και προγραμματισμός.

3.1 Unity, Unreal Engine και γλώσσες προγραμματισμού.

3.2 Προγραμματισμός του “Davy Jones’ Locker”.

4. Ήχος και η σχεδίαση ήχου.

4.1 Ο ήχος στο “Davy Jones’ Locker”.

5. Βελτιστοποίηση των επιδόσεων του βιντεοπαιχνιδιού.

5.1 Πολύγωνα - Υφές - Levels of Details.

5.2 Draw Calls και Performance Jumps.

5.3 Προγραμματισμός και ήχος.

5.4 Έλεγχος των επιδόσεων.

6. Το αποτέλεσμα του “Davy Jones’ Locker”.

ΕΠΙΛΟΓΟΣ, 77

7. Η εξέλιξη του “Davy Jones’ Locker”, οι γνώσεις που απέκτησα και το μέλλον.

ΒΙΒΛΙΟΓΡΑΦΙΑ, 78

ΕΙΣΑΓΩΓΗ

Τα βιντεοπαιχνίδια τα τελευταία χρόνια κάνουν την παρουσία τους όλο και μεγαλύτερη στον τομέα της ψυχαγωγίας. Οι προσωπικοί ηλεκτρονικοί υπολογιστές γίνονται συνεχώς περισσότερο προσβάσιμοι και οι τεχνολογίες εξελίσσονται σε τέτοιο βαθμό που η αγορά ενός υπολογιστή με στόχο το παίξιμο οποιουδήποτε βιντεοπαιχνιδιού να γίνεται όλο και πιο εφικτή. Ενώ η βιομηχανία ξεκίνησε με τίτλους που χρησιμοποιούσαν δισδιάστατα γραφικά λόγω των περιορισμένων δυνατοτήτων των τότε συστημάτων, σύντομα ήρθαν στην εξίσωση και τα τρισδιάστατα γραφικά, ενώ μέχρι και σήμερα υπάρχουν προβλήματα στην δημιουργία των βιντεοπαιχνιδιών τα οποία η κάθε ομάδα τα αντιμετωπίζει και τα επιλύει με τον δικό της ξεχωριστό τρόπο.

Η επιθυμία του να δημιουργήσω ένα βιντεοπαιχνίδι εξ' ολοκλήρου πηγάζει από τα νεαρά χρόνια της ηλικίας μου. Πάντα θαύμαζα τις δουλείες μεγάλων στούντιο παιχνιδιών και το πως συνδέονται όλα τα είδη τέχνης μέσα σε ένα είδος ψυχαγωγίας. Για τον παραπάνω λόγο, ο σκοπός μου ήταν να μάθω εγώ ο ίδιος τη διαδικασία της δημιουργίας ενός τέτοιου έργου και να μοιραστώ τις γνώσεις μου μέσω αυτής της εργασίας. Έτσι, αποφάσισα πως η πτυχιακή αυτή εργασία είχε ως στόχο την δημιουργία ενός βιντεοπαιχνιδιού από τα θεμέλια του, όπως επίσης και το μοίρασμα των γνώσεων που απέκτησα κατά τη διάρκεια. Το είδος του παιχνιδιού είναι επηρεασμένο από άλλα βιντεοπαιχνίδια εταιριών όπως της Capcom με τους τίτλους "Resident Evil", καθώς επίσης και από τα πραγματικά δωμάτια διαφυγής, γνωστά και ως "Escape Rooms". Η ιδέα του έργου αφορά ένα βιντεοπαιχνίδι γρίφων με μηχανισμούς εξερεύνησης των τρισδιάστατων αντικειμένων και του χώρου, καθώς και μίας ελαφριάς horror ατμόσφαιρας για την εμβύθιση του χρήστη στον χώρο.

Επομένως, η πτυχιακή αυτή εργασία αφορά την διαδικασία της δημιουργίας ενός τρισδιάστατου βιντεοπαιχνιδιού από την αρχή του έως το τέλος. Αυτό συμπεριλαμβάνει τις μεθόδους που προσωπικά χρησιμοποίησα για να εξάγω το αποτέλεσμα που ήθελα. Θα υπάρξει παράδειγμα του τρόπου της δημιουργίας του concept art καθώς και μία επίδειξη των εργαλείων που χρησιμοποιήθηκαν για την δημιουργία των τρισδιάστατων αντικειμένων. Θα αναλυθεί η διαδικασία του τρόπου του ντυσίματος των αντικειμένων με υλικά και η εισαγωγή των αντικειμένων μέσα σε μία μηχανή βιντεοπαιχνιδιών, όπου και θα δοθούν διάφορα παραδείγματα των μηχανισμών που έχουν δημιουργηθεί εκεί. Τέλος, θα υπάρξει

αναφορά στους τρόπους που οι επιδόσεις ενός βιντεοπαιχνιδιού μπορούν να επηρεαστούν και αντίστοιχα τις λύσεις που ακολούθησα στο “Davy Jones’ Locker” αντίστοιχα.

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

1. Βιντεοπαιχνίδια

Η παρούσα πτυχιακή εργασία αφορά τη δημιουργία ενός βιντεοπαιχνιδιού, επομένως είναι λογικό να αναλυθεί τι σημαίνει ο όρος βιντεοπαιχνίδι ή και ηλεκτρονικό παιχνίδι. Επίσης θα εξεταστούν κάποια ιστορικά γεγονότα, τα είδη βιντεοπαιχνιδιών, το πως οι συνδυασμοί διαφορετικών ειδών μπορούν να δημιουργήσουν ξεχωριστά αποτελέσματα και ο συνδυασμός που χρησιμοποιήθηκε για τη εκτέλεση της παρούσας πτυχιακής εργασίας. Τέλος, θα εξεταστεί και ο ρόλος του σχεδιαστή παιχνιδιών αλλά και της εννοιολογικής τέχνης.

1.1 Τι είναι ένα βιντεοπαιχνίδι - ηλεκτρονικό παιχνίδι; και ποια είδη υπάρχουν;

Η λέξη βιντεοπαιχνίδι έρχεται από τις λέξεις βίντεο και παιχνίδι. Η λέξη βίντεο είναι η καταγραφή διακεκομμένων σταθερών εικόνων για την δημιουργία μίας ενιαίας συλλογής εικόνων που έχουν ως αποτέλεσμα την ψευδαίσθηση της συνεχής κίνησης σε μία οθόνη¹. Η λέξη παιχνίδι ήταν πάντα μία διχασμένη λέξη με πολλές έννοιες. Ένα παιχνίδι έχει κανόνες που πρέπει να γνωρίζουν και οι δύο παίκτες, οι παίκτες πρέπει να έχουν ίδιες δυνατότητες και να υπακούσουν αυστηρά στους κανόνες με αποτέλεσμα να ολοκληρώσουν την δραστηριότητα. Στόχος ενός παιχνιδιού είναι το να βγει ένας νικητής. Ένα παιχνίδι συνήθως έχει ως επιπλέον στόχο την διασκέδαση ή την ψυχαγωγία, συνάμα, μπορεί να έχει δύσκολες προκλήσεις και πολλές φορές μπορεί να χρησιμοποιείται για λόγους εκμάθησης². Σε συνδυασμό των δύο λέξεων βίντεο και παιχνίδι, βγαίνει η λέξη βιντεοπαιχνίδι, το οποίο έχει τους σκοπούς ενός παιχνιδιού με μέσω της διάδρασης των παικτών πάνω στο βίντεο μίας οθόνη. Αντίστοιχα ονομάζεται και ηλεκτρονικό παιχνίδι, καθώς ηλεκτρονικό σημαίνει πως

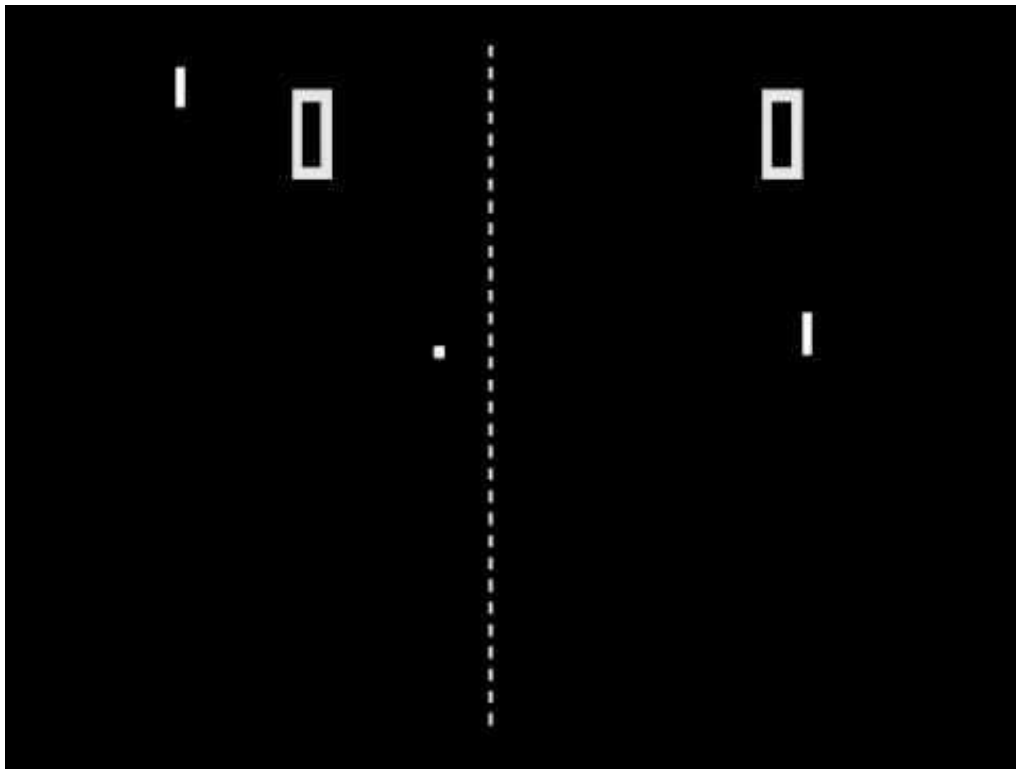
¹ Huang, D. A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., & Niebles, J. C. (2018b). What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7366 - 7373

² Suits, B. (1967). What Is a Game? *Philosophy of Science*, 34(2), 148–156.

γίνεται χρήση κάποιου ηλεκτρονικού μέσου για το παίξιμο αυτού του παιχνιδιού. Στην πραγματικότητα όμως, ένα βιντεοπαιχνίδι δεν αφορά μόνο την τέχνη του βίντεο σε ψηφιακή μορφή. Η δημιουργία ενός βιντεοπαιχνιδιού είναι η μείξη όλων των τεχνών και η σωστή διάδραση μεταξύ τους, ώστε να μπορέσει να γεννηθεί ένα άρτιο αποτέλεσμα.

Η ανάγκη του ανθρώπου να ψυχαγωγείται ήταν πάντα στο επίκεντρο της φύσης του. Για αυτόν τον λόγο ανά τα χρόνια δημιουργούσε πάντα παιχνίδια διαφορετικών κατατάξεων. Μερικά παιχνίδια ήταν επικεντρωμένα σε νεαρές ηλικίες, όπως για παράδειγμα το κρυφτό και το κυνηγητό, καθώς μερικά παιχνίδια εστίαζαν στην ψυχαγωγία εφήβων και ενήλικων ατόμων, όπως για παράδειγμα τα επιτραπέζια παιχνίδια και τα παιχνίδια με τράπουλες. Όταν ξεκίνησε η ψηφιακή εποχή, δεν άργησαν να υπάρχουν και πειραματισμοί στον βωμό της ψυχαγωγίας με τη δημιουργία των πρώτων βιντεοπαιχνιδιών με τίτλους “Galaxy Game” και “Pong” τις χρονολογίες 1971 και 1972 αντίστοιχα. Από την αρχή βλέπουμε το ενδιαφέρον της δημιουργίας διάφορων ειδών στον τομέα των βιντεοπαιχνιδιών. Το “Galaxy Game” ήταν ένα παιχνίδι επιστημονικής φαντασίας, με τη βοήθεια της φαντασίας ο χρήστης βρισκόταν στο διάστημα και πυροβολούσε αντικείμενα με στόχο την συλλογή πόντων και την αύξηση της βαθμολογίας του. Ενώ το “Pong” ήταν ένα παιχνίδι παρέας, στο οποίο ανταγωνιζόταν δύο παίκτες μεταξύ τους σε μία αναμέτρηση ενός δισδιάστατου ψηφιακού πινγκ πονγκ. Από τότε, με το πέρασμα των χρόνων έχουν δημιουργηθεί πολλά δεκάδες διαφορετικά μη τυποποιημένα είδη για όλες τις διαφορετικές προτιμήσεις του κάθε ανθρώπου³. Για παράδειγμα, κάποια πολύ γνωστά είδη είναι τα παιχνίδια δράσης, μυστηρίου, ρόλων, τρόμου, ανοιχτού κόσμου, πολλαπλών παικτών και πολλά ακόμα.

³ Rollings, A., & Adams, E. W. (2003). *Andrew Rollings and Ernest Adams on Game Design*. New Riders.
& Wolf, M. J. P. (2012). *Encyclopedia of Video Games: M-Z: The Culture, Technology and Art of Gaming* (Vol. 1). Greenwood. 543.



Εικόνα 1. “Pong” 1972.

1.2 Ο ρόλος της κάμερας και του χώρου.

Στα είδη των βιντεοπαιχνιδιών ανήκουν και ορολογίες που καθιστούν την τοποθέτηση της κάμερας στον χώρο. Η κάμερα μπορεί να τοποθετηθεί στα μάτια του χαρακτήρα που ελέγχει ο παίκτης, κάτι που το προσδιορίζει ως παιχνίδι πρώτου προσώπου. Εάν η κάμερα τοποθετηθεί πάνω από τον χαρακτήρα και βλέπουμε τις κινήσεις του, μπορεί να χαρακτηριστεί ως τρίτου προσώπου ή όψη από πάνω προς τα κάτω. Η κάμερα μπορεί να τοποθετηθεί στα πλάγια, αφήνοντας τον χαρακτήρα να κινείται μόνο αριστερά και δεξιά του χώρου. Η τοποθέτηση της κάμερας συνήθως σχετίζεται έμμεσα και με τον χώρο του παιχνιδιού. Τα βιντεοπαιχνίδια, επίσης, χαρακτηρίζονται ως τρισδιάστατα, δισδιάστατα και μία ενδιάμεση κατηγορία, τα παιχνίδια των δυόμιση διαστάσεων που συνδυάζουν τα δύο παραπάνω. Όλα τα παραπάνω παίζουν καθοριστικό ρόλο στον σχεδιασμό του βιντεοπαιχνιδιού και το συναίσθημα που θέλουν να αποδώσουν οι σχεδιαστές.

Ανάλογα με το βιντεοπαιχνίδι, ο χρήστης αντιλαμβάνεται από δύο έως τρεις διαστάσεις. Δύο διαστάσεις εννοείται πως ο παίκτης βλέπει μόνο την οριζόντια X και την κάθετη Y διάσταση του χώρου. Σε ένα τέτοιο παιχνίδι ο χαρακτήρας μπορεί να κινηθεί μόνο προς τους δύο αυτούς άξονες. Αυτά τα παιχνίδια αποτελούν τη γέννηση της αρχής του κόσμου των

βιντεοπαιχνιδιών, καθώς για να λειτουργήσουν χρειαζόταν μόνο ζωγραφιές των στοιχείων και των χαρακτήρων, πράμα που τα έκανε προσιτά για να καταφέρουν να λειτουργήσουν χωρίς επιπλοκές στα τότε συστήματα.



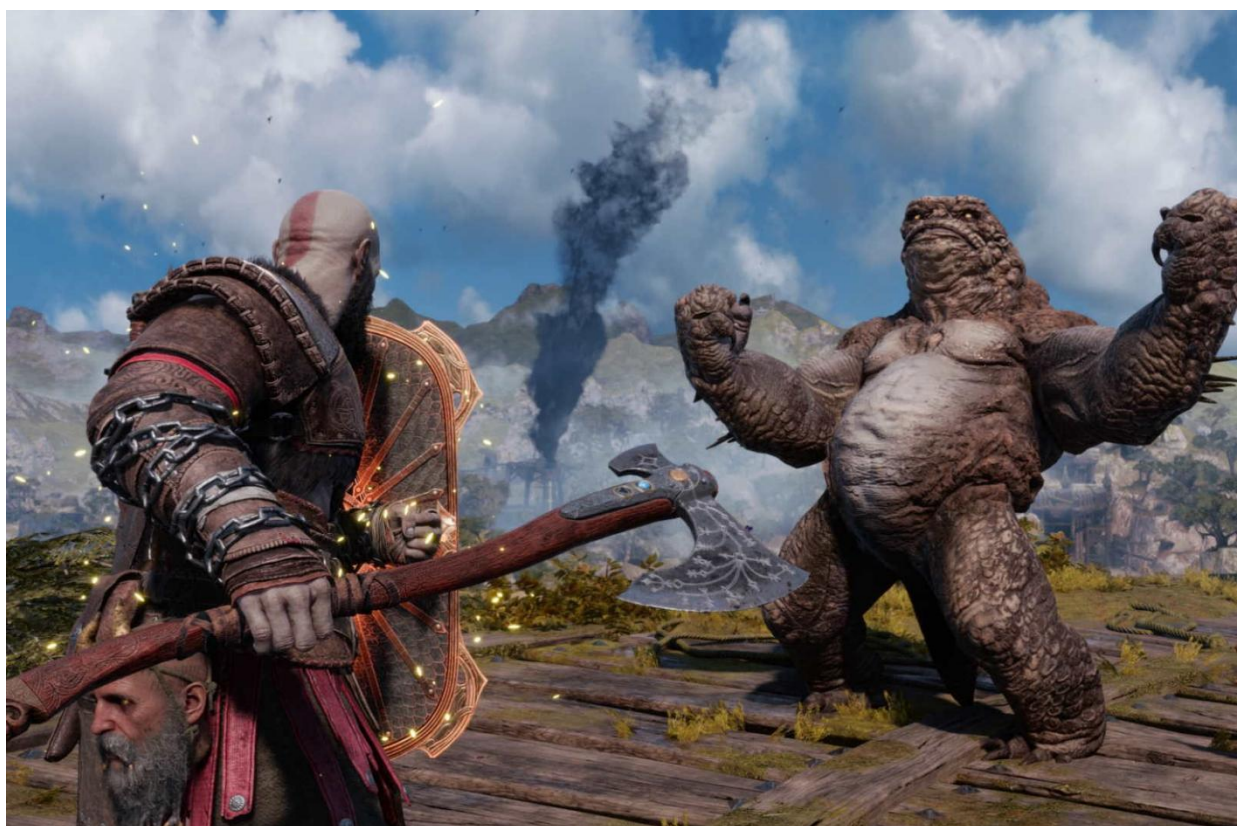
Εικόνα 2. δισδιάστατο παιχνίδι "Super Mario" 1985.

Στη συνέχεια υπάρχουν τα τρισδιάστατα βιντεοπαιχνίδια. Όσο εξελισσόταν η τεχνολογία των ηλεκτρονικών υπολογιστών, οι μηχανές παιχνιδιών και τα συστήματα που μπορούσαν να παίξουν παιχνίδια γινόταν όλο και πιο δυνατά. Με αυτόν τον τρόπο, στη δεκαετία του 90, ήρθαν στο προσκήνιο οι τρισδιάστατοι χάρτες υφής, με την αγγλική ονομασία “texture maps” στους οποίους θα υπάρξει ανάλυση στο κύριο μέρος της πτυχιακής εργασίας, καθώς είναι ένα από τα σημαντικότερα εργαλεία του κόσμου των βιντεοπαιχνιδιών ακόμη και σήμερα. Το πρώτο παιχνίδι με τρισδιάστατα στοιχεία ήταν το παιχνίδι ρόλων “Ultima Underworld- The Stygian Abyss”⁴, όπου μπορούσε ο παίκτης να δει και τους τρεις άξονες (X, Y, Z) στην οθόνη καθώς και να κινηθεί προς αυτούς.

⁴ Lampton, C. (2018, April 7). *The Birth of 3D Gaming*. Sublime Confusion.



Εικόνα 3. "Ultima Underworld - The Stygian Abyss" 1992.



Εικόνα 4. Τρισδιάστατο παιχνίδι "God of War Ragnarök" 2022.

Τέλος υπάρχουν και τα παιχνίδια των δυόμιση διαστάσεων, γνωστά και ως “2.5D”. Οι συγκεκριμένοι τίτλοι είναι συνήθως τρισδιάστατοι χώροι με δισδιάστατα στοιχεία και συνήθως δισδιάστατους χαρακτήρες. Σε αυτά τα παιχνίδια ο χρήστης κατά κύριο λόγο μπορεί να κινήσει τον χαρακτήρα σε δύο κατευθύνσεις, δίνοντας την ψευδαίσθηση του τρισδιάστατου χώρου λόγω της χρήσης τρισδιάστατου φόντου. Πολλά πλήρως τρισδιάστατα βιντεοπαιχνίδια για να μην καταλαμβάνουν πολλούς πόρους, με στόχο την βελτιστοποίηση των επιδόσεων του παιχνιδιού, χρησιμοποιούν δισδιάστατα στοιχεία αντί τρισδιάστατων για τη δημιουργία πολλών οπτικών εφέ. Όσο αναφορά το τι θεωρείται δισδιάστατο ή τρισδιάστατο στοιχείο και χαρακτήρας, και το πως επηρεάζεται η επίδοση του παιχνιδιού, θα αναλυθεί στο κύριο μέρος της εργασίας με πολλές περισσότερες λεπτομέρειες, στο κεφάλαιο της τρισδιάστατης μοντελοποίησης και το αντίστοιχο κεφάλαιο της βελτιστοποίησης των επιδόσεων του βιντεοπαιχνιδιού.



Εικόνα 5. "Don't Starve" 2013. Κίνηση σε δισδιάστατους άξονες, με δισδιάστατα στοιχεία και συνθέσεις πάνω σε ένα τρισδιάστατο επίπεδο για πάτωμα.



Εικόνα 6. "League of Legends" 2009. Τρισδιάστατο παιχνίδι με πολλά δισδιάστατα οπτικά εφέ.

1.3 Η τέχνη του σχεδιασμού παιχνιδιών, της οπτικοποίησης ιδεών και του σχεδιασμού χαρακτήρων.

Η τέχνη του σχεδιασμού ενός παιχνιδιού είναι μία τέχνη καλλιτεχνική αλλά και τεχνική. Κάποιος που δουλεύει στον σχεδιασμό ενός παιχνιδιού πρέπει να γνωρίζει και να πετυχαίνει μεγάλους καλλιτεχνικούς στόχους, ενώ ταυτόχρονα να γνωρίζει τι είναι εφικτό να υλοποιηθεί στον τεχνικό τομέα της δημιουργίας⁵. Ο σχεδιαστής ενός βιντεοπαιχνιδιού είναι αυτός που έχει τον λόγο για το ποια θα είναι η εξέλιξη του βιντεοπαιχνιδιού. Λαμβάνει τις αποφάσεις για την ολοκληρωμένη εμπειρία του παίκτη, όπως για παράδειγμα την τοποθέτηση της κάμερας, την δυσκολία του παιχνιδιού, την δυσκολία και την τοποθέτηση των γρίφων και όλα τα συναφή. Στη συνέχεια, ο "concept artist" είναι ο σχεδιαστής που δημιουργεί τέχνη σε χαρτί ή σε ψηφιακή μορφή, η οποία τέχνη απεικονίζει τα δημιουργικά στοιχεία και χαρακτηριστικά των χαρακτήρων ή των αντικειμένων ενός παιχνιδιού. Στη συνέχεια αυτά τα σχέδια δίνονται στους μοντελίστες με σκοπό την ετοιμασία τους για χρήση μέσα στο βιντεοπαιχνίδι⁶.

⁵ Crawford, C., & Crawford, C. (1984). *The Art of Computer Game Design: Reflections of a Master Game Designer*. McGraw-Hill Education, 51.

⁶ Concept Art Empire. (2018, February 14). What is a Concept Artist? *Concept Art Empire*, 1.

Ο concept artist συνεργάζεται άμεσα με τον σχεδιαστή του παιχνιδιού. Καθώς ο δεύτερος δίνει το έγγραφο σχεδιασμού του αντικειμένου στον πρώτο, και ο concept artist συνεχίζει στην εικονογράφηση του, συμβάλλοντας στο ύφος του παιχνιδιού και της γενικής ιδέας. Για αυτόν τον λόγο, ο concept artist είναι ένας από τους κύριους υπεύθυνους για το πως θα μοιάσει η εμφάνιση ενός ολοκληρωμένου βιντεοπαιχνιδιού. Αντίστοιχα, οι δημιουργοί των χαρακτήρων σε σκίτσο, γνωστοί και ως “character designers”, δημιουργούν το ύφος και το στυλ των χαρακτήρων του βιντεοπαιχνιδιού πριν καταλήξει στα χέρια του μοντελιστή χαρακτήρων.



Εικόνα 7. Concept art όπλου στο βιντεοπαιχνίδι “The Outer Worlds” 2019.



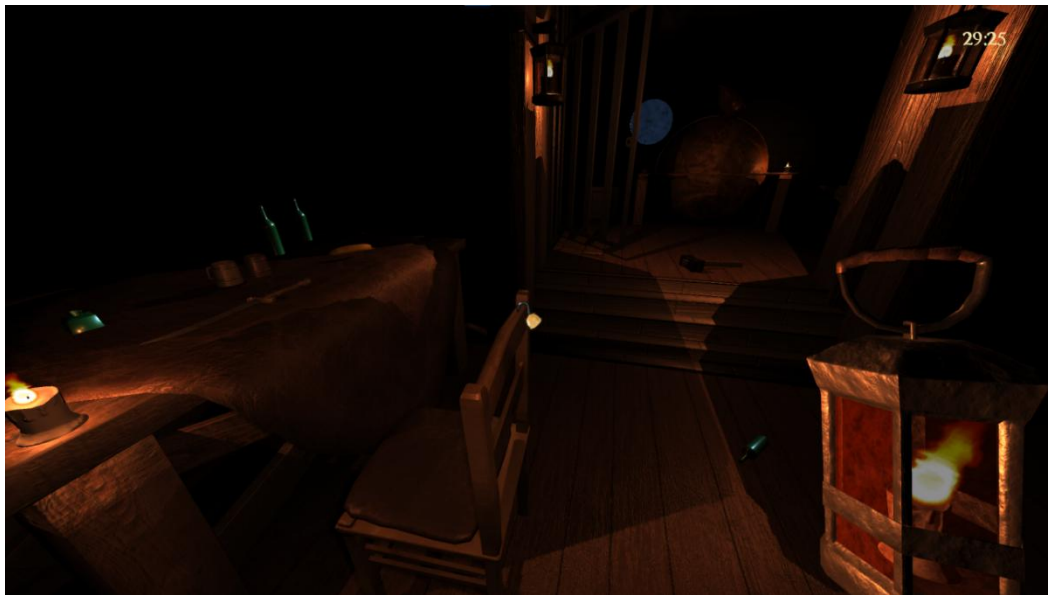
Εικόνα 8. Σχεδιασμός του πρωταγωνιστή στο βιντεοπαιχνίδι “Disco Elysium” 2019.

1.4 Ο σχεδιασμός του “Davy Jones’ Locker”.

Το βιντεοπαιχνίδι της πτυχιακής εργασίας είναι τοποθετημένο στα μέσα του δεκάτου εβδόμου αιώνα, στην χρυσή εποχή των πειρατών. Το είδος στο οποίο συγκαταλέγεται είναι, ένα τρισδιάστατο βιντεοπαιχνίδι γρίφων πρώτου προσώπου με μηχανισμούς εξερεύνησης. Το συναίσθημα που προσπαθεί να μεταδώσει είναι σκοτεινό, μοναχικό και λίγο ανατριχιαστικό. Ενώ οι γρίφοι θυμίζουν τα μοντέρνα δωμάτια διαφυγής, γνωστά ως “escape rooms”, με μη ρεαλιστικά στοιχεία που καθοδηγούν τον παίκτη στη λύση του γρίφου και όχι στην τρομακτική δυσκολία και απομάκρυνση του.

Ο τίτλος “The Davy Jones’ Locker” λέγεται πως ήταν μία πειρατική φράση που πρώτη φορά τη συναντάμε γραπτά από τον συγγραφέα Daniel Defoe το 1726 “...for he would toss them all

into Davy Jones' Locker if they did.”⁷. Davy Jones' Locker στην ουσία ονομαζόταν ο βυθός της θάλασσας. Η φράση αυτή ενέπνευσε το παιχνίδι της πτυχιακής εργασίας, καθώς στην ιστορία ο παίκτης είναι φυλακισμένος στο αμπάρι ενός μεγάλου πειρατικού πλοίου, ξυπνάει και συνειδητοποιεί πως όλοι έχουν εξαφανιστεί, με αποτέλεσμα να προσπαθεί να δραπετεύσει πριν τον καταπιεί ο βυθός της θάλασσας. Ο τρόπος που δούλεψα στον τομέα του concept art, ήταν η ένωση τύπου “moodboard” εικόνων από το διαδίκτυο που ήταν παραπλήσια στο αποτέλεσμα που ήθελα να πετύχω και η δημιουργία ενός γρήγορου σκίτσου του αντικειμένου που επιδίωκα να μοντελοποιήσω.



Εικόνα 9. Σκηνή του βιντεοπαιχνιδιού "Davy Jones' Locker".

⁷ Defoe, Daniel (1726). *The four years of voyages of capt. George Roberts*. P. [89](#).



Εικόνα 10. Concept art μαχαιριού με το τρισδιάστατο αποτέλεσμα στην κάτω εικόνα "Davy Jones' Locker".

ΚΥΡΙΟ ΜΕΡΟΣ

2. Τα είδη των γραφικών, μοντελισμός και υφές.

Στον κόσμο των βιντεοπαιχνιδιών υπάρχουν πολλά διαφορετικά καλλιτεχνικά στυλ, πολλά διαφορετικά προγράμματα μοντελισμού και τεχνικές που προσφέρουν λεπτομέρειες στα γραφικά. Το καλλιτεχνικό στυλ συνδέεται άμεσα με τις υφές ή τις δισδιάστατες ζωγραφιές που θα χρησιμοποιηθούν στα μοντέλα, καθώς σημαντικό ρόλο προσφέρει και ο τρόπος που έχει μοντελοποιηθεί ένα αντικείμενο. Οι δύο μεγαλύτερες κατηγορίες είναι τα στυλιζαρισμένα και τα ρεαλιστικά γραφικά, ενώ ταυτόχρονα αυτά χωρίζονται σε υποκατηγορίες. Ονομαστικά, τα πιο γνωστά στυλιζαρισμένα γραφικά που συνηθίζουμε να βλέπουμε σε δισδιάστατα παιχνίδια είναι τα πιξελιαστά, vector, cutout, cell shading και ζωγραφισμένα στο χέρι γραφικά. Συνάμα βρίσκουμε πιξελιαστά, vector, cell shading και ζωγραφισμένα στο χέρι πολύ συχνά σε τρισδιάστατα παιχνίδια. Από την άλλη, τα ρεαλιστικά γραφικά χωρίζονται σε τρεις υποκατηγορίες. Τα απλά ρεαλιστικά χωρίς απαραίτητη ανάγκη φωτορεαλισμού, τα ρεαλιστικά φαντασίας που αφορούν τα παιχνίδια φαντασίας με στοιχεία που δεν υπάρχουν στον πραγματικό κόσμο, και τα φωτορεαλιστικά που είναι φτιαγμένα για να είναι όσο πιο κοντά στην πραγματικότητα γίνεται.



Εικόνα 11. Δισδιάστατο παιχνίδι ζωγραφισμένο στο χέρι σε στυλ που θυμίζει τα παλιά κινούμενα σχέδια της Disney. "Cuphead" 2017.

Αποφασισμένος, λοιπόν, να φτιάξει κανείς ένα βιντεοπαιχνίδι, έχοντας αναφέρει τα παραπάνω, ενώ έχει καταλήξει σε ποιο είδος θα συγκαταλέγεται, πρέπει τώρα να αποβεί στο ύψος των γραφικών που θα χρησιμοποιηθούν και το πόσο λεπτομερείς θα είναι. Όλα αυτά συμβάλλουν σημαντικά στο αποτέλεσμα και στις δυνατότητες χρήσης του βιντεοπαιχνιδιού.

2.1 Τα γραφικά στο “Davy Jones’ Locker”.

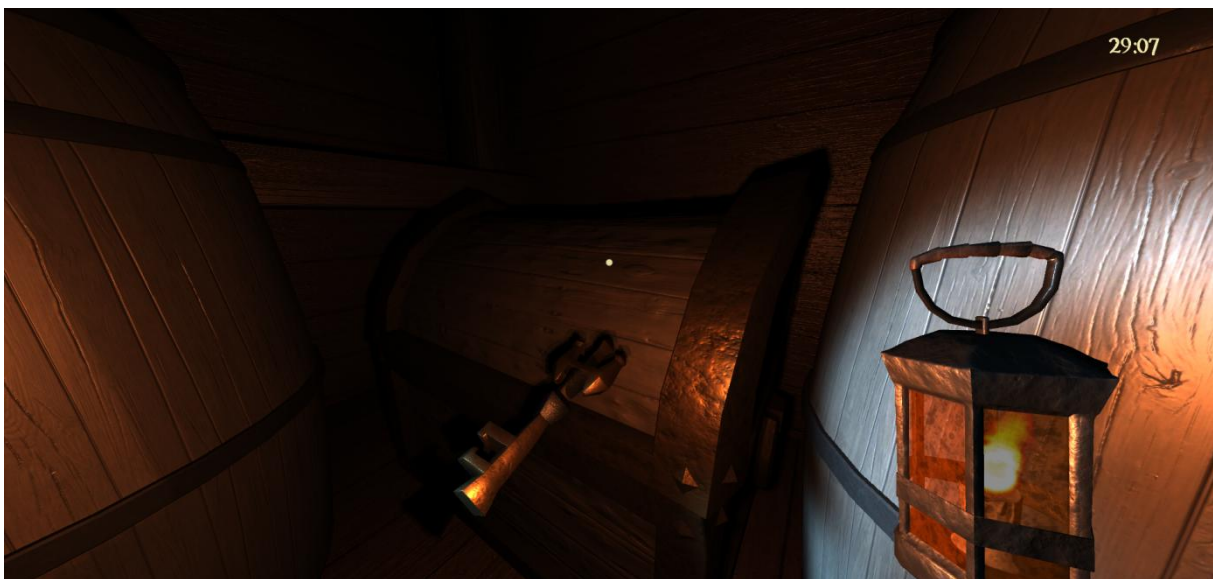
Η απόφαση μου αυτή της πτυχιακής εργασίας να γίνει με απλά ρεαλιστικά γραφικά φαντασίας ήταν επηρεασμένη από πολλαπλούς λόγους. Ένας κύριος ήταν η εξοικίωση μου με τη συγκεκριμένη ροή εργασιών. Τα τρισδιάστατα γραφικά που γνώριζα να δημιουργώ, κατά κύριο λόγο αφορούσαν τον ρεαλισμό, ενώ οι γνώσεις μου για κάποιο στυλιζαρισμένο είδος ήταν περιορισμένες. Η εμπύθιση ενός παίκτη μέσα σε έναν κόσμο φτιαγμένο με ρεαλιστικά γραφικά είναι επιπρόσθετα δύσκολη διαδικασία. Είναι αναγκαίο να δοθεί άψογη προσοχή στην λεπτομέρεια, καθώς οτιδήποτε μπορεί να φανεί αταίριαστο αφαιρεί από τον ρεαλισμό και υποβαθμίζει το αποτέλεσμα. Ένωσα έτσι το συναίσθημα της πρόκλησης, πράμα που με ενέπνευσε αρκετά να μελετήσω τον πραγματικό κόσμο καθ’ όλη τη διάρκεια της εργασίας.



Εικόνα 12. Τρισδιάστατο ρεαλιστικό αντικείμενο στο "Davy Jones' Locker".

Κάτι που με απασχόλησε ιδιαίτερα όταν ξεκίνησα να προετοιμάζομαι ήταν η χρωματική παλέτα που θα έπρεπε να χρησιμοποιήσω. Ένα σκοτεινό πειρατικό καράβι θα έπρεπε να έχει

πολλές σκοτεινές γωνίες, θα κυριαρχούσε το χρώμα του καφέ από το ξύλο και του δέρματος που χρησιμοποιούνταν συχνά σαν υλικά, καθώς και το κόκκινο της φωτιάς από τα κεριά. Ένα ακόμα χρώμα που χρησιμοποίησα ήταν το μπλε για το χρώμα του εξωτερικού χώρου που βλέπουμε από τα παράθυρα ώστε ο παίκτης να καταλαβαίνει πως βρισκόμαστε σε μία νυχτερινή ώρα. Ήθελα όμως να προσθέσω λεπτομέρειες άλλων χρωμάτων σε κάποια επιπλέον αντικείμενα. Για παράδειγμα, κατά τη διάρκεια της εξερεύνησης βρίσκουμε σε μία γωνία ένα πειρατικό σεντούκι με χρυσές λεπτομέρειες, επειδή επιθυμούσα όμως το χρώμα να χρησιμοποιείται και σαν στοιχείο σε μερικές περιπτώσεις, το κλειδί που ξεκλειδώνει το σεντούκι αυτό, έχει πάνω το ίδιο χρυσό χρώμα.



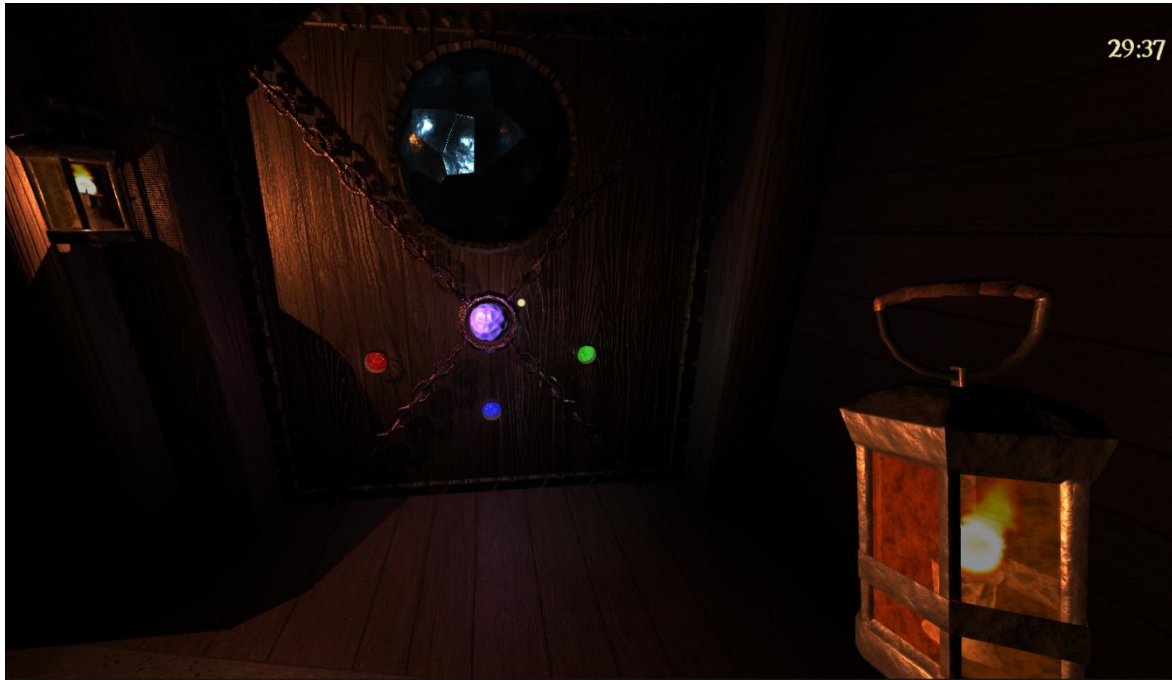
Εικόνα 13. Το σεντούκι με το κλειδί που το ανοίγει. "Davy Jones' Locker".

Με παρόμοιο τρόπο χρησιμοποίησα και κάποιες πινελιές αίματος στη σκηνή με αριθμούς και νούμερα. Η χρήση του κόκκινου και του σκούρου κόκκινου αίματος ήθελα να ελκύσουν το βλέμμα του παίκτη, ώστε να συνειδητοποιήσει πως τα στοιχεία αυτά είναι άμεσα συνδεδεμένα με τους γρίφους. Άλλωστε, η χρήση του αίματος σε μονοπάτια έχει χρησιμοποιηθεί σε πολλά βιντεοπαιχνίδια, με τον σκοπό να δείξει ότι κάτι έχει συμβεί προηγουμένως στην σκηνή, να προετοιμάσει τον παίκτη για κάτι που πρόκειται να συμβεί ή ακόμα και για να τον οδηγήσει στο σωστό μονοπάτι. Αντίστοιχα σε Δωμάτια Διαφυγής (Escape Rooms) μπορούμε να δούμε σχέδια στους τοίχους από ψεύτικο αίμα, κάτι που χρησιμοποιείται πολύ συχνά σε γρίφους.



Εικόνα 14. Σκούρο κόκκινο σημάδι από αίμα. "Davy Jones' Locker".

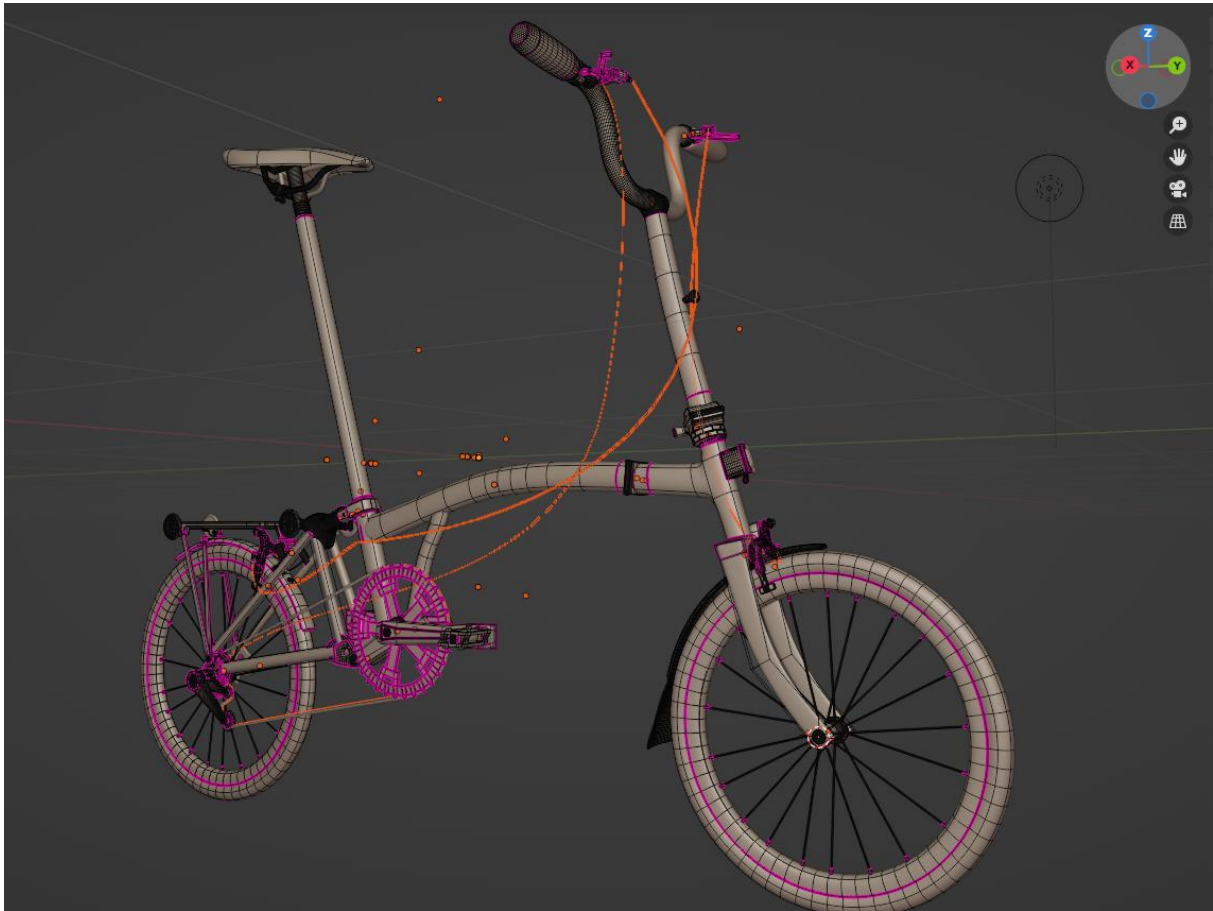
Το στοιχείο της φαντασίας στο “Davy Jones’ Locker”, είναι η τελική πόρτα με διάφορους κρυστάλλους πάνω της, οι οποίοι είναι κομμάτι ενός γρίφου που πρέπει να λυθεί για να καταφέρει ο παίκτης να διαφύγει από το καράβι. Τα ξένα φωτεινά χρώματα του μπλέ, του κόκκινου και του πράσινου ήθελα να συμβολίζουν πως πίσω από αυτή την πόρτα υπάρχει κάτι διαφορετικό. Ήθελα να υπάρχει η απορία του τι συμβαίνει όταν ο παίκτης ανοίγει αυτήν την πύλη και ολοκληρώνει την εμπειρία του παιχνιδιού.



Εικόνα 15. Η τελευταία πόρτα με τους κρύσταλλους, "Davy Jones' Locker".

2.2 Μοντελισμός και πολύγωνα.

Τα περισσότερα λογισμικά που χρησιμοποιούνται για τη δημιουργία τρισδιάστατων αντικειμένων έχουν παρόμοιες αρχές. Όταν μάθει κανείς τη λογική πίσω από τα είδη μοντελισμού, πλέον το να κατανοήσει ένα καινούριο πρόγραμμα είναι θέμα χρόνου και συνήθειας. Τα δύο είδη μοντελισμού είναι τα “hard surface” και “organic”. Hard surface περιλαμβάνει τον μοντελισμό αντικειμένων με λείες και στατικές επιφάνειες που πολύ συχνά χρησιμοποιούνται και εργαλεία που κρατάνε το αντικείμενο ομοιόμορφο από όλους τους άξονες, παρόμοιο με ένα αντικείμενο που έχει δημιουργηθεί από εργοστάσιο. Ενώ αντιθέτως, organic σημαίνει η μοντελοποίηση αντικειμένων που είναι πιο ελεύθερα χωρίς περιορισμούς. Η δημιουργία ενός ρομπότ ή μίας μηχανικής πόρτας ακολουθεί τα θεμέλια του hard surface μοντελισμού, ενώ η δημιουργία ενός μήλου ή ενός ανθρώπινου σώματος, ενώ ακολουθούν ένα γενικό σχήμα ανατομίας του αντικειμένου, είναι ελεύθερο χωρίς πολλούς περιορισμούς για τη υλοποίηση ενός σωστού αποτελέσματος.



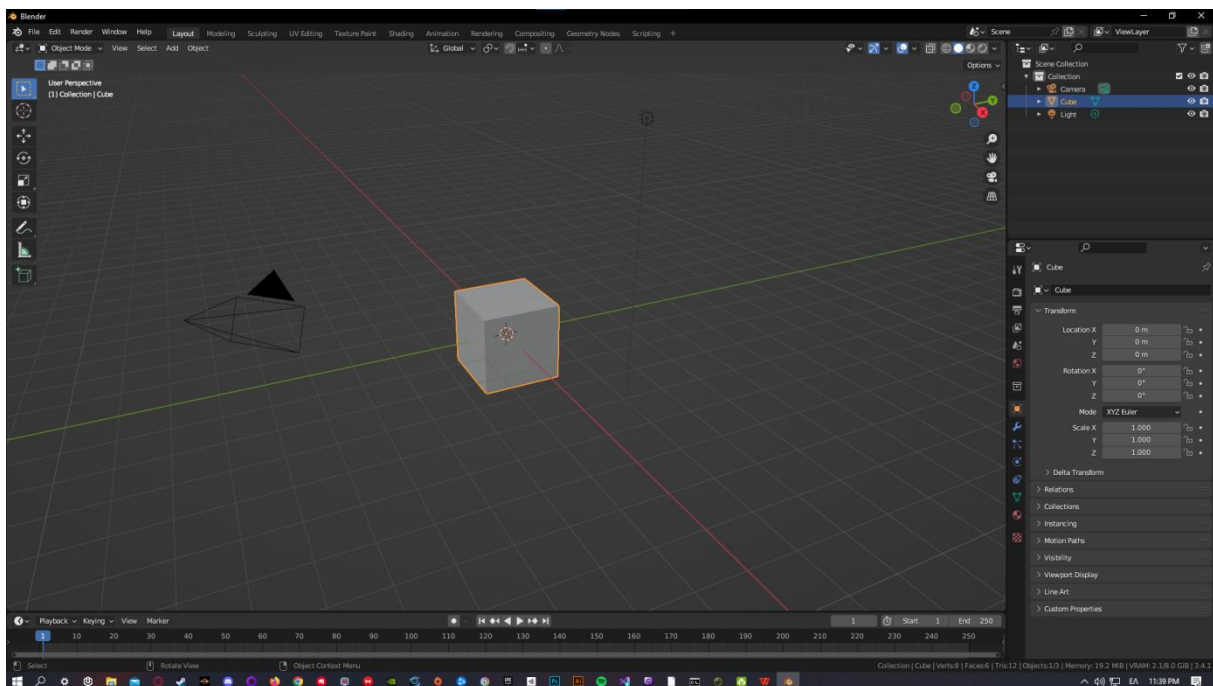
Εικόνα 16. Hard surface μοντελοποίηση ποδηλάτου στο λογισμικό Blender 3D.



Εικόνα 17. Organic μοντελοποίηση ενός τέραςτος από το στούντιο 3D-Ace.

Ενώ τα τυπικά λογισμικά μπορούν να χρησιμοποιηθούν για οποιαδήποτε εργασία σε hard surface και organic μοντελοποίησης, υπάρχουν λογισμικά που προσπαθούν να μιμηθούν την

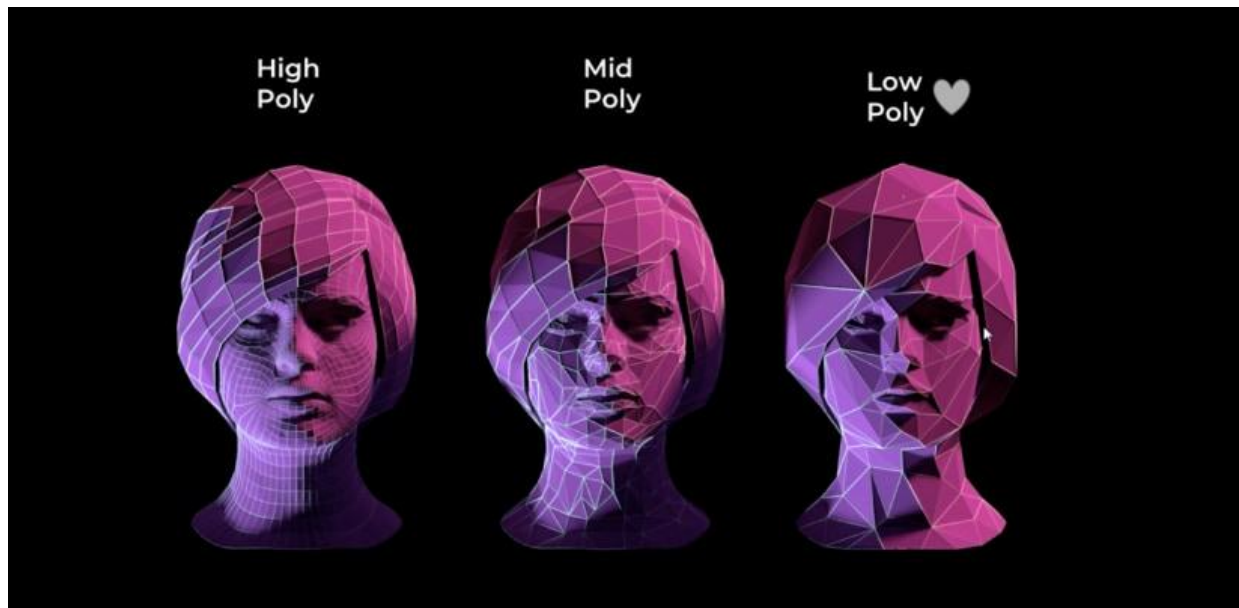
τέχνη της γλυπτικής σε ψηφιακή μορφή. Αυτά τα λογισμικά κατά κύριο λόγο χρησιμοποιούνται για την δημιουργία organic τύπου μοντέλων, ενώ μερικοί τα προσθέτουν στη ροή εργασίας τους ακόμη και στον hard surface μοντελισμό για να αποδώσουν κάποιες επιπλέον λεπτομέρειες στα μοντέλα. Τα πιο γνωστά λογισμικά κλασσικού τύπου μοντελισμού που χρησιμοποιούνται εδώ και πολλά χρόνια είναι το 3DS Max και το Maya 3D της εταιρίας Autodesk, καθώς το πιο διαδεδομένο πρόγραμμα ψηφιακής γλυπτικής είναι το ZBrush της εταιρίας Pixologic. Το κόστος των παραπάνω λογισμικών είναι πολύ υψηλό και τα καθιστά απρόσιτα σε ανεξάρτητους δημιουργούς και μικρά στούντιο, γεγονός που έχει αναγκάσει να εμφανιστεί στο προσκήνιο ένα επιπλέον πρόγραμμα τρισδιάστατων γραφικών, το Blender 3D. Το Blender 3D είναι ένα ανοιχτού κώδικα λογισμικό με ενσωματωμένα εργαλεία για ψηφιακή γλυπτική, καθώς και πολλών ακόμα εργασιών που στην πραγματικότητα χρειάζονται επιπλέον λογισμικά για να εξασφαλιστεί ένα ολοκληρωμένο αποτέλεσμα. Με λίγα λόγια είναι ένα πολύ ικανό εργαλείο που λύνει τα χέρια πολλών δημιουργών για να μπορέσουν να εισέλθουν στον κόσμο της ανάπτυξης βιντεοπαιχνιδιών με μηδαμινό κόστος, για τους παραπάνω λόγους, ενσωμάτωσα στη ροή εργασιών μου το Blender 3D.



Εικόνα 18. Η αρχική διεπαφή χρήστη του λογισμικού Blender 3D.

Αφού εξηγήσαμε ποια είναι τα είδη μοντελοποίησης παραπάνω, τώρα θα αναλύσουμε και τι συμβαίνει κατά τη διάρκεια αυτή. Τα τρισδιάστατα μοντέλα αποτελούνται από διάφορα πολύγωνα, μοντελοποίηση εννοούμε τον χειρισμό και πολλαπλασιασμό αυτών των

πολυγώνων με σκοπό τη πλάση τους σε ένα ολοκληρωμένο τρισδιάστατο μοντέλο μέσα στον χώρο του λογισμικού. Από όσα περισσότερα πολύγωνα αποτελείται ένα τρισδιάστατο αντικείμενο, θεωρητικά τόσο πιο ομαλές θα είναι οι επιφάνειες του χωρίς σκληρές γωνίες. Όταν ένα παιχνίδι αποτελείται από μοντέλα με χαμηλό αριθμό πολυγώνων, τα αποκαλούμε ως “low poly” ή “low polygon mesh”. Αντίστοιχα τα παιχνίδια που στοχεύουν στην δημιουργία πιο ομαλών επιφανειών και χρησιμοποιούν μοντέλα με μεγάλο αριθμό πολυγώνων, λέγονται “high poly” ή “high polygon mesh”. Γνωρίζοντας αυτές τις δύο κατηγορίες, κάποιος μπορεί να αναρωτηθεί για ποιο λόγο κάποιος να επιλέξει να φτιάξει ένα βιντεοπαιχνίδι με low poly μοντέλα. Υπάρχουν δύο λόγοι που μπορούν να απαντήσουν αυτή την ερώτηση. Πρώτον τα πολύγωνα επηρεάζουν την επίδοση του παιχνιδιού στα συστήματα βιντεοπαιχνιδιών, με αποτέλεσμα κάποια αδύναμα συστήματα να μην μπορούν να εκτελέσουν το αρχείο σε ικανοποιητικό βαθμό για τον χρήστη. Για αυτόν τον λόγο μερικά στούντιο στοχεύουν στο να είναι διαθέσιμο το βιντεοπαιχνίδι τους σε περισσότερα συστήματα από το να ακολουθήσουν μία αισθητική πολλών πολυγώνων. Οι πρώτοι τρισδιάστατοι τίτλοι ήταν με τα λιγότερα δυνατά πολύγωνα για αυτόν ακριβώς τον λόγο, η τεχνολογία των προσωπικών υπολογιστών δεν είχε τη δυνατότητα του να αντέξει περισσότερα πολύγωνα. Το ίδιο ακριβώς συμβαίνει ακόμα με τα βιντεοπαιχνίδια που αναπτύσσονται και απευθύνονται σε μικρές φορητές κονσόλες όπως τα κινητά τηλέφωνα. Ο δεύτερος λόγος που κάποιος μπορεί να επιλέξει τα low poly γραφικά είναι ο λόγος της αισθητικής τους. Η αισθητική αυτών των παιχνιδιών είναι κατά βάση στυλιζαρισμένη, ώστε τα πολύγωνα έρχονται δεύτερα στον τομέα της ομορφιάς, καθώς πρώτο είναι το στυλ των υφών που έχουν χρησιμοποιηθεί. Υπάρχουν πολλοί άνθρωποι που προτιμούν τους τίτλους με ξεχωριστά γραφικά από τους τίτλους που είναι φωτορεαλιστικοί και εκθαμβωτικοί. Ένας μεγάλος αριθμός τίτλων προσπαθούν να πετύχουν κάτι ενδιάμεσο. Όσο το δυνατόν λιγότερα πολύγωνα χωρίς αυτό να φαίνεται στο τελικό αποτέλεσμα. Αυτή η τακτική προσφέρει καλές επιδόσεις ταυτόχρονα με ομαλές επιφάνειες στα μοντέλα και είναι η τακτική που ακολούθησα εγώ στο παιχνίδι της πτυχιακής εργασίας.



Εικόνα 19. Παράδειγμα των τριών επιπέδων πολυγώνων από τη σελίδα domestika.org.

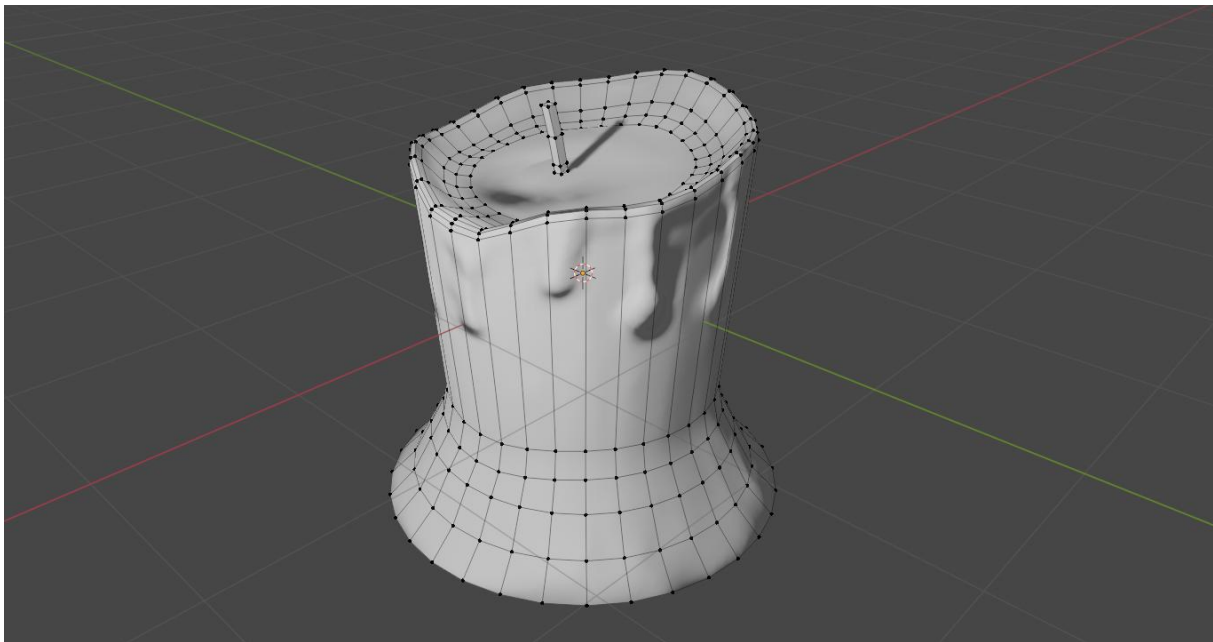


Εικόνα 20. Πολύγωνα σε πιστόλι μέσα στο "Davy Jones' Locker".



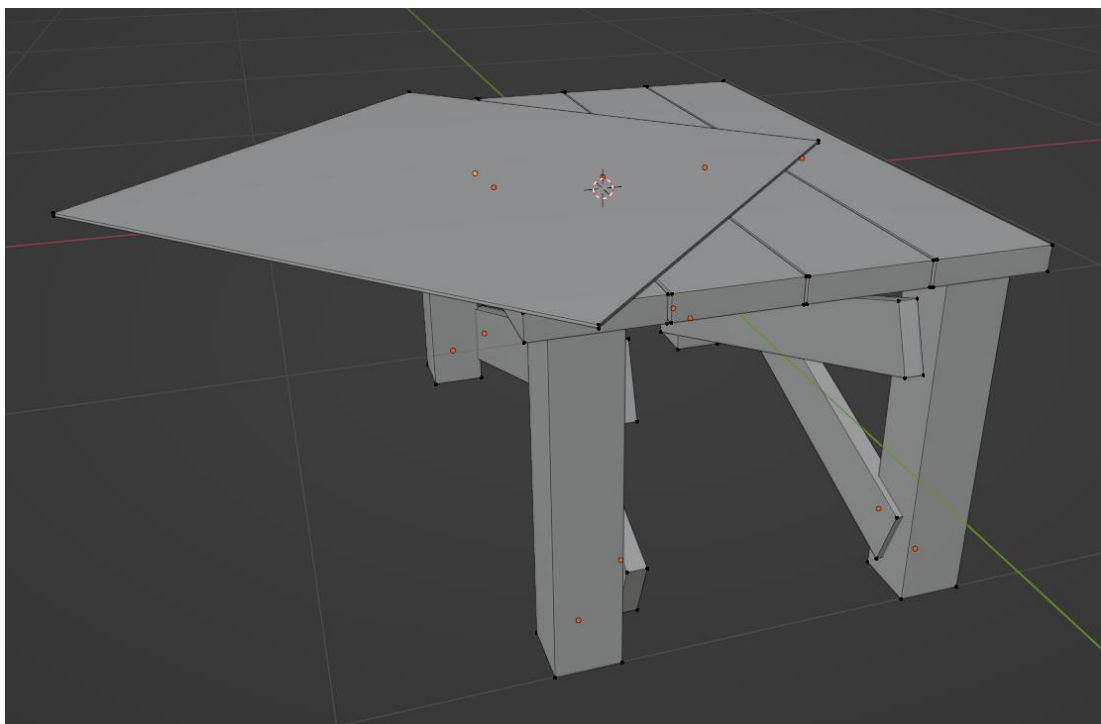
Εικόνα 21. "Tomb Raider" 1996 ενάντια του 2017.

Η αναλογία των πολυγώνων στα μοντέλα σε μία σκηνή μπορεί όμως να διαφέρει. Όταν ένα αντικείμενο είναι το επίκεντρο της προσοχής του παίκτη, κατά κύριο λόγο το μοντέλο του θα είναι φτιαγμένο με περισσότερη προσοχή στη λεπτομέρεια. Αντίστοιχα εάν κάποια αντικείμενα υπάρχουν μόνο για να γεμίσουν τη σκηνή ώστε να μην φαίνεται άδειο το τοπίο, αυτά τα αντικείμενα που δεν δημιουργούν άμεση διάδραση με τον παίκτη αποτελούνται από τα ελάχιστα δυνατά πολύγωνα. Παραδείγματος χάρη το πιστόλι στην παραπάνω εικόνα έχει έναν αριθμό πολυγώνων που δίνει την αίσθηση του ομαλού επίπεδου του μοντέλου, ενώ στην παρακάτω εικόνα του κεριού, έχουν χρησιμοποιηθεί πολύ λιγότερα πολύγωνα δίνοντας τις λεπτομέρειες στο μοντέλο μέσω της εικόνας της υφής του και όχι μέσω γεωμετρίας. Τη διαδικασία αυτή θα την αναπτύξουμε περισσότερο στο κεφάλαιο “Υφές - Textures”. Προς το παρόν, στη συνέχεια θα εξηγηθεί με περισσότερες λεπτομέρειες η διαδικασία ροής εργασιών που ακολουθήθηκε για την ανάπτυξη όλων των τρισδιάστατων μοντέλων.



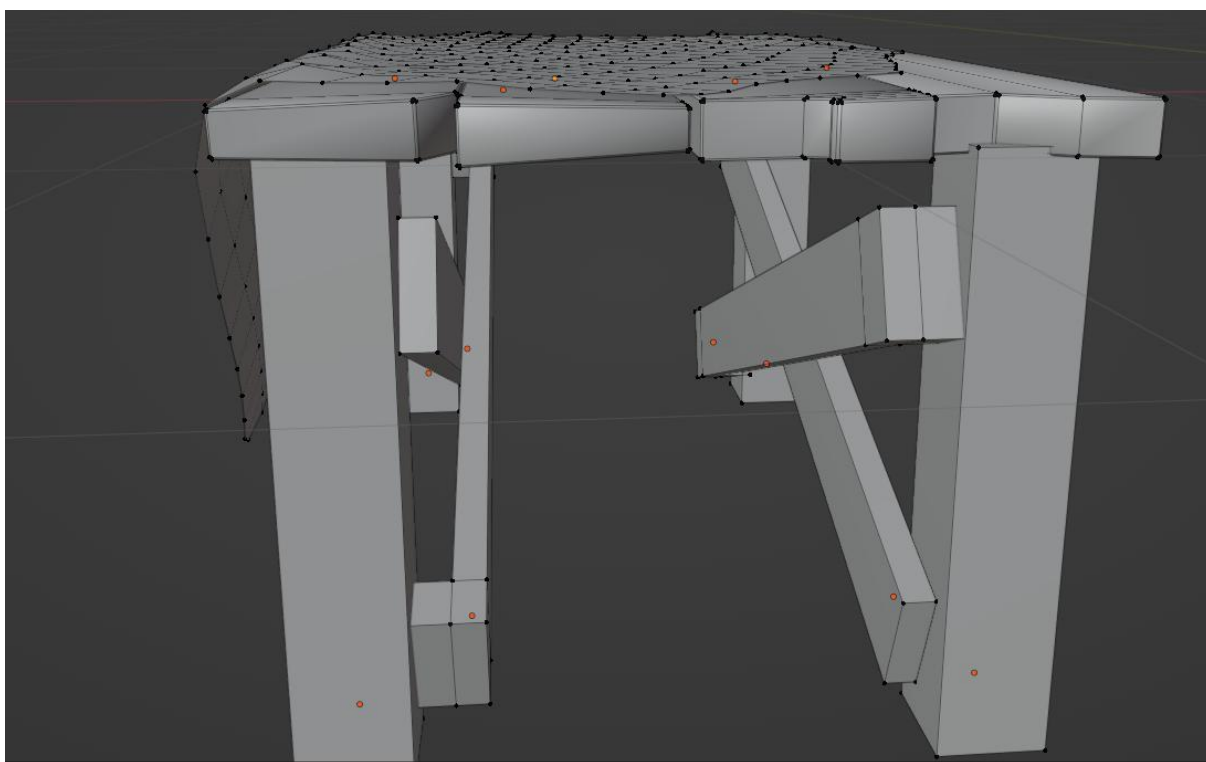
Εικόνα 22. Κερί για το "Davy Jones' Locker".

Πριν ξεκινήσω τη μοντελοποίηση του αντικειμένου που πρόκειται να ασχοληθώ στην προκειμένη περίπτωση, η πρώτη ερώτηση που κάνω στον εαυτό μου είναι το εάν θέλω το αντικείμενο να είναι κομμάτι της διάδρασης με τον παίκτη. Όταν απαντήσω αυτή την ερώτηση, συνεχίζω στην υλοποίηση του μοντέλου. Στην αρχή, φτιάχνω ένα πρόχειρο τρισδιάστατο μοντέλο χωρίς λεπτομέρειες με λίγα πολύγωνα.



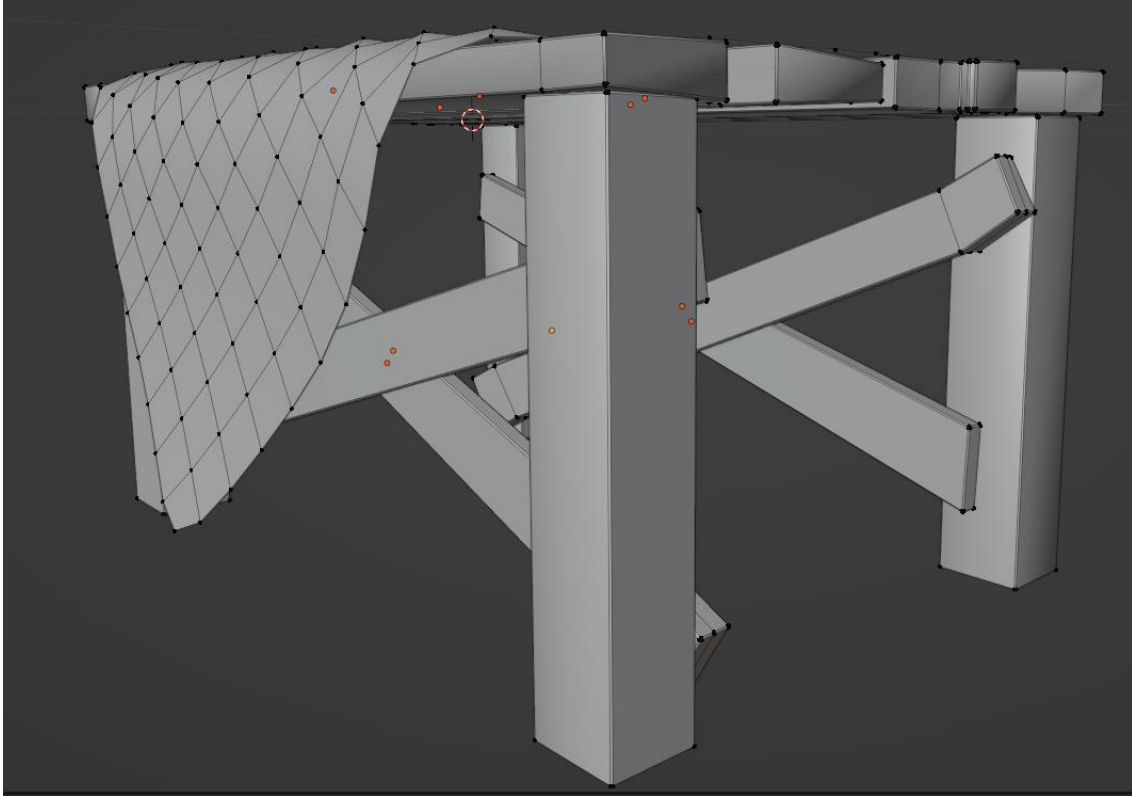
Εικόνα 23. Πρόχειρη γεωμετρία τρισδιάστατου μοντέλου.

Στη συνέχεια προσθέτω κάποιες πρωταρχικές εμφανής λεπτομέρειες που καθορίζουν περισσότερο το σχήμα του αντικειμένου. Στην προκειμένη περίπτωση, κόβω μεγάλα κομμάτια των σανίδων για να δημιουργήσω περισσότερη ποικιλομορφία στο μοντέλο. Μετά θέλω να γυρίσω τις σανίδες λίγο στον έναν άξονα για να μην είναι όλες επίπεδες στην ίδια ευθεία και προσθέτω λίγη γεωμετρία κάτω από το τραπέζι ώστε να βγάζει περισσότερο νόημα η τοποθέτηση τους σε χιαστί. Σε αυτό το στάδιο προσθέτω γεωμετρία και στο αντικείμενο πάνω από τις σανίδες και του δίνω σχήμα να πέφτει στα πλάγια του τραπεζιού σαν πανί, καθώς αυτό προορίζεται για να γίνει ένα πανί δέρματος.

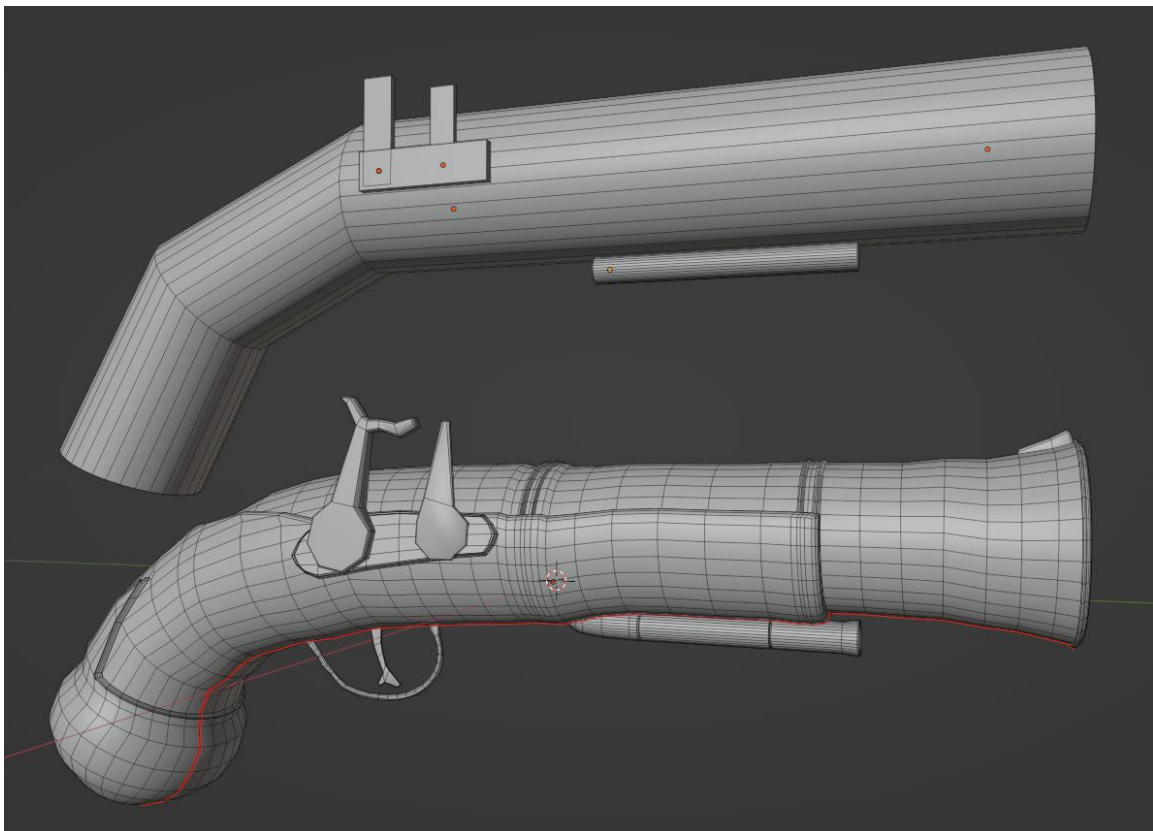


Εικόνα 24. Δεύτερη φάση της μοντελοποίησης.

Τέλος, συνήθως προσθέτω επιπλέον μικρές λεπτομέρειες για να δοθεί λίγο περισσότερη φυσικότητα στο μοντέλο, εάν όμως προσθέσω μεγάλο αριθμό τέτοιων λεπτομερειών, προσπαθώ να τον μεταφράσω σε γεωμετρία μέσω των υφών - textures, κάτι που θα επεξηγήσουμε στο επόμενο κεφάλαιο πως λειτουργεί και τη διαδικασία που ακολουθούμε. Στην προκειμένη περίπτωση ήθελα απλά να εξαφανίσω τις γωνίες ενενήντα μοιρών, καθώς παρατηρώντας τον πραγματικό κόσμο τίποτα δεν έχει πραγματικά απόλυτες γωνίες, πράμα που αν το δούμε σε ένα βιντεοπαιχνίδι επηρεάζει αρνητικά το υποσυνείδητο του χρήστη χωρίς να το καταλαβαίνει.



Εικόνα 25. Τελευταίο βήμα της μοντελοποίησης του τραπεζιού.



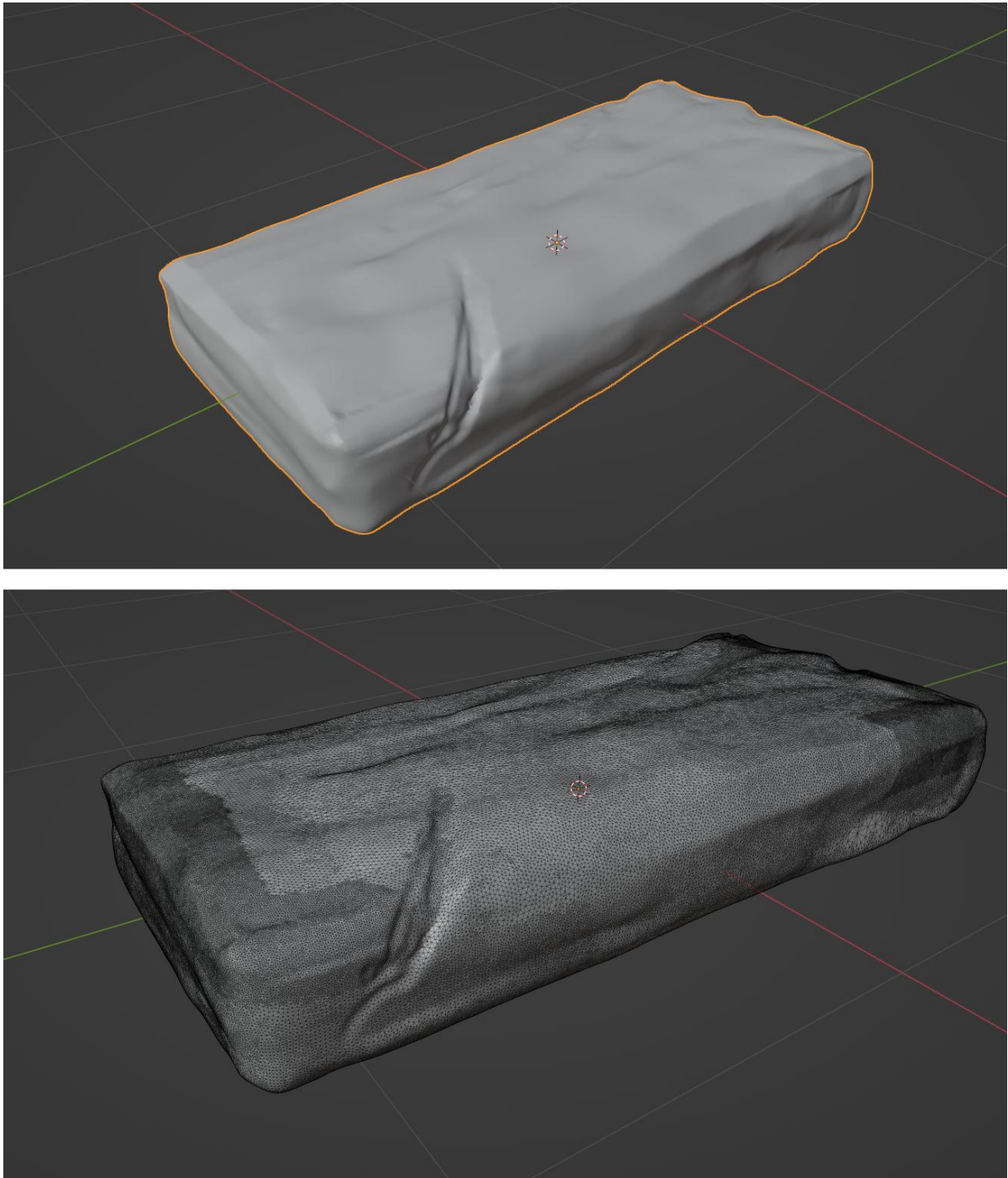
Εικόνα 26. Το πρωταρχικό με το τελικό στάδιο από το πιστόλι.



Εικόνα 27. Έπιπλο με διάφορες επιπλέον μικρές λεπτομέρειες - μαχαιριές εφαρμοσμένες στη γεωμετρία του μοντέλου.

Όσο αναφορά την ψηφιακή γλυπτική και την εφαρμογή της σε ένα μοντέλο βιντεοπαιχνιδιού, είναι λίγο πιο περίπλοκη. Για αρχή θα δούμε ένα παράδειγμα του προβλήματος που δημιουργείται χρησιμοποιώντας τη τεχνική της γλυπτικής και την λύση του. Χρησιμοποιώντας τα εργαλεία της ψηφιακής γλυπτικής δίνεται η ελευθερία της μοντελοποίησης ενός αντικειμένου σε πολύ γρήγορους ρυθμούς με πολλές λεπτομέρειες θυσιάζοντας όμως τις επιδόσεις του βιντεοπαιχνιδιού. Όταν ένα μοντέλο επεξεργασθεί ψηφιακά με γλυπτική, δημιουργούνται χιλιάδες επιπλέον πολύγωνα στην επιφάνεια, πράμα που καθιστά το αντικείμενο άχρηστο στην πρωταρχική του αυτή μορφή. Για τον παραπάνω λόγο, προχωράμε στη συνέχεια σε μία τεχνική που λέγεται “retopology”. Με αυτή την τεχνική δημιουργούμε ένα low poly αντικείμενο χονδρικά στο σχήμα του high poly αντικειμένου και συνεχίζουμε στην χρήση του high poly αντικειμένου με σκοπό να δώσουμε

ψεύτικη γεωμετρία στο low poly αντικείμενο. Θα εμβαθύνουμε στην τεχνική αυτή στο επόμενο κεφάλαιο “Υφές - Textures”.



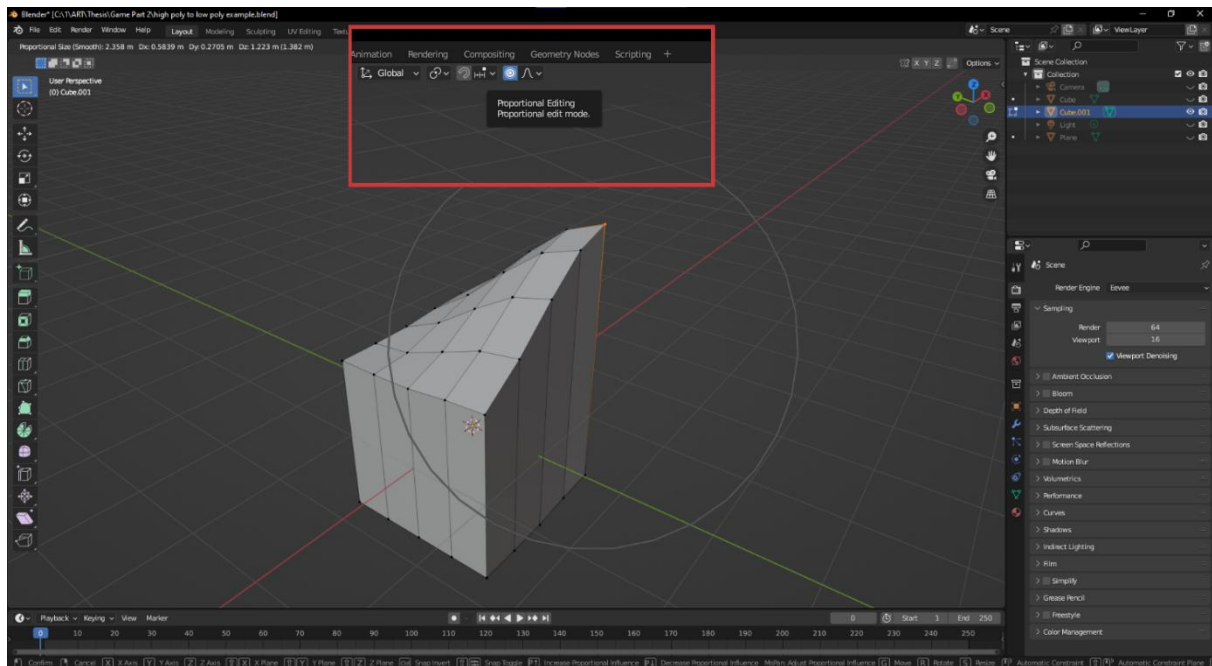
Εικόνα 28. Παράδειγμα εφαρμογής της ψηφιακής γλυπτικής σε ένα τούβλο για τη δημιουργία λεπτομερειών, με αποτέλεσμα πάνω από εκατό χιλιάδες πολύγωνα.

Έχοντας τις παραπάνω γενικές γνώσεις, το επόμενο βήμα είναι η κατανόηση των εργαλείων μέσα στα λογισμικά μοντελισμού. Γνωρίζοντας ελάχιστα βασικά εργαλεία, μπορεί ο καθένας

με υπομονή να μοντελοποιήσει τα θεμέλια ενός αντικειμένου που στη συνέχεια μπορεί να προχωρήσει στην τελική μορφή του. Ονομαστικά τα βασικά εργαλεία είναι τα εξής. Τα εργαλεία μετασχηματισμού, με σκοπό την μετακίνηση του αντικειμένου και των πολυγώνων του. Τα εργαλεία επέκτασης με σκοπό την επέκταση και τη γέννηση νέων πολυγώνων. Το εργαλείο ένωσης με σκοπό την ένωση πολυγώνων. Το εργαλείο αντιστροφής με σκοπό της δημιουργίας πολυγώνων προς το εσωτερικό του επιλεγμένου πολυγώνου. Και το εργαλείο περικοπής με σκοπό τον χωρισμό ήδη υπάρχοντων πολυγώνων⁸. Αρχικά, μέσα στο λογισμικό του Blender 3D, πατώντας το κουμπί συντόμευσης “Tab”, μπαίνει το πρόγραμμα σε λειτουργία επεξεργασίας, σε αυτή την κατάσταση φαίνονται τα πολύγωνα του αντικειμένου και μπορούν να υποστούν επεξεργασία.

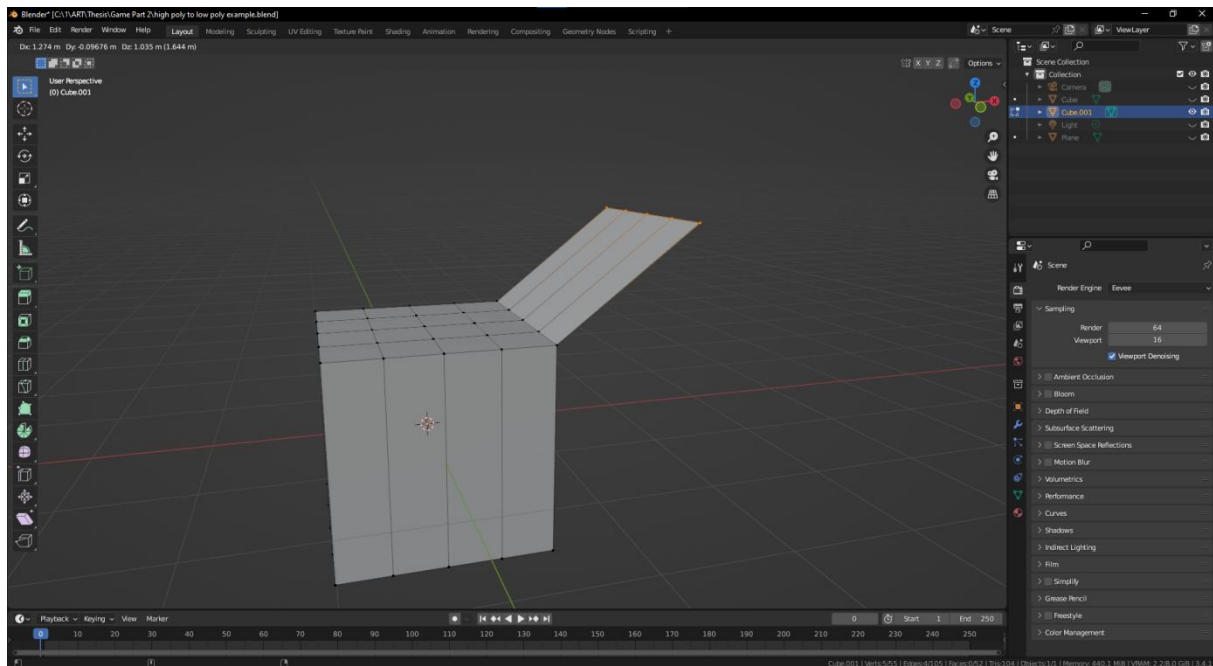
Τα εργαλεία μετασχηματισμού είναι τρία. Της κίνησης (με το κουμπί συντόμευσης “G”), της περιστροφής (με το κουμπί συντόμευσης “R”) και της μεγέθυνσης (με το κουμπί συντόμευσης “S”). Τα εργαλεία αυτά είναι τα πιο βασικά εργαλεία για την μεταβολή του μοντέλου μέσα στον χώρο, καθώς μπορούν να τα χρησιμοποιηθούν για την μεταβολή ολόκληρου του αντικειμένου ή και μόνο ενός αριθμού πολυγώνων που έχουν επιλεγθεί από τον χρήστη. Επιπλέον σημαντικό εργαλείο είναι και η χρήση αναλογικής επεξεργασίας, με σκοπό να επηρεάζονται πολλαπλά παρεμφερή πολύγωνα με την κίνηση ενός μόνο πολυγώνου.

⁸ *Modeling — Blender Manual*. (n.d.-b). <https://docs.blender.org/manual/en/latest/modeling/index.html>



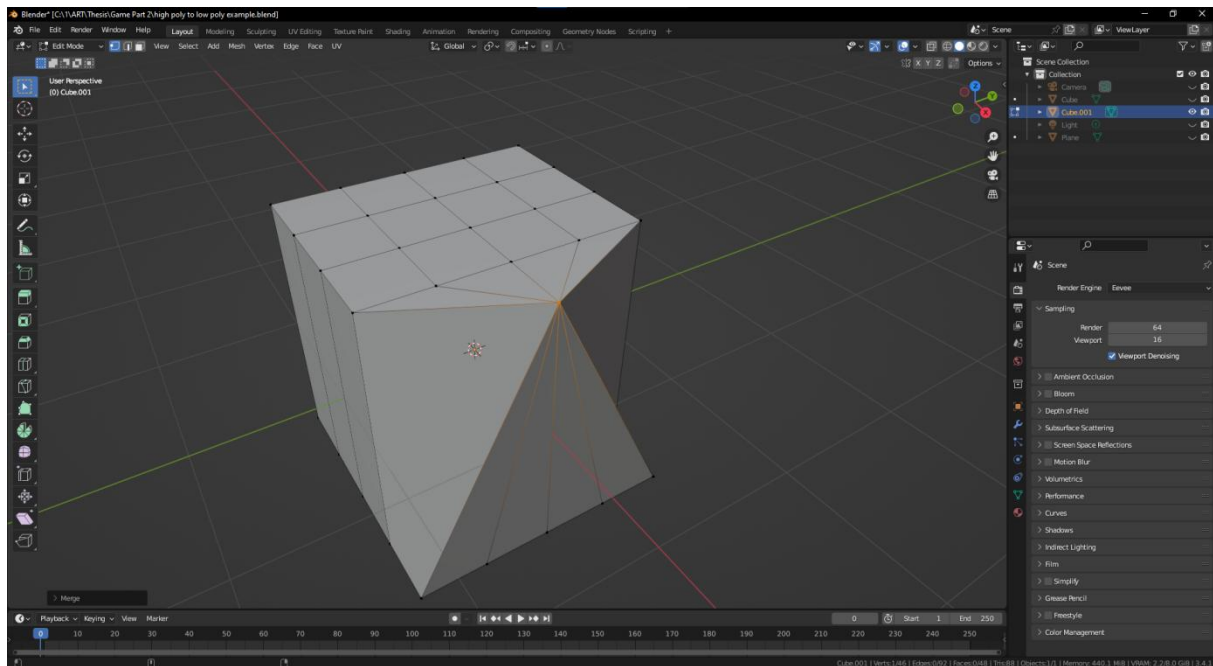
Εικόνα 29. Το κουμπί της αναλογικής επεξεργασίας στο λογισμικό Blender 3D.

Τα εργαλεία επέκτασης λειτουργούν με σκοπό τη δημιουργία καινούριας γεωμετρίας επεκτείνοντας τα ήδη υπάρχοντα πολύγωνα. Επιλέγοντας πρώτα τα πολύγωνα στην λειτουργία επεξεργασίας, πατώντας το κουμπί συντόμευσης “E” και μετακινώντας το ποντίκι στον χώρο, όταν είμαστε σίγουροι για το σχήμα της νέας γεωμετρίας, με το αριστερό κλικ επιβεβαιώνουμε την κίνηση αυτή. Αυτό το εργαλείο σε συνδυασμό με τα εργαλεία μετασχηματισμού χρησιμοποιούνται καθ’ όλη τη διάρκεια της μοντελοποίησης.



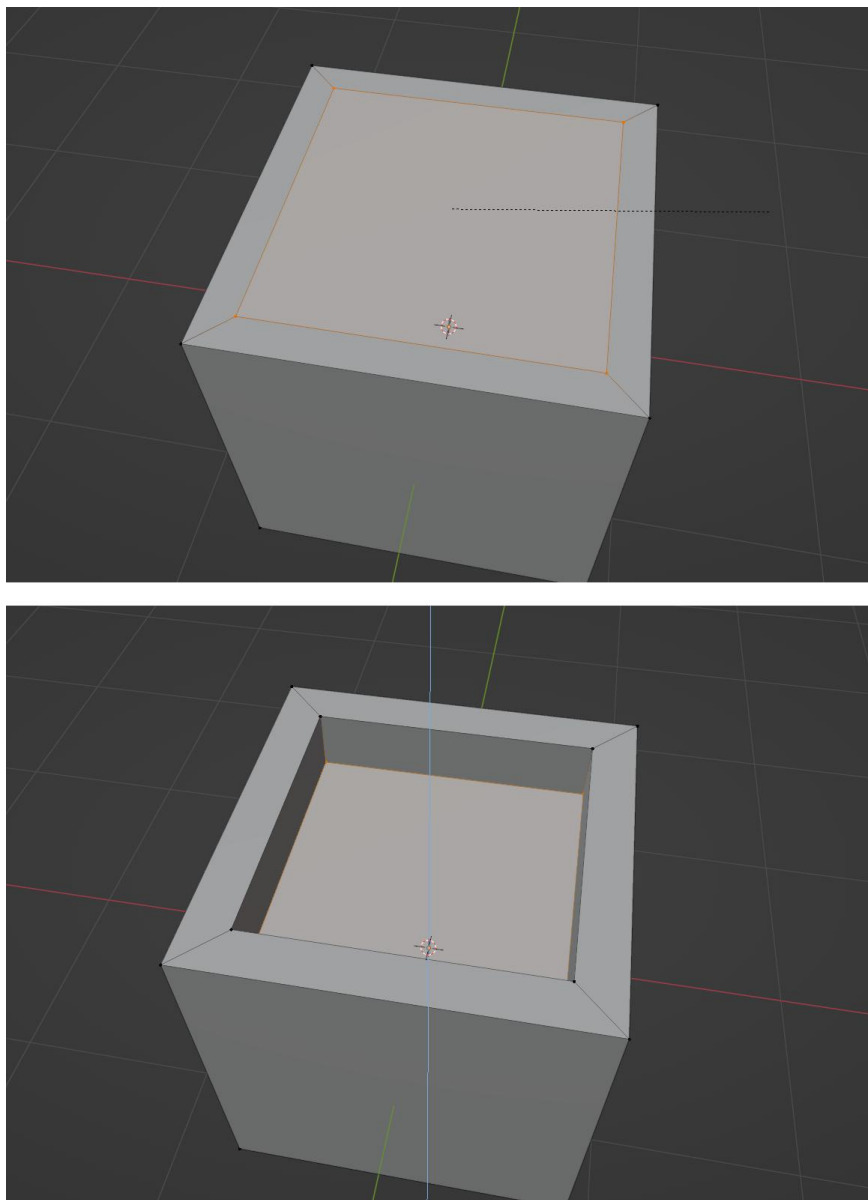
Εικόνα 30. Το εργαλείο επέκτασης εν ώρα λειτουργίας.

Το εργαλείο ένωσης έχει ως σκοπό την ένωση πολυγώνων που δεν χρειάζονται, τον καθαρισμό του αντικειμένου από τον μεγάλο αριθμό πολυγώνων ή τη δημιουργία γωνιών. Βασικός κανόνας στην τρισδιάστατη μοντελοποίηση είναι η συντήρηση τεσσάρων πλευρών σε όλα τα πολύγωνα, τουλάχιστον στα πρωταρχικά στάδια της επεξεργασίας. Καθώς εάν δημιουργηθούν πολύγωνα με τρεις πλευρές, στη συνέχεια γεννιούνται πολλά προβλήματα στη διαδικασία, αναγκάζοντας συχνά τον καλλιτέχνη στην αναδημιουργία του αντικειμένου από την αρχή. Επομένως, το εργαλείο αυτό πρέπει να χρησιμοποιείται με προσοχή, καθώς εάν ενωθούν πολύγωνα και δημιουργηθούν γωνίες, θα ξεκινήσει ένας ειρμός προβλημάτων. Το πλήκτρο συντόμευσης του εργαλείου είναι το “M” αφού έχουν ήδη επιλεγεί τα πολύγωνα.



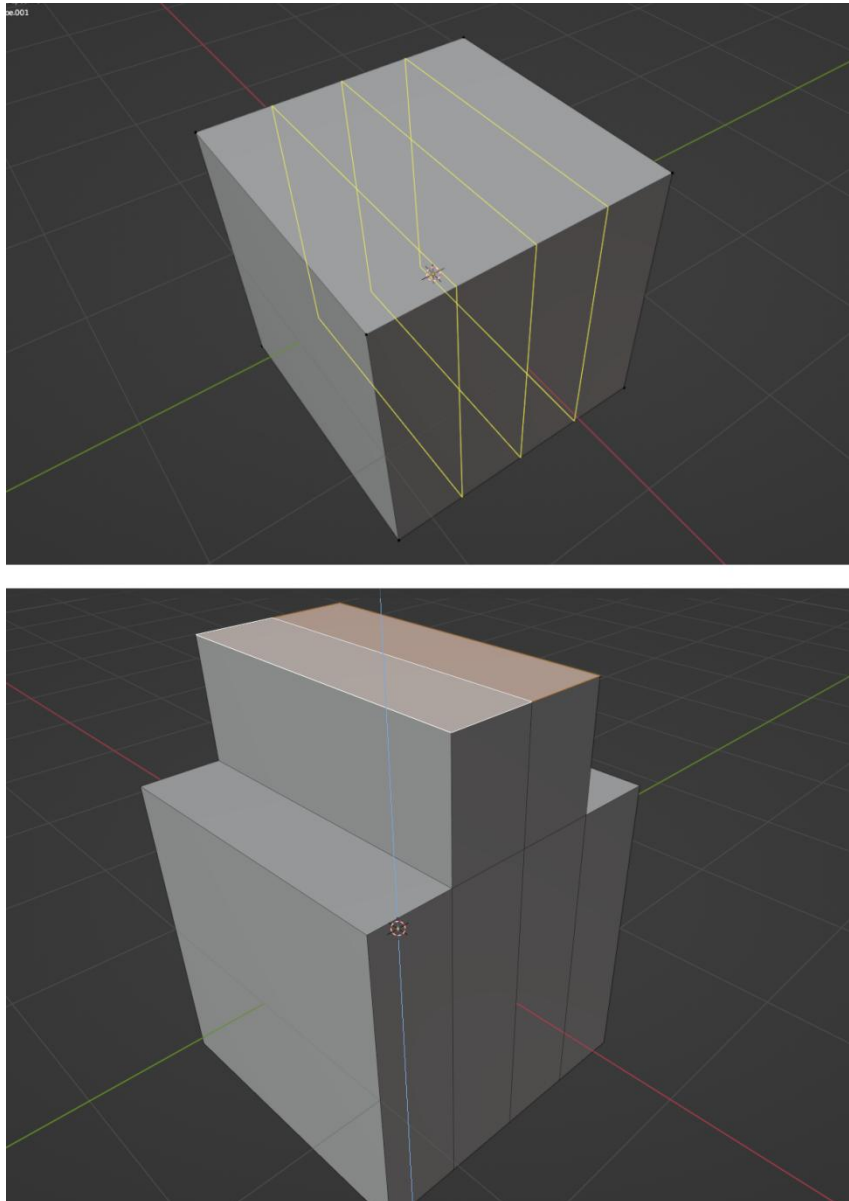
Εικόνα 31. Εργαλείο ένωσης και αλλαγή πολυγώνων σε τρεις πλευρές.

Το εργαλείο αντιστροφής χρησιμοποιείται συνήθως για την δημιουργία και στη συνέχεια επεξεργασία των πολυγώνων στο εσωτερικό. Για παράδειγμα με αυτό το εργαλείο μπορεί να δημιουργηθεί κάποιο παράθυρο ή ένα πλαίσιο αναλόγως την περίπτωση. Είναι ένα από τα πιο συνηθισμένα εργαλεία και χρησιμοποιείται συνεχώς στη ροή εργασιών όλων των καλλιτεχνών του τομέα. Το κουμπί συντόμευσης του είναι το “I”, εφόσον έχει ήδη επιλεγθεί την πλευρά που στοχεύεται προς επεξεργασία.



Εικόνα 32. Εργαλείο αντιστροφής με συνέχεια το εργαλείο επέκτασης.

Το εργαλείο περικοπής χωρίζει τα πολύγωνα σε πολλαπλές ενότητες. Χρησιμοποιείται συνήθως για τον πολλαπλασιασμό πολυγώνων συνδεδεμένων σε σειρά με σκοπό την επεξεργασία τους στη συνέχεια. Η χρήση του γίνεται με τα κουμπιά συντόμευσης “Ctrl + R”, έχοντας το ποντίκι πάνω στη σειρά πολυγώνων που χρειάζονται περικοπή.



Εικόνα 33. Εργαλείο περικοπής με συνέχεια το εργαλείο επέκτασης.

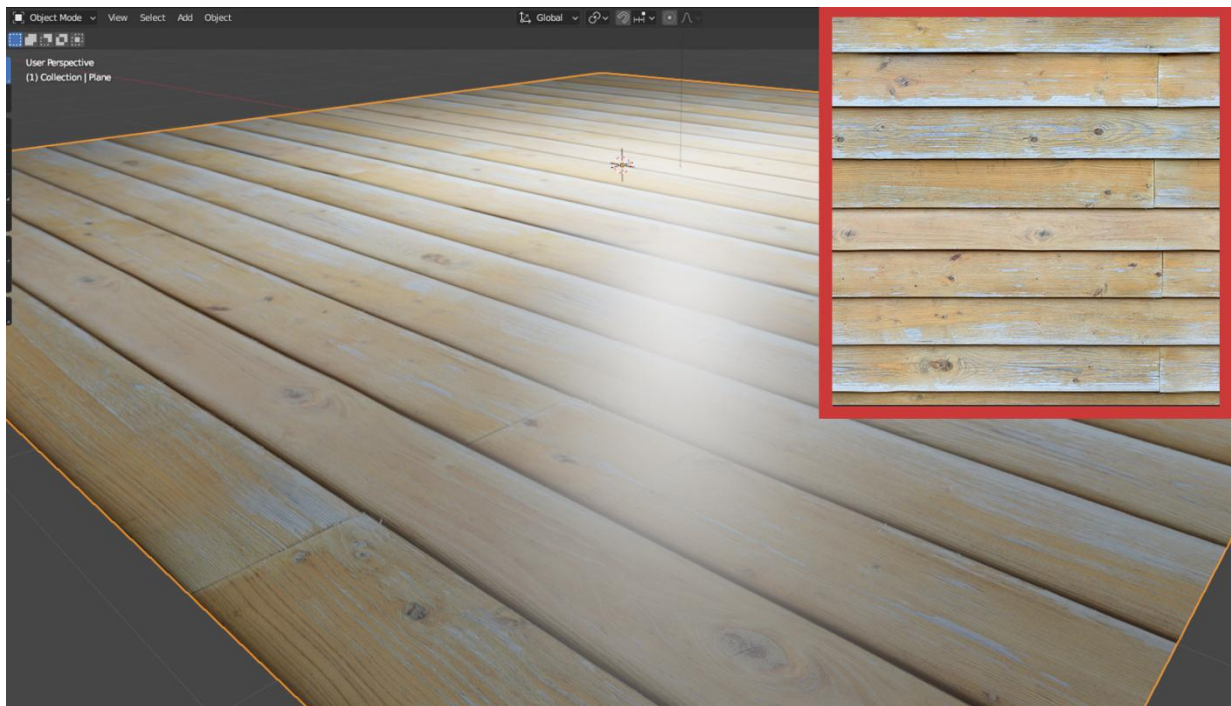
Σε όλα αυτά τα λογισμικά υπάρχουν πολλά δεκάδες ακόμη απαραίτητα εργαλεία. Οποιοδήποτε μοντέλο μπορεί να δημιουργηθεί με πολλαπλούς διαφορετικούς τρόπους, εάν το αποτέλεσμα είναι ικανοποιητικό, τίποτα δεν θεωρείται απόλυτα λάθος.

2.3 Υφές - Textures.

Όταν ο καλλιτέχνης έχει ολοκληρώσει το τρισδιάστατο μοντέλο, είναι πλέον το στάδιο του ντυσίματος του με υλικά, γνωστά και ως “materials”. Τα υλικά αποτελούνται από διάφορες υφές, γνωστά στα και ως “textures”. Οι εικόνες αυτές δίνουν διαφόρων ειδών πληροφορίες στο λογισμικό δημιουργίας τρισδιάστατων γραφικών και στη μηχανή βιντεοπαιχνιδιών. Η

ροή εργασιών με αυτές τις εικόνες λέγεται “Physically Based Rendering (PBR)”, και οι εικόνες χωρίζονται σε πέντε είδη. Το πρώτο είδος είναι γνωστό ως “diffuse map” ή “base color map”, το δεύτερο ως “metalness map”, το τρίτο ως “normal map”, το τέταρτο ως “height map” και το τελευταίο ως “roughness map” ή “smoothness”. Τα είδη αυτά μπορούν να εξαχθούν μέσω διαφόρων λογισμικών, το πιο γνωστό με όνομα “Substance Designer”, καθώς όμως για την πτυχιακή χρησιμοποιήθηκε το δωρεάν λογισμικό “Materialize”, καθώς και για τη δημιουργία κάποιων διαδικαστικά δημιουργημένων υφών μέσω του Blender 3D.

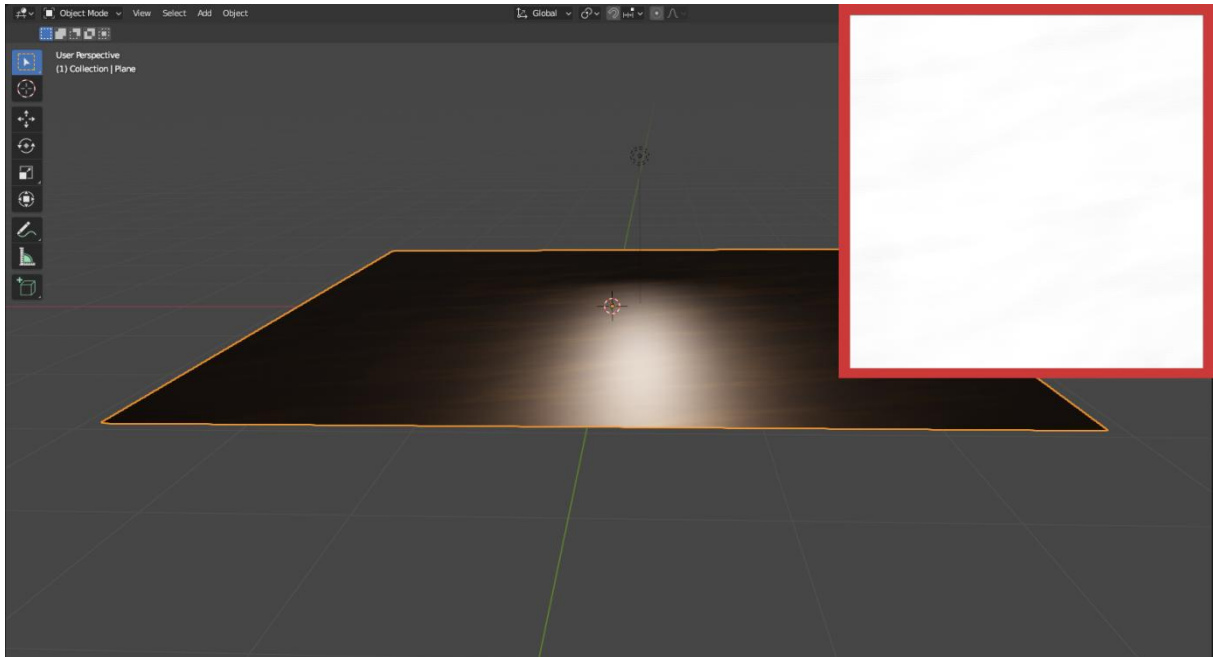
Diffuse map είναι μόνο το χρώμα του αντικειμένου. Σαν χρώμα, για παράδειγμα, μπορούν να θεωρηθούν τα χρώματα ενός ξύλου με τις λεπτομέρειες του, καθώς δεν είναι αναγκαίο να είναι μόνο ένα μονό χρώμα. Βάζοντας το χρώμα αυτό στο αντικείμενο, είναι παρόμοιο σαν να έχει πρακτικά ζωγραφίσει κάποιος το αντικείμενο, μόνο όμως με το diffuse map το φως που φωτίζει το αντικείμενο δεν μοιάζει σαν να έχει φωτίσει για παράδειγμα ένα ξύλο, αλλά περισσότερο σαν να έχει φωτίσει μία φωτογραφία ενός ξύλου.



Εικόνα 34. Ένα επίπεδο μόνο με την εικόνα ενός Diffuse map.

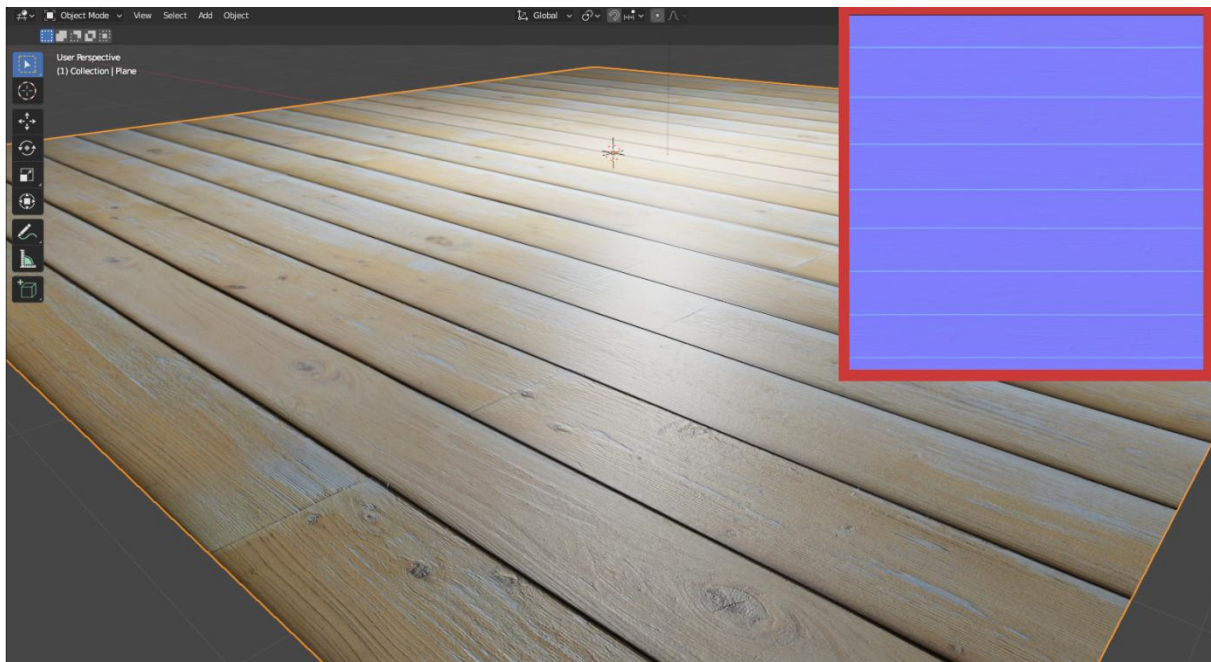
Metalness map είναι μία εικόνα που λειτουργεί ως χάρτης για το ποια σημεία του αντικειμένου θα έχουν την ψευδαίσθηση της μεταλλικότητας. Αυτός ο χάρτης μοιάζει με μία ασπρόμαυρη εικόνα, τα άσπρα σημεία θα αλλάζουν το υλικό σε μεταλλικό ενώ τα μαύρα σημεία θα αφαιρούν τη μεταλλικότητα του. Σε ένα υλικό όπως το ξύλο η εικόνα θα μοιάζει τελείως μαύρη, καθώς δεν υπάρχει καθόλου μεταλλικότητα σε μία ξύλινη επιφάνεια, για αυτό

συνήθως το πεδίο αυτό αφήνεται κενό. Στην προκειμένη περίπτωση, το παράδειγμα θα είναι μία επιφάνεια χρυσού αλουμινόχαρτου.



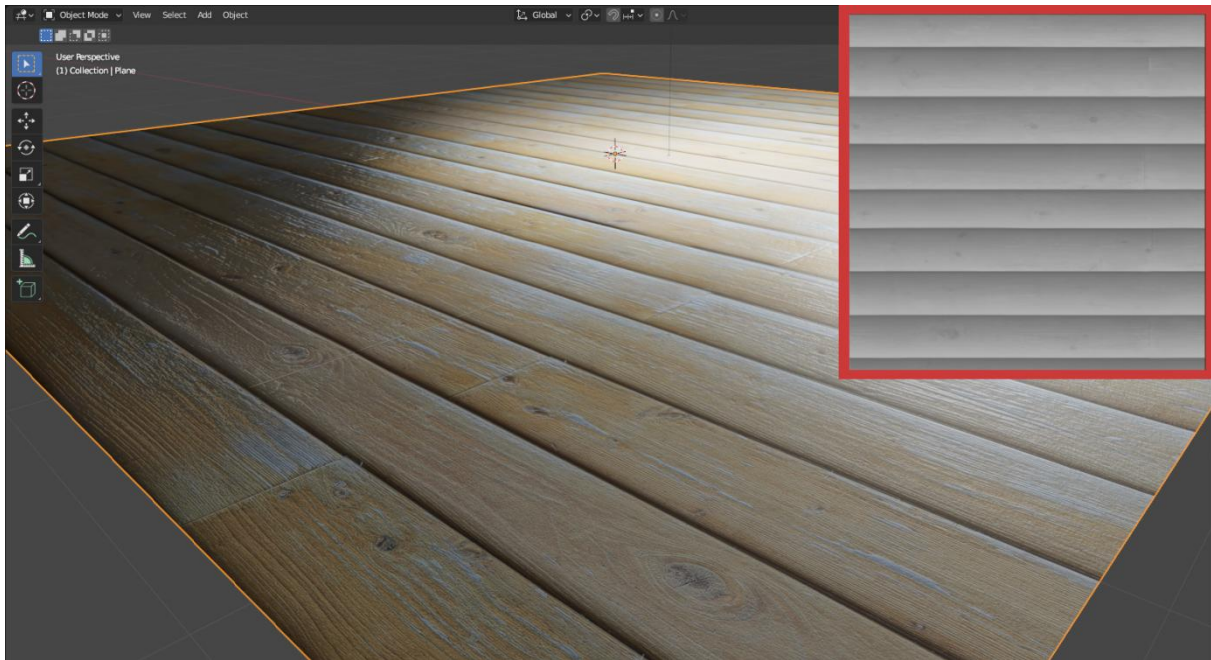
Εικόνα 35. Ένα επίπεδο με diffuse και metalness maps. Το άσπρο σημαίνει πως κατά κύριο λόγο ολόκληρο το αντικείμενο είναι μεταλλικό.

Normal map μοιάζει με μία εικόνα με μπλε λεπτομέρειες, οι οποίες στα λογισμικά μεταφράζεται ως μία ψεύτικη γεωμετρία. Η normal χαρτογράφηση δίνει πληροφορίες με σκοπό την παραποίηση του φωτισμού στον χώρο, δίνοντας το αποτέλεσμα βαθουλωμάτων και σημαδιών πάνω στο τρισδιάστατο αντικείμενο. Το είδος αυτό είναι πολύ σημαντικό, η ψεύτικη γεωμετρία απελευθερώνει πόρους από το σύστημα που θα χρειαζόταν να χρησιμοποιηθούν από το σύστημα, καθώς διαβάζοντας μόνο μία εικόνα αφαιρούμε τα χιλιάδες πολύγωνα που θα χρειαζόταν για να προσθέσουμε τις συγκεκριμένες λεπτομέρειες.



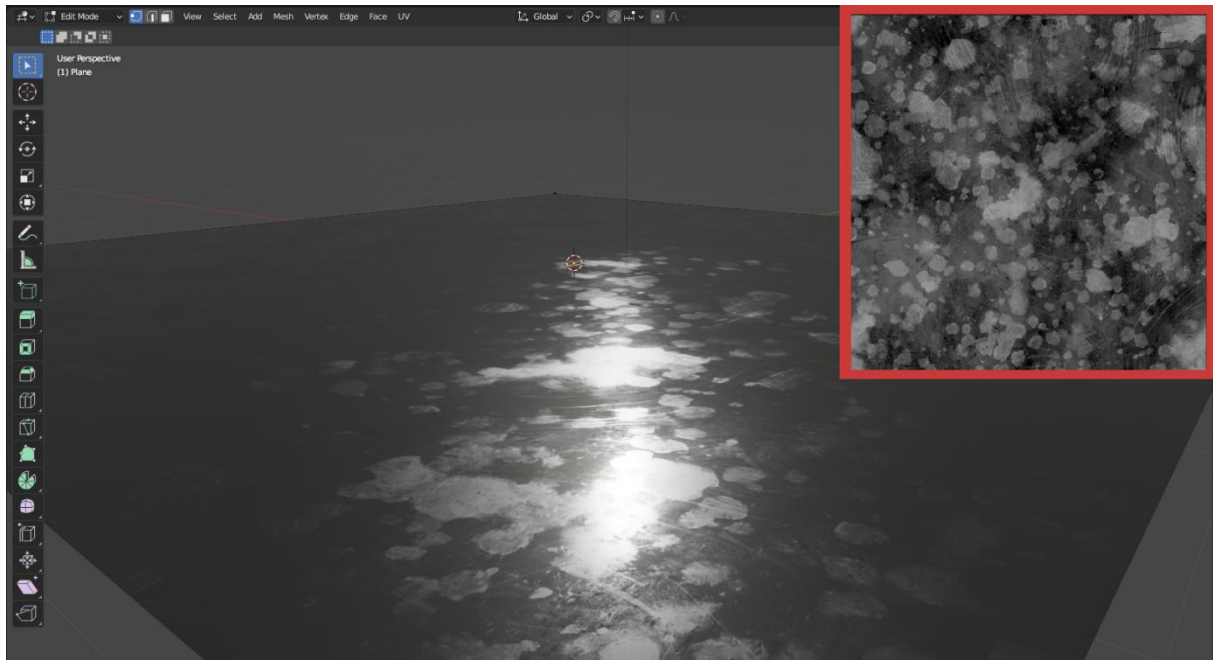
Εικόνα 36. Προσθήκη του normal map στο επίπεδο του ξύλου.

Height map ονομάζεται η πληροφορία που μπορεί να επηρεάσει τα πολύγωνα του αντικειμένου. Η συγκεκριμένη χαρτογράφηση είναι μία εικόνα στην κλίμακα του ασπρόμαυρου η οποία λέει στο λογισμικό το ποια σημεία θα μετακινήσουν τη γεωμετρία στον κατακόρυφο άξονα. Η χαρτογράφηση αυτή μερικές φορές μπορεί να λειτουργήσει και ως επιπρόσθετη λεπτομέρεια μαζί με το normal map, πράγμα που απορροφά περισσότερους πόρους για ένα ωραιότερο αποτέλεσμα, η συνηθισμένη της μορφή όμως είναι η επεξεργασία των πολυγώνων. Μία τέτοια χαρτογράφηση χρησιμοποιείται συνήθως από τα βιντεοπαιχνίδια για τη γρήγορη δημιουργία του επιπέδου της γης και των βουνών σε έναν μεγάλο χάρτη ανοιχτού κόσμου.



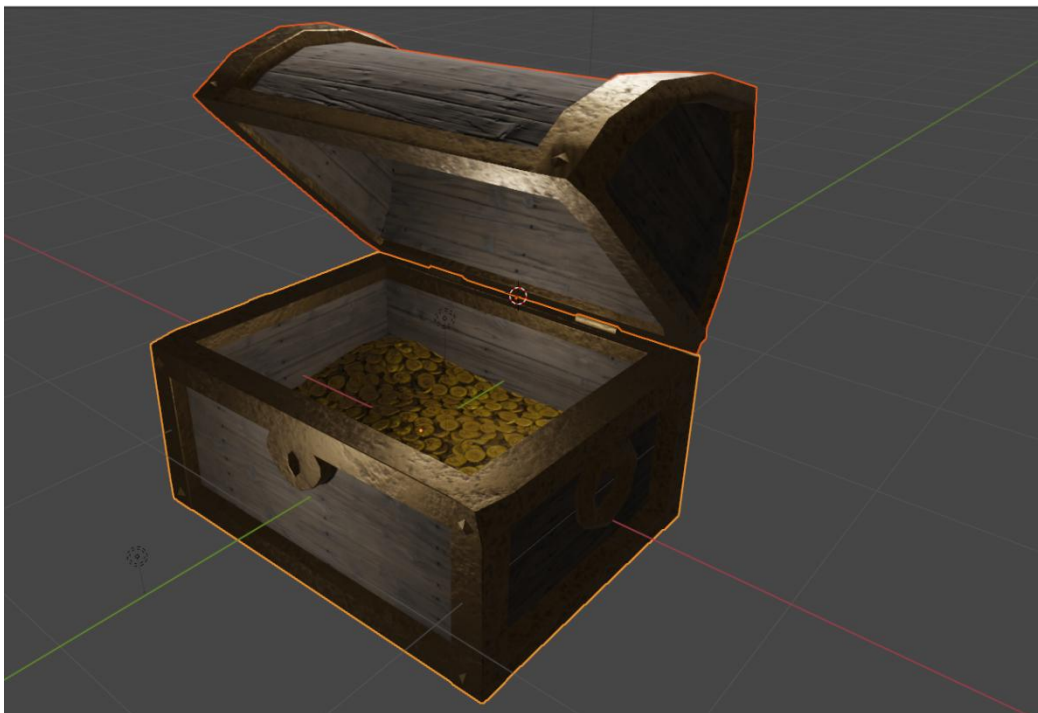
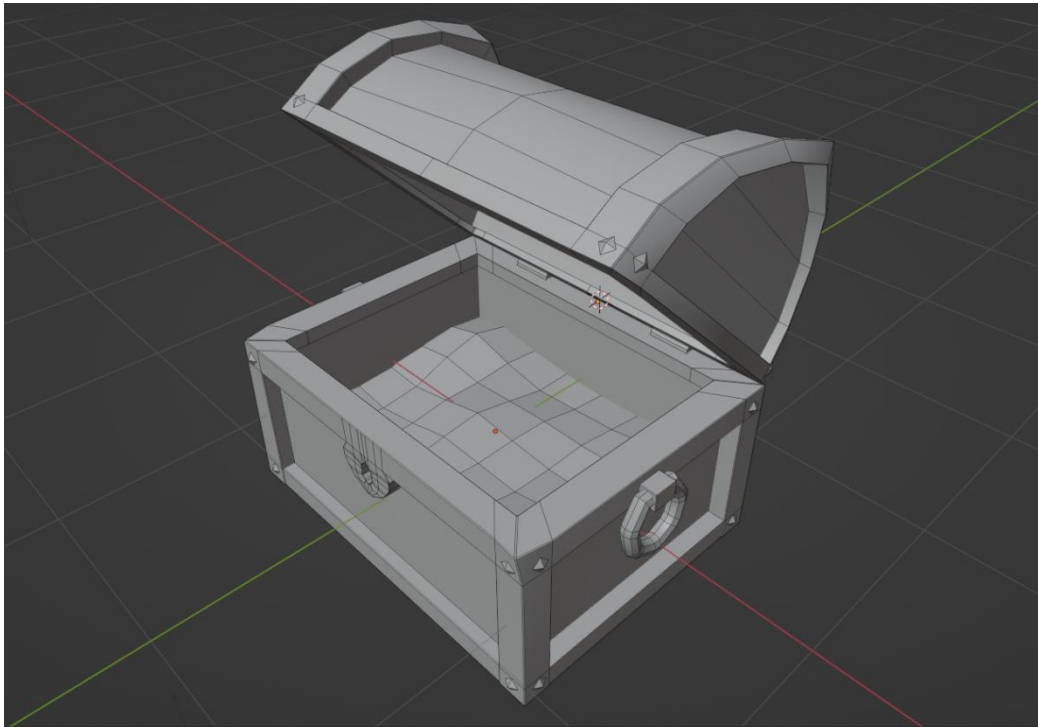
Εικόνα 37. Επιπρόσθετη η χρήση του height map χωρίς την επεξεργασία πολυγώνων από αυτό.

Roughness ή smoothness map είναι η πληροφορία που θα δώσει η εικόνα στο αντικείμενο για το ποια σημεία του θα είναι πιο τραχιά από τα υπόλοιπα. Αυτές οι χαρτογραφήσεις είναι απαραίτητες για τον ρεαλισμό ενός αντικειμένου, καθώς με αυτόν τον τρόπο μπορούν να προστεθούν μουντζούρες και κηλίδες πάνω σε αντικείμενα, όπως και γρατζουνιές ή θολούρες. Τέτοιες χαρτογραφήσεις συνήθως χρησιμοποιούνται συνεχώς πάνω σε καθρέφτες, παράθυρα και μεταλλικά αντικείμενα. Η roughness χαρτογράφηση αντίστοιχα λέγεται και smoothness, η ορολογία αυτή όμως εννοεί ακριβώς το αντίθετο, για αυτόν τον λόγο εάν κάποιο λογισμικό ζητάει για smoothness map, συνήθως υπάρχει και η επιλογή της μετατροπής του roughness map σε smoothness αυτόματα. Στην παραπάνω εικόνα δεν θα παρατηρηθούν μεγάλες αλλαγές με την χρήση ενός roughness map, για αυτό η επόμενη εικόνα αφορά μία ψηφιακή επιφάνεια που μοιάζει με καθρέφτη ή κάποιο μέταλλο.



Εικόνα 38. Χρήση όλων των χαρτογραφήσεων σε ένα υλικό που μοιάζει με καθρέφτη ή κάποιο μέταλλο.

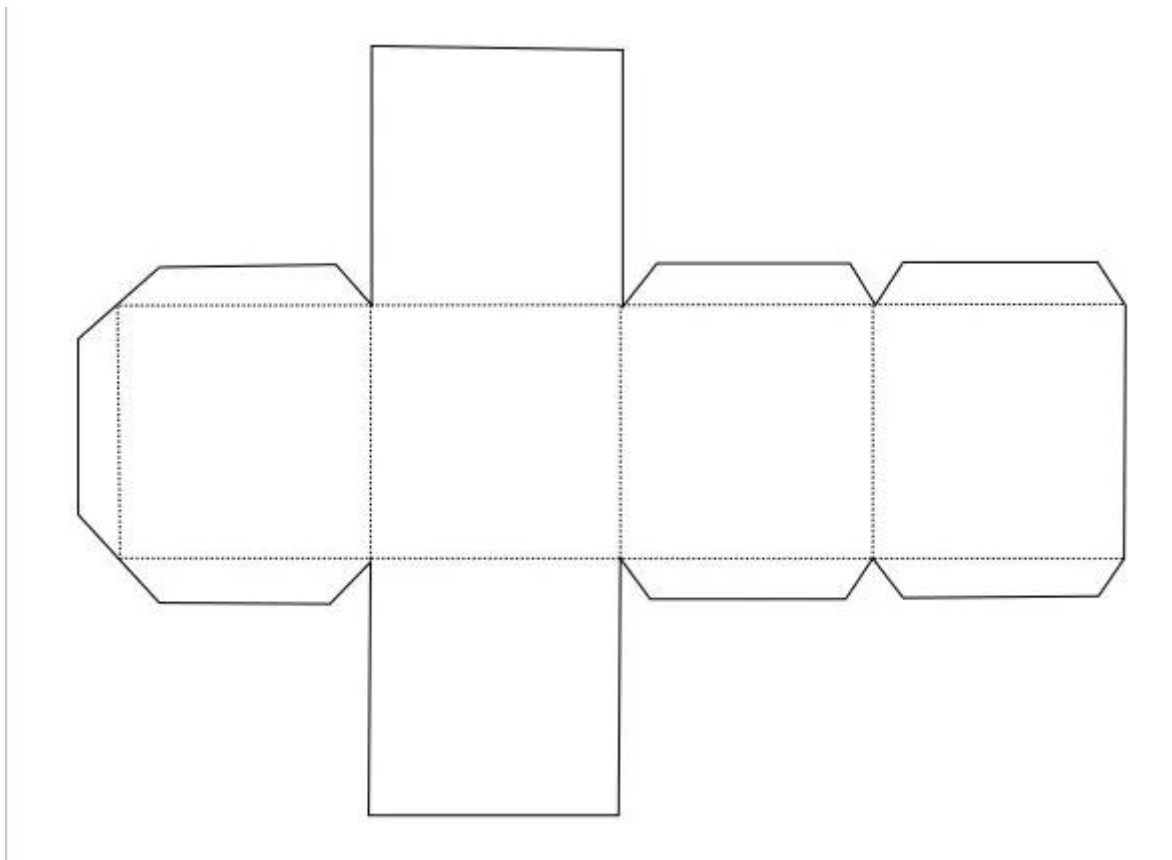
Οι παραπάνω υφές είναι οι βασικότερες υφές που χρησιμοποιούνται από τους καλλιτέχνες για τη δημιουργία των γραφικών σε ένα βιντεοπαιχνίδι. Στο “Davy Jones’ Locker” χρησιμοποιήθηκαν όλα αυτά τα είδη χαρτών, καθώς για την επίτευξη της βέλτιστης επίδοσης του βιντεοπαιχνιδιού ώστε να μπορέσει να εκτελεστεί στα περισσότερα συστήματα, πολλά αντικείμενα δεν χρησιμοποιούν όλα τα είδη ταυτόχρονα.



Εικόνα 39. Το σεντούκι του “Davy Jones’ Locker” χρησιμοποιώντας τις χαρτογραφίες "diffuse", "normal" και "roughness".

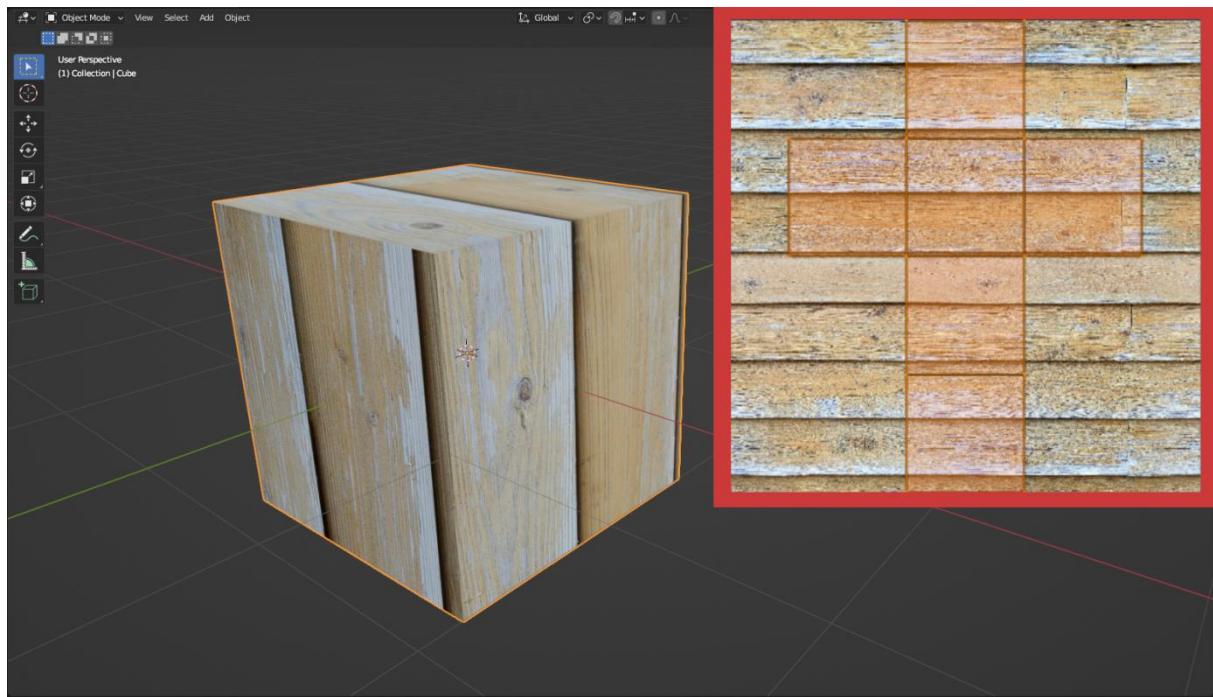
Όταν ο καλλιτέχνης έχει βρει τα υλικά τα οποία θέλει να χρησιμοποιήσει για το ντύσιμο του τρισδιάστατου αντικειμένου, πρέπει να τα εφαρμόσει με τον σωστό τρόπο πάνω στο αντικείμενο. Αυτή η δουλειά είναι μία επιπλέον τεχνική με ονομασία “UV unwrap”. Όσο πιο περίπλοκο είναι το αντικείμενο αυτή η τεχνική γίνεται δυσκολότερη. Εξηγώντας το UV

unwrapping με λίγα λόγια, είναι το κόψιμο του αντικειμένου σε συγκεκριμένο τρόπο και η τοποθέτηση του πάνω στην εικόνα της υφής. Αυτό πάνω σε μία επίπεδη επιφάνεια είναι πολύ εύκολο, αλλά πάνω σε ένα μοντέλο ενός ανθρώπου ή ζώου είναι μία χρονοβόρα εργασία που χρειάζεται επαγγελματικές γνώσεις από τον καλλιτέχνη. Για να καταλάβει κανείς καλύτερα τι εννοείται λέγοντας “κόψιμο ενός αντικειμένου”, ίσως θα είναι πιο κατανοητό σε ένα παράδειγμα ενός πραγματικού αντικειμένου. Εάν θέλουμε να φτιάξουμε ένα τετράγωνο κουτί με όλες τις επιφάνειες ίσες, η παρακάτω εικόνα είναι το πως θα μοιάζει το πατρόν του κουτιού.



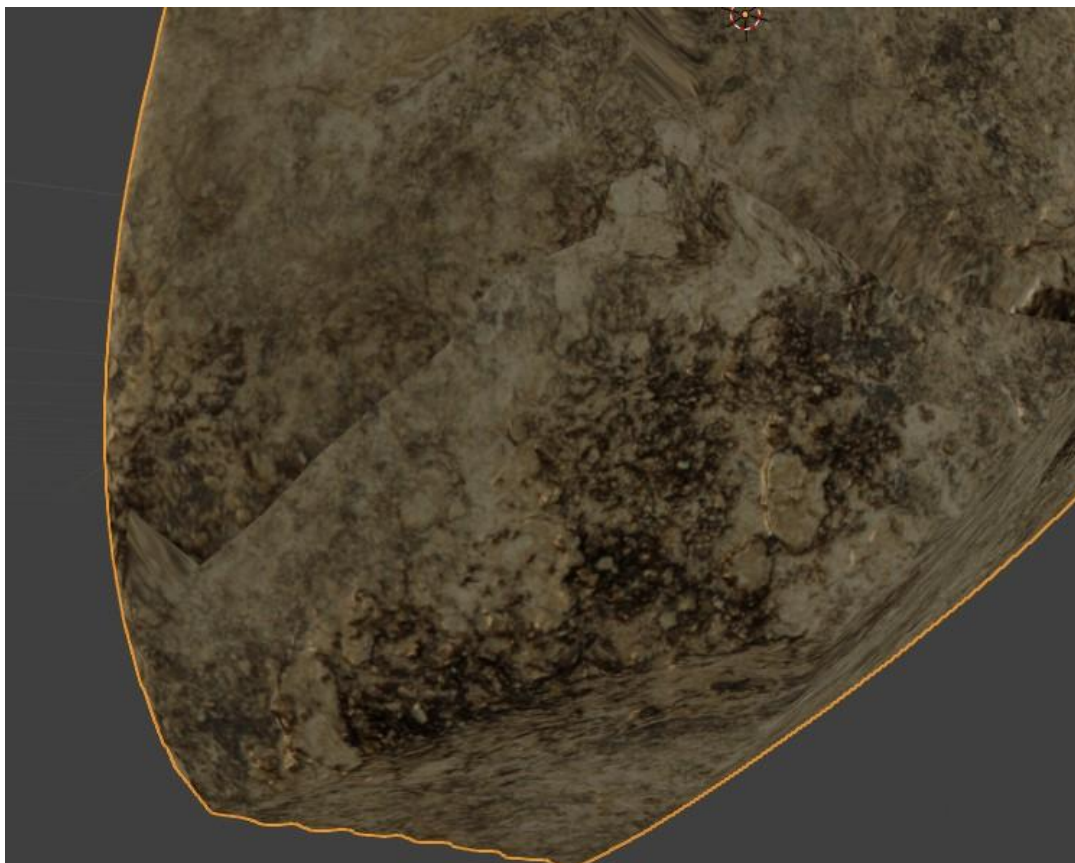
Εικόνα 40. Πατρόν δημιουργίας ενός τετράγωνου κουτιού.

Αντίστοιχα, το ξετύλιγμα ενός κουτιού στο λογισμικό τρισδιάστατων μοντέλων θα μοιάζει παρόμοιο. Γνωστή ονομασία του κοψίματος αυτού είναι το “seam”.



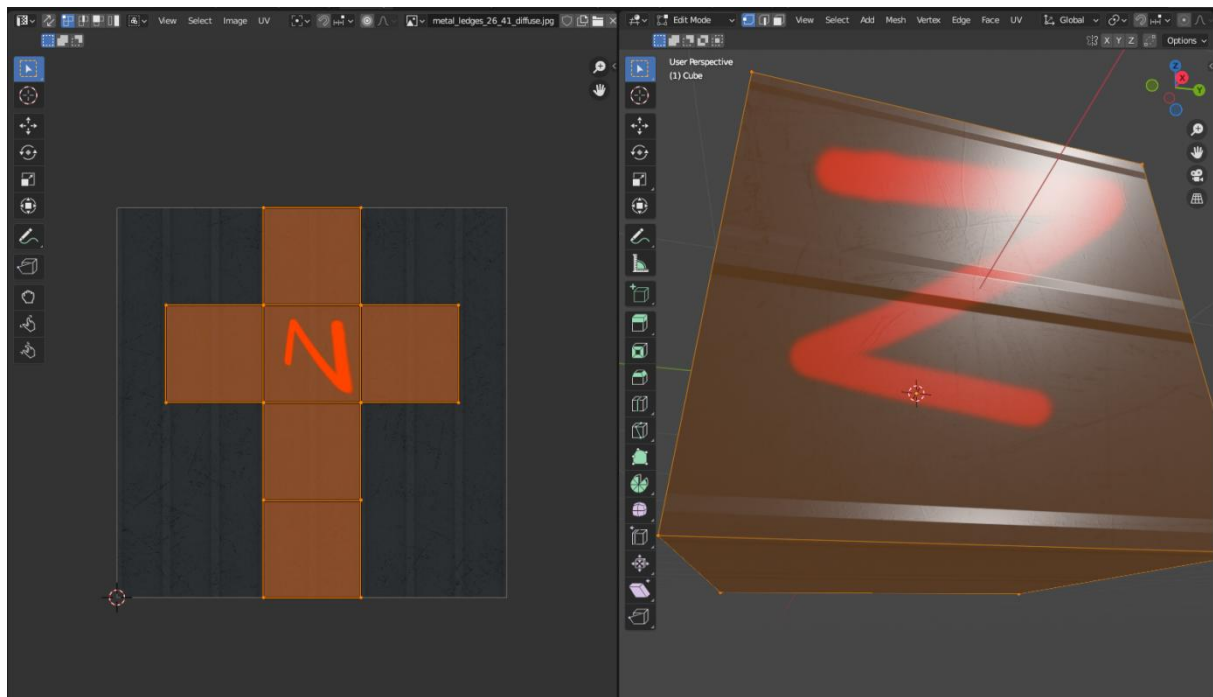
Εικόνα 41. UV unwrap ενός κουτιού μέσα στον τρισδιάστατο χώρο.

Παρατηρώντας την παραπάνω εικόνα είναι ορατό το πως έχει ξετυλιχθεί ο κύβος πάνω στο υλικό, είναι όμως ορατό και το πως στην κάθε μεριά του κύβου τα ξύλα έχουν διαφορετική κατεύθυνση. Αυτό είναι η συνέπεια του κοψίματος, για αυτόν τον λόγο σε πιο περίπλοκα μοντέλα μπορεί ο καλλιτέχνης να ορίσει τα πολύγωνα στα οποία θα κοπεί το αντικείμενο με σκοπό να χειραγωγήσει το πως θα φαίνονται τα υλικά πάνω του. Συνήθως το κόψιμο αυτό γίνεται σε μεριές του αντικειμένου που είναι δύσκολα ορατές από τον παίκτη, εάν όμως γίνει με λάθος τρόπο ή ο παίκτης ψάξει για το seam, μπορεί εύκολα να παρατηρηθεί η αλλαγή. Πολύ συχνό φαινόμενο αυτού είναι μεγάλοι βράχοι σε βιντεοπαιχνίδια που είναι συχνά ορατή αυτή η αλλαγή.



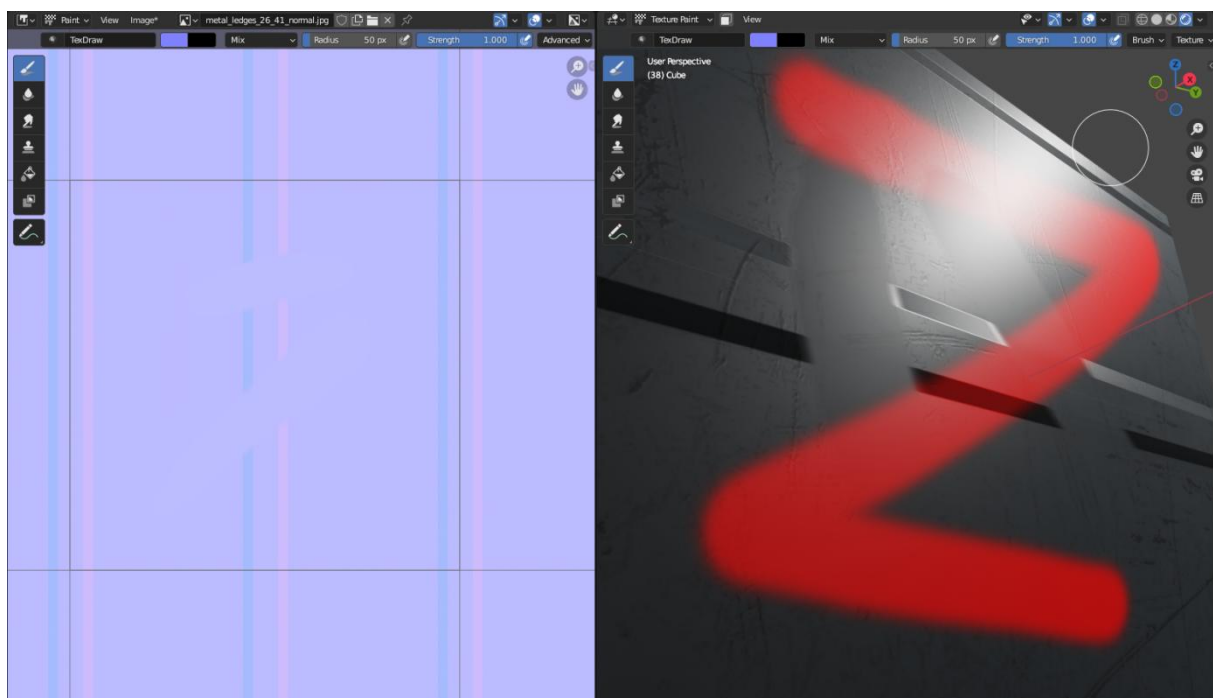
Εικόνα 42. Ορατό κόψιμο (seam) πάνω σε πέτρα.

Έχοντας τροποποιήσει το UV ενός μοντέλου και έχοντας τοποθετήσει τις υφές με τον τρόπο της αρεσκείας του καλλιτέχνη, μία καινούρια τεχνική έρχεται στο προσκήνιο για να δοθούν περισσότερες λεπτομέρειες και ρεαλισμός στο αντικείμενο. Οι πιο συνηθισμένες λεπτομέρειες που προστίθενται σε αυτό το στάδιο είναι διάφορες φθορές στα σημεία του αντικειμένου που θα είχαν φθαρεί στην πραγματικότητα, όπως για παράδειγμα οι γωνίες του. Επιπλέον πληροφορία θα μπορούσε να είναι μία ζωγραφιά, ένα αυτοκόλλητο, ή ακόμα και ψεύτικα βαθουλώματα και βίδες μέσω των παραπάνω υφών. Ο τρόπος που δίνεται η επιπλέον λεπτομέρεια είναι η τοποθέτηση τους στις παραπάνω χαρτογραφήσεις. Εάν χρειάζεται να προστεθεί απλά μία ζωγραφιά πάνω στο αντικείμενο, χωρίς την αλλαγή του υλικού στο συγκεκριμένο σημείο, επηρεάζουμε μόνο το diffuse map.



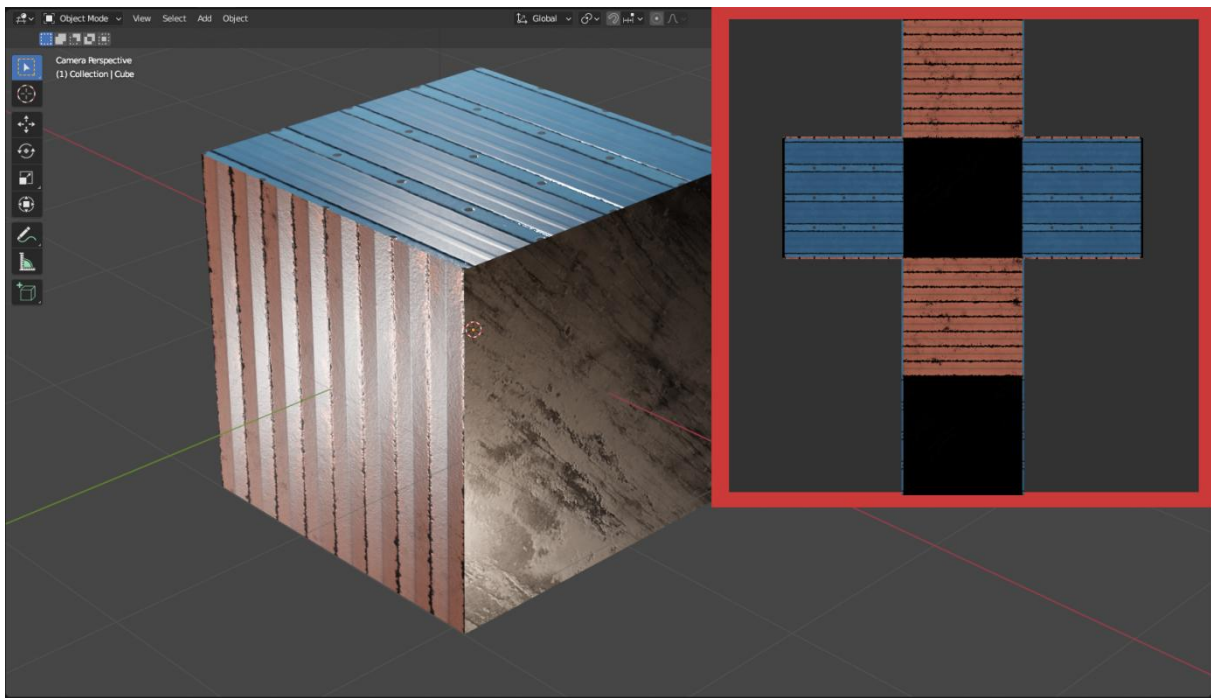
Εικόνα 43. Επεξεργασία του diffuse map.

Αντίστοιχα αυτό μπορεί να χρησιμοποιηθεί και για τις υπόλοιπες χαρτογραφήσεις δίνοντας περισσότερες δυνατότητες. Επηρεάζοντας το normal map για παράδειγμα, μπορούμε να αφαιρέσουμε ψευδή γεωμετρία ή να προσθέσουμε επιπλέον, όπως για παράδειγμα βίδες ή βαθουλώματα.



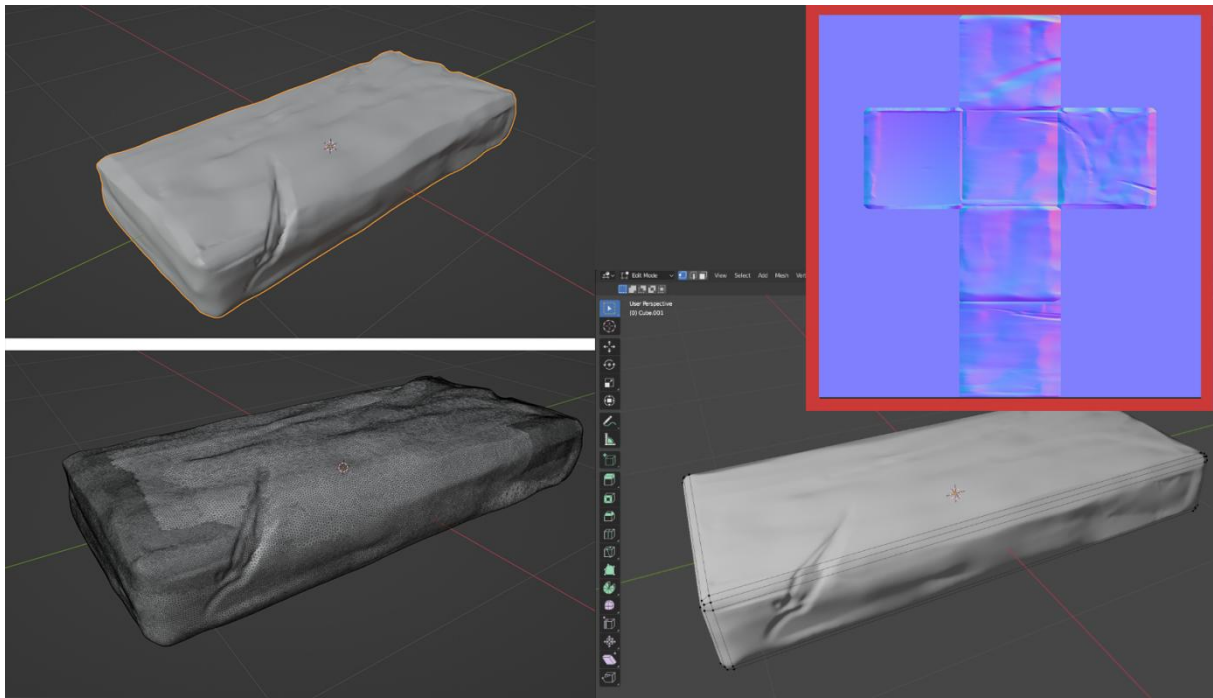
Εικόνα 44. Επεξεργασία του normal map.

Όταν πλέον έχει επιτευχθεί το επιθυμητό αποτέλεσμα, ο καλλιτέχνης προχωράει στο επόμενο στάδιο, το σημείο του “Texture Baking”. Texture baking είναι η συλλογή και μετάφραση όλων των υφών ενός αντικειμένου σε μία εικόνα με σκοπό την μείωση των πόρων που θα χρειάζεται το σύστημα για να εκτελέσει το βιντεοπαιχνίδι.



Εικόνα 45. Μετάφραση όλων των υλικών σε μία εικόνα. Diffuse.

Μία επιπλέον χρήση του baking είναι η μεταφορά πληροφορίας από ένα αντικείμενο σε ένα άλλο. Όπως αναφέρθηκε στο κεφάλαιο με τα πολύγωνα, τα high poly αντικείμενα που έχουν επεξεργασθεί με ψηφιακή γλυπτική συνδυάζοντας την τεχνική του baking μπορούν να μεταφραστούν σε normal map πάνω σε ένα αντικείμενο low poly. Με αυτόν τον τρόπο αποθηκεύεται η πληροφορία του high poly αντικειμένου με τις επιδόσεις του low poly αντικειμένου.



Εικόνα 46. Μετάφραση της γεωμετρικής πληροφορίας από high poly σε εικόνα normal map.

Τέλος, μία τεχνική που μπορεί να εμπλουτίσει τα γραφικά ενός βιντεοπαιχνιδιού είναι η χρήση των λεγόμενων “Decals”. Τα decals είναι δισδιάστατες εικόνες, κατά κύριο λόγο σε “.png” μορφή με σκοπό την επιπρόσθετη πληροφορία στη σκηνή. Τα decals συνήθως χρησιμοποιούνται για αντικείμενα που δεν χρειάζεται να βρίσκονται σε τρισδιάστατη μορφή μέσα στη σκηνή. Τα αντικείμενα αυτά μπορεί να είναι μία αφίσα σε κάποιον τοίχο, αίματα σε τοίχους και πατώματα ή ακόμα και κάποια οπτικά εφέ όπως φωτιές κεριών, εκρήξεις και άλλα που ο παίκτης δεν τα βλέπει από όλους τους άξονες με την κάμερα του.



Εικόνα 47. Decals αίματος στο "Davy Jones' Locker".

2.4 Σχεδίαση της διεπαφής και εμπειρίας χρήστη.

Ένα πολύ σημαντικό μέρος της ανάπτυξης ενός βιντεοπαιχνιδιού είναι τα γραφικά των μενού που βλέπει ο παίκτης πριν και κατά τη διάρκεια του παιχνιδιού. Ο σχεδιασμός αυτός ονομάζεται διεπαφή χρήστη και περιλαμβάνεται από δισδιάστατες εικόνες και τις διαδράσεις τους. Για παράδειγμα, πατώντας το κουμπί “settings”, εμφανίζεται ένα καινούριο μενού στο οποίο μπορεί ο παίκτης να αλλάξει τις ρυθμίσεις του βιντεοπαιχνιδιού.



Εικόνα 48. Αρχικό Μενού του "Davy Jones' Locker".

Για το “Davy Jones’ Locker” ζωγράφισα προσωπικά τα στοιχεία, θέλοντας να δώσω μία πιο καλλιτεχνική και ζωγραφική αίσθηση σε αντίθεση με τον ρεαλισμό του πραγματικού παιχνιδιού. Μέσα στο βιντεοπαιχνίδι, όταν ο παίκτης μπορεί να διαδράσει με τα αντικείμενα, εμφανίζονται κάποιες πληροφορίες στην εικόνα του με σκοπό την ενημέρωση των δυνατών κινήσεων που μπορεί να κάνει την στιγμή εκείνη.



Εικόνα 49. Στοιχεία διεπαφής εμπειρίας χρήστη. Κείμενα και γάντζος.

3. Μηχανές παιχνιδιών και προγραμματισμός.

Μηχανή παιχνιδιών ονομάζουμε το λογισμικό στο οποίο προσθέτουμε τα μοντέλα και τα στοιχεία που έχουμε φτιάξει πρωτίστως, ώστε να προχωρήσουμε στην δημιουργία διαδράσεων τους με τον παίκτη. Στην ουσία λειτουργεί ως το εργαλείο στο οποίο κατασκευάζεται το βιντεοπαιχνίδι και οι μηχανισμοί του. Ταυτόχρονα, οι μηχανές παιχνιδιών

πλέον δεν είναι περιορισμένες μόνο στη βιομηχανία των βιντεοπαιχνιδιών. Τα αποτελέσματα που μπορούν να προκύψουν είναι προχωρημένα σε ένα αξιοθαύμαστο επίπεδο, πράγμα που τις κάνει εργαλείο πλέον για τη δημιουργία ταινιών ή τρισδιάστατων κινουμένων σχεδίων, όπως συνέβη με τις ταινίες “Westworld” της HBO και “Mandalorian” της Disney με τη βοήθεια της Unreal Engine (μηχανής παιχνιδιών)⁹. Εν κατακλείδι, η μηχανή παιχνιδιών έχει ως ρόλο να προσφέρει μία διεπαφή χρήστη λογισμικού, καθώς είναι υπεύθυνη και για τη φυσική του παιχνιδιού, προσφέροντας συνάμα και την ελευθερία του να προγραμματίσεις με κάποια γλώσσα προγραμματισμού όλες τις διαδράσεις που επρόκειτο να προσθέσουν στην εμπειρία του παίκτη.



Εικόνα 50. Σκηνή γυρισμένη σε περιβάλλον μέσα στην μηχανή παιχνιδιών Unreal Engine¹⁰.
"Mandalorian" 2019.

⁹ Statt, N., & Roettgers, J. (2022, April 6). With Unreal Engine 5, Epic wants to transform how games and movies get made. *Protocol*. <https://www.protocol.com/newsletters/entertainment/epic-unreal-engine-5>

¹⁰ Yeo, A. (2022, March 1). *The secret to the incredible landscapes of “The Mandalorian”: Epic Games’ Unreal Engine*.

Mashable. <https://mashable.com/video/the-mandalorian-epic-games-unreal-engine-set-landscape>

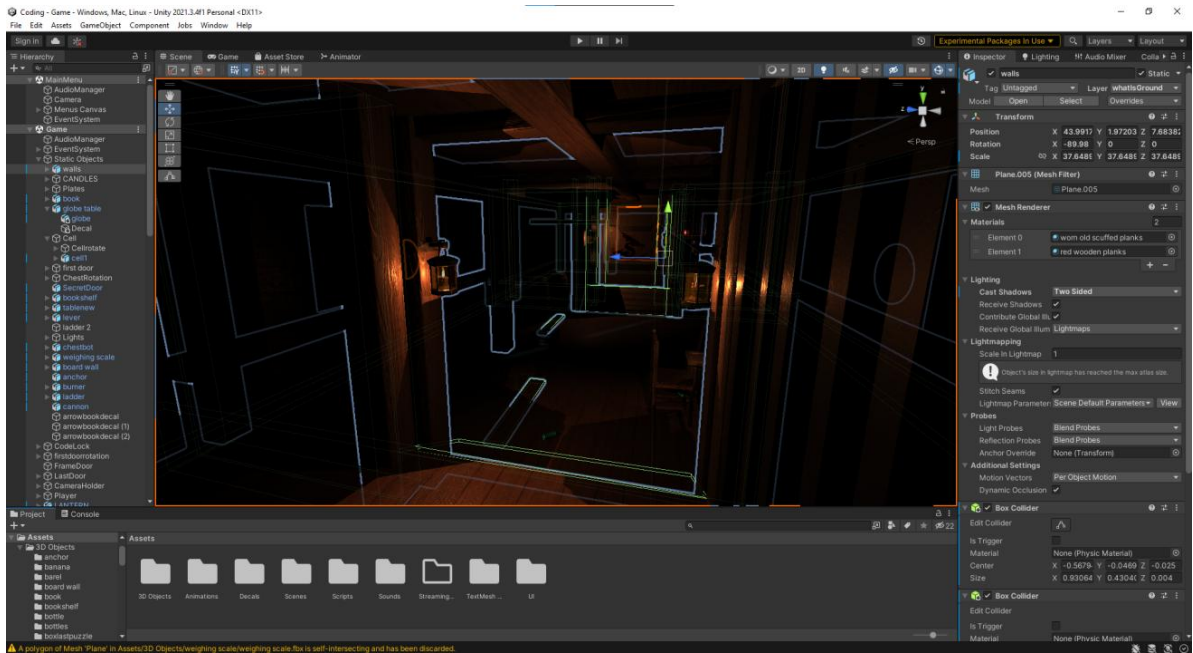
3.1 Unity, Unreal Engine και γλώσσες προγραμματισμού.

Οι δύο πιο σημαντικές και γνωστές μηχανές βιντεοπαιχνιδιών είναι η Unity και η Unreal Engine. Και οι δύο είναι δωρεάν προς όλους με μοναδικό κόστος ένα ποσοστό εσόδων από τα κέρδη του βιντεοπαιχνιδιού μετά τη δημοσίευση του. Οι διαφορές των δύο μηχανών κρύβονται σε λεπτομέρειες στα γραφικά του βιντεοπαιχνιδιού και στον φωτισμό, στη φυσική πίσω από τη μηχανή, στην χρήση γλώσσας προγραμματισμού και σε πολλά εργαλεία που αφορούν διάφορους τομείς, όπως και στην βέλτιστη επίδοση του τίτλου χωρίς να απορροφά πολλούς πόρους από το σύστημα στο οποίο θα εκτελείται. Στην πραγματικότητα και στις δύο αυτές μηχανές μπορεί κανείς να δημιουργήσει παρόμοια αποτελέσματα στο ζήτημα των γραφικών και της αισθητικής. Η γλώσσα προγραμματισμού που χρησιμοποιείται στην Unreal Engine είναι η C++, ενώ στην Unity είναι η C#. Η C# είναι μία εξέλιξη της C++ και είναι πολύ πιο εύκολη να την μάθει ένας αρχάριος προγραμματιστής. Η C++ όμως είναι πιο ελαφριά στη χρήση της, πράγμα που βοηθάει στην βελτιστοποίηση της επίδοσης του βιντεοπαιχνιδιού¹¹. Σημαντικό επίσης είναι πως ο κώδικας της C++ είναι ανεξάρτητος πλατφόρμας, ενώ ο κώδικας της C# είναι συγκεκριμένος για τα Windows¹², χωρίς όμως αυτό να περιορίζει τη χρήση του σε διάφορα λογισμικά όπως MAC και Linux, ενώ μπορεί να γίνει export και για διάφορες πλατφόρμες.

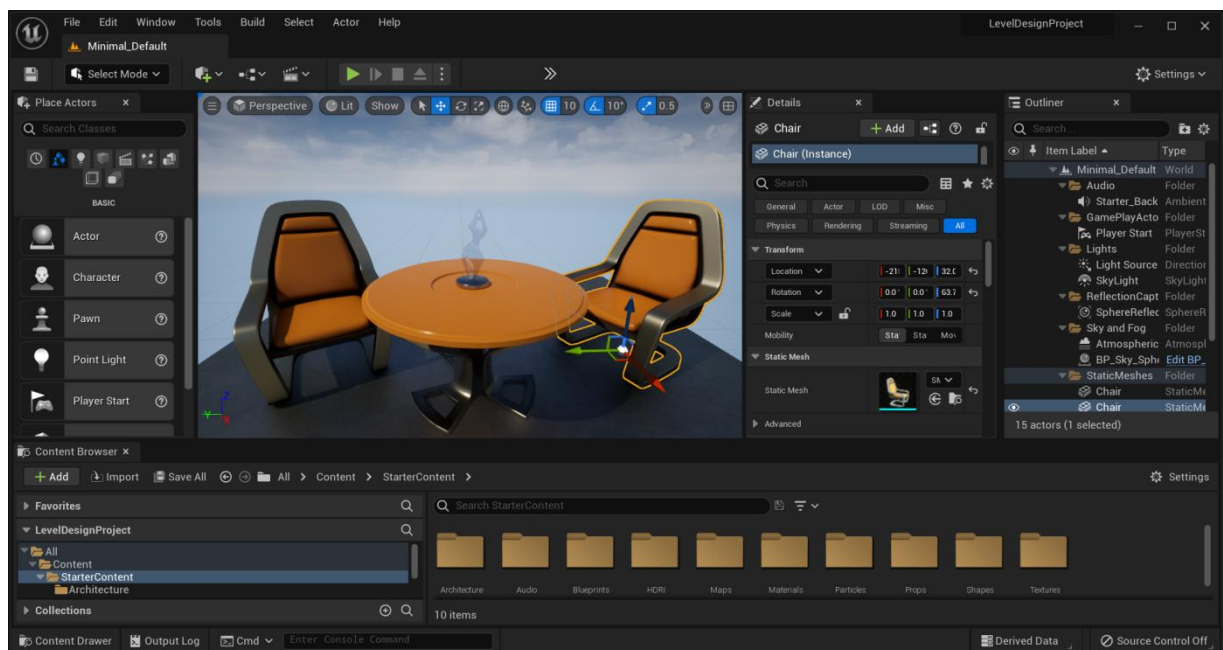
¹¹ Shiotsu, Y. (2021). C# vs. C++: Which Language is Right for Your Software Project? *Upwork*.

<https://www.upwork.com/resources/c-sharp-vs-c-plus-plus#:~:text=C%2B%2B%20is%20much%20more%20lightweight,applications%20where%20performance%20is%20important.>

¹² Great Learning. (2022, July 11). *C# vs C++: Differences and Similarities Between C# & C++* - Great Learning. Great Learning Blog: Free Resources What Matters to Shape Your Career! <https://www.mygreatlearning.com/blog/c-sharp-vs-cpp/>



Εικόνα 51. Διεπαφή χρήστη του "Unity Engine".



Εικόνα 52. Διεπαφή χρήστη του "Unreal Engine".

Πέραν των δύο παραπάνω λογισμικών, πολλά μεγάλα στούντιο δημιουργούν το δικό τους εξ' ολοκλήρου για τη δημιουργία των βιντεοπαιχνιδιών τους. Η πράξη αυτή ενώ είναι μία πολύ δύσκολη και χρονοβόρα διαδικασία, εν τέλει προσφέρει περισσότερη ελευθερία τροποποίησης του λογισμικού στα δικά τους πρότυπα και περισσότερα κέρδη κατά τη δημοσίευση των βιντεοπαιχνιδιών τους, καθώς δεν δεσμεύονται πλέον στην επιπρόσθετη πληρωμή ενός τρίτου λογισμικού. Παραδείγματος χάρη, τρεις κολοσσοί της βιομηχανίας

όπως η Bethesda και η Ubisoft έχουν τα δικά τους λογισμικά. Της Bethesda ονομάζεται “Creation Engine”¹³ και της Ubisoft “AnvilNext”¹⁴. Αυτό όμως δεν σημαίνει πως όλα τα στούντιο βιντεοπαιχνιδιών πρέπει να φτιάξουν το δικό τους λογισμικό. Εάν ένα μικρότερο στούντιο δεν έχει το εργατικό δυναμικό και τα χρήματα για την επένδυση μίας νέας μηχανής βιντεοπαιχνιδιών, σπεύδει στα ήδη υπάρχοντα και πολύ αποτελεσματικά λογισμικά, καθώς είναι προτιμότερο να προχωρήσει ευθύς στη δημιουργία. Για το “Davy Jones’ Locker” χρησιμοποίησα τη μηχανή βιντεοπαιχνιδιών Unity, καθώς ήταν ένα λογισμικό που χρησιμοποιούσα συχνά κατά τη διάρκεια της φοιτητικής μου ζωής και ήμουν περισσότερο οικείος με αυτό και τη γλώσσα C#.

3.2 Προγραμματισμός του “Davy Jones’ Locker”.

Ο τρόπος που δούλεψα στο κομμάτι του προγραμματισμού είναι ο εξής. Αρχικά σημείωσα τι μηχανισμούς ήθελα να έχει το παιχνίδι και την αλληλεπίδραση μεταξύ τους. Στη συνέχεια έγραψα τα αρχικά στάδια του κώδικα και με τη βοήθεια του διαδικτύου επίλυσα προβλήματα που έβρισκα κατά τη διάρκεια. Όταν πλέον ήμουν περισσότερο οικείος με τις εντολές, οι λύσεις που έβρισκα ήταν δικές μου πλήρως και προχώρησα στην επίλυση προβλημάτων που πρωτίστως δεν είχα βρει λύσεις. Όταν πλέον εφάρμοσα όλους τους κώδικες στα αντίστοιχα αντικείμενα μέσα στο Unity, έκανα πολλαπλές δοκιμές για το πως αλληλεπιδρούν μεταξύ τους και προχώρησα στην επίλυση επιπλέον προβλημάτων και διαρροών που δημιουργήθηκαν στην πορεία. Επίσης, επειδή δεν γνώριζα εξ’ αρχής πολλούς μηχανισμούς που θα χρειαζόταν ένα ολοκληρωμένο βιντεοπαιχνίδι, πολλοί μηχανισμοί γεννήθηκαν ή άλλαξαν κατά τη διάρκεια.

Η αρχή του βιντεοπαιχνιδιού ξεκίνησε με τη δημιουργία της κίνησης του χαρακτήρα και της κάμερας. Η αρχική ιδέα ήταν η αργή κίνηση σε ταχύτητα περπατήματος, χωρίς να υπάρχει η δυνατότητα του άλματος και του καθίσματος. Όταν όμως δημιούργησα το πρώτο μέρος, κατά τη διάρκεια των δοκιμών παρατήρησα πως υπήρχε η συνεχής επιθυμία του να μπορεί ο χαρακτήρας να σκύψει και να πηδήξει για να φανεί κάτι καλύτερα στην κάμερα. Για τον παραπάνω λόγο, εν τέλει συμπλήρωσα και τον κώδικα του σκυψίματος και του άλματος. Για

¹³ “The Graphics Technology of Fallout 4”. Bethesda Softworks. November 4, 2015.

<https://bethesda.net/en/article/2Y37xeRPeUW0EgkgKW8oA/the-graphics-technology-of-fallout-4>

¹⁴ Osborn, A. (2017, September 16). Why Ubisoft Ditched Unreal in Favor of Its Own Proprietary Tech. *IGN*.

<https://www.ign.com/articles/2017/09/15/why-ubisoft-ditched-unreal-in-favor-of-its-own-proprietary-tech>

την υλοποίηση της κίνησης ξεκίνησα με το να δηλώνεται ο χαρακτήρας ως “isGrounded” όταν ακουμπάει σε κάποιο επίπεδο που είναι δηλωμένο σε layer “whatIsGround”, ώστε όταν ο χαρακτήρας θα βρίσκεται σε αυτήν την κατάσταση να μπορεί να δεχθεί κάποιες εντολές που θα του δώσουμε. Για παράδειγμα, εάν ο χαρακτήρας δεν βρισκόταν ποτέ σε αυτή την κατάσταση, θα μπορούσε ο παίκτης να πατάει το κουμπί του άλματος όταν ο χαρακτήρας θα βρίσκεται ακόμα στον αέρα, με αποτέλεσμα να μπορεί να πηδάει συνεχώς σαν να πετάει στην χώρα, έτσι λέγοντας στο σύστημα πως μπορεί ο χαρακτήρας να πηδάει μόνο όταν είναι “isGrounded”, περιορίζονται τέτοια λάθη. Στη συνέχεια δήλωσα τα κουμπιά που θα χρειαζόταν να πατήσει ο παίκτης για την κίνηση και με έναν κώδικα “if()” την κατάσταση που θα βρισκόταν ο παίκτης κατά τη διάρκεια της κίνησης, όπως για παράδειγμα, πατώντας το κουμπί “Ctrl” ο χαρακτήρας θα βρίσκεται σε θέση “crouching”, οπότε θα σκύβει και θα μειώνεται η ταχύτητα του από την προκαθορισμένη στην κατάσταση “walking”. Όσο αναφορά την κίνηση της κάμερας, χρειάστηκε να δηλωθούν οι μεταβλητές X και Y στην κίνηση του ποντικιού μέσω της εντολής “Input.GetAxisRaw()” και στη συνέχεια ο πολλαπλασιασμός αυτού με “Time.fixedDeltaTime” και με ένα νούμερο για την οριοθέτηση της ευαισθησίας, ώστε ο παίκτης να μπορεί να περιορίσει το πόσο γρήγορα θα κουνιέται η κάμερα όταν κουνιέται το ποντίκι. Στην συνέχεια, παρατηρήθηκε ένα πρόβλημα, ο παίκτης μπορούσε να γυρίσει την κάμερα χωρίς περιορισμό, με αποτέλεσμα όταν ο παίκτης κοιτούσε στον ουρανό να μπορούσε να γυρίσει την κάμερα και να συνεχίσει να κοιτάει προς τα πίσω του, αυτό λύθηκε χρησιμοποιώντας την εντολή “Mathf.Clamp()” με σκοπό να περιοριστεί το πόσο ψηλά ή χαμηλά θα μπορεί να κοιτάει ο χαρακτήρας.

```
public void Update()
{ float mouseX = Input.GetAxisRaw("Mouse X") * Time.fixedDeltaTime * sensX;
  float mouseY = Input.GetAxisRaw("Mouse Y") * Time.fixedDeltaTime * sensY;

  yRotation += mouseX;
  xRotation -= mouseY;
  xRotation = Mathf.Clamp(xRotation, -90f, 90f);

  //rotate cam and orientation
  transform.rotation = Quaternion.Euler(xRotation, yRotation, 0);
  orientation.rotation = Quaternion.Euler(0, yRotation, 0); }
```

Πίνακας 1. Κίνηση της κάμερας.

Όταν ολοκλήρωσα την κίνηση του χαρακτήρα αποφάσισα να φτιάξω το αρχικό μενού του παιχνιδιού. Στο αρχικό μενού ήθελα να συμπεριλαμβάνεται το κουμπί για την έναρξη (Start) του παιχνιδιού, το κουμπί των ρυθμίσεων (Settings) και κάτω κάτω το κουμπί της εξόδου (Quit). Πατώντας το κουμπί της έναρξης χρησιμοποιείται μία εντολή “SceneManager.LoadScene()” με σκοπό να φορτώσει την επιλεγμένη σκηνή του παιχνιδιού και να ξεκινήσει ο χρήστης να παίζει. Αντίστοιχα, με το κουμπί της εξόδου ο χρήστης κλείνει το παιχνίδι με την εντολή “Application.Quit();”. Όσο αναφορά όμως το κουμπί των ρυθμίσεων χρησιμοποιείται πάλι η εντολή που χρησιμοποιήθηκε για την έναρξη, αντίθετα όμως αντί να ξεκινήσει το παιχνίδι, ανοίγει απλά ένα νέο παράθυρο με τις ρυθμίσεις. Στις ρυθμίσεις υπάρχουν ολισθητές για την ρύθμιση της ευαισθησίας του ποντικιού και την αυξομείωση του ήχου, κάτι που θα αναλυθεί αργότερα. Για να μπορούν να λειτουργούν οι αλλαγές στους ολισθητές με αποτέλεσμα να επιδρούν στο παιχνίδι και να αποθηκεύονται για την επόμενη φορά, χρησιμοποιήθηκε η τάξη “PlayerPrefs”. Μέσω της “PlayerPrefs” αποθηκεύονται στο σύστημα οι ρυθμίσεις, ώστε όταν ο χρήστης ανοίγει και πάλι το πρόγραμμα δεν χρειάζεται να τις ξανά αλλάξει.

```
public void Save()
{ PlayerPrefs.SetFloat("Sensitivity", mouseSensitivitySlider.value); }

private void Load()
{ mouseSensitivitySlider.value = PlayerPrefs.GetFloat("Sensitivity"); }
```

Πίνακας 2. Αποθήκευση και φόρτιση των προηγούμενων ρυθμίσεων.

Κάτι καινούριο για εμένα ήταν όταν χρειάστηκε να ασχοληθώ με τον ήχο μέσα στην διεπαφή του Unity. Ψάχνοντας έμαθα πως για να μπορούν να υπάρχουν επιλογές στις ρυθμίσεις για την αυξομείωση του ήχου θα έπρεπε να δημιουργηθεί ένας “AudioManager”. Ο AudioManager είναι ένας κώδικας που χρησιμοποιείται για την διαχείριση του ηχοσυστήματος σε ένα έργο της Unity. Με αυτό δίνεται η δυνατότητα του να παρέχεται ένα κεντρικό σύστημα ήχου για τον χειρισμό του σε όλους τους τομείς του παιχνιδιού. Στην προκειμένη περίπτωση, ήθελα να χωρίσω τους ήχους σε δύο κατηγορίες. Ένα, στους ήχους φόντου όπως η μουσική και τα κύμματα της θάλασσας και δύο, στα ηχητικά εφέ που είναι οι ήχοι της διάδρασης του παιχνιδιού και του μενού. Αυτό επιτεύχθηκε χρησιμοποιώντας τα “Switch Statements” του Unity C#. Τα “Switch Statements” είναι δομές ελέγχου που μπορούν

να χρησιμοποιηθούν για την επιλογή ενός από πολλών κομματιών κώδικα για να εκτελεστούν με βάση μίας τιμής μεταβλητή. Με λίγα λόγια είναι ένας μεγάλος κώδικας “if/else” αλλά αντί να ελέγχει πολλαπλές συνθήκες, ελέγχει μόνο μία. Με αυτόν τον τρόπο, στην προκειμένη περίπτωση δίνεται η δυνατότητα να χωριστούν οι ήχοι σε δύο “case” για τον ευκολότερη χειραγώγηση τους. Όταν πλέον οι ήχοι ήταν χωρισμένοι, υπήρχε η δυνατότητα του να τοποθετηθούν οι νέοι ήχοι που θα προστίθενται στο παιχνίδι, μέσα σε ένα από τα δύο “case”.

```
switch (s.audioType)
{
    case Sound.AudioTypes.soundEffect:
        s.source.outputAudioMixerGroup = soundEffectsMixerGroup;
        break;

    case Sound.AudioTypes.music:
        s.source.outputAudioMixerGroup = musicMixerGroup;
        break; }
}
```

Πίνακας 3. Χωρισμός των ήχων σε δύο κατηγορίες.

Όταν πλέον ξεκίνησα να δημιουργώ τους βασικούς μηχανισμούς του παιχνιδιού, ο πρώτος μηχανισμός που ήθελα να επιτύχω ήταν η διάδραση με τα αντικείμενα της σκηνής μέσω ενός μηχανισμού επιθεώρησης και τη μεταφορά τους. Ήθελα ο παίκτης να μπορεί να σηκώνει τα περισσότερα αντικείμενα της σκηνής και να μπορεί πατώντας το κουμπί “E” να φέρνει το αντικείμενο κοντά στην κάμερα για επιθεώρηση όλων των πλευρών του. Για να καταφέρω το αποτέλεσμα αυτό χρησιμοποίησα μία δομή “RaycastHit”. Το “RaycastHit” χρησιμοποιεί υποτυπώδης ακτίνες στο πρόγραμμα της Unity και ψάχνει για τα colliders των αντικειμένων. Με αυτόν τον τρόπο ορίζοντας ετικέτες (tags) στα αντικείμενα της σκηνής, μπορεί κανείς να χρησιμοποιήσει το RaycastHit ώστε να δημιουργήσει διαδράσεις με τα αντικείμενα αυτά. Χρησιμοποιώντας αυτήν την τεχνική έχω θέσει μερικά αντικείμενα με την ετικέτα “Object”. Στη συνέχεια, πατώντας το αριστερό κλικ του ποντικιού, έχω γράψει κώδικα, χρησιμοποιώντας την τάξη “transform”, που μεταφέρει το αντικείμενο σε ένα κενό (empty) αντικείμενο μπροστά στην κάμερα. Στη συνέχεια, παρατηρήθηκε πως όταν το αντικείμενο χτυπούσε σε collisions άλλων αντικειμένων, ξέφευγε από το κενό αντικείμενο που είχε τεθεί πρωτίστως. Το πρόβλημα αυτό λύθηκε χρησιμοποιώντας μία μέθοδο “Vector3.Distance”, το οποίο υπολογίζει την απόσταση δύο σημείων μέσα στον τρισδιάστατο χώρο. Με αυτό, δήλωσα πως όταν το αντικείμενο φεύγει σε μία απόσταση από το κενό αντικείμενο, να

επιστρέφει πίσω σε αυτό. Επίσης, κουνιώντας αντικείμενα που θα έπρεπε να δίνουν την αίσθηση του βάρους μέσα στη σκηνή, όπως κάποια βαρέλια, σύνδεσα τον παραπάνω κώδικα με τον κώδικα της κίνησης του χαρακτήρα, και με ένα bool περιόρισα την ταχύτητα του χαρακτήρα και την κίνηση του ποντικιού.

```
if (heldObj == null)
{
    RaycastHit hit;
    if (Physics.Raycast(transform.position, transform.TransformDirection(Vector3.forward), out
    hit, pickupRange))
    { PickupObject(hit.transform.gameObject);
      DragObject(hit.transform.gameObject); }
}
```

Πίνακας 4. Το RaycastHit σε χρήση.

```
if (heldObj != null && moveBack) //moveBack so the item will move back to holdArea only if
its true
{
    if(Vector3.Distance(heldObj.transform.position, holdArea.position) > 0.1f)
    {
        Vector3 moveDirection = (holdArea.position - heldObj.transform.position);
        heldObjRB.AddForce(moveDirection * pickupForce);
    }
}
```

Πίνακας 5. Το Vector3.Distance με το σύστημα επιστροφής σε χρήση.

Το RaycastHit επίσης χρησιμοποιείται και για το σύστημα επιθεώρησης των αντικειμένων. Με τον ίδιο παραπάνω τρόπο, μεταφέρει το αντικείμενο σε ένα κενό αντικείμενο μπροστά στην κάμερα, με μόνη διαφορά του να περιορίζει την κίνηση του χαρακτήρα και της κάμερας πλήρως, να αποθηκεύει την αρχική κατάσταση του αντικειμένου και να επιτρέπει στον παίκτη να γυρίζει το αντικείμενο με τη χρήση του ποντικιού. Όταν ο παίκτης τελειώνει με την επιθεώρηση, έχει τη δυνατότητα να επιστρέψει το αντικείμενο πίσω στην αρχική κατάσταση και να ενεργοποιήσει ξανά όλες τις κινήσεις του χαρακτήρα και της κάμερας.

Όταν πλέον ολοκλήρωσα και τον πρώτο μηχανισμό του παιχνιδιού, ήθελα να δίνονται στοιχεία στον χρήστη ώστε να καταλαβαίνει μέσω της διεπαφής του παιχνιδιού τι μπορεί να μεταφερθεί και τι μπορεί να επιθεωρηθεί. Για παράδειγμα, όταν ο παίκτης κοιτάει αντικείμενα με ετικέτα “Object”, τότε χρησιμοποιείται η μέθοδος “gameObject.SetActive()”, και έχοντας δηλωμένες δισδιάστατες εικόνες σαν αντικείμενα μέσα στον κώδικα, ενεργοποιούνται και απενεργοποιούνται αντίστοιχα. Για παράδειγμα, το προκαθορισμένο σταυρόνημα του παιχνιδιού είναι μία μεγάλη βούλα στο κέντρο της οθόνης, όταν σημαδεύεται ένα αντικείμενο που μπορεί ο παίκτης να διαδράσει μαζί του, τότε η εικόνα του σταυρονήματος αυτού απενεργοποιείται και ενεργοποιείται το σταυρόνημα που έχει μία εικόνα ενός ζωγραφισμένου γάντζου.

```
crosshair.gameObject.SetActive(true);  
crosshair2.gameObject.SetActive(false);
```

Πίνακας 6. Η χρήση του SetActive στο σταυρόνημα.

Αφού πλέον είχα τη βάση του παιχνιδιού, σκεφτόμουν το πως θα μπορούσα να εμπλουτίσω τη σκηνή με γρίφους. Ήθελα να υπάρχει η γενική ιδέα της χρήσης των αντικειμένων όπως τον πραγματικό κόσμο. Στους γρίφους που θα πρέπει ο παίκτης να τοποθετεί ένα αντικείμενο για να δημιουργηθεί μία αλληλεπίδραση, ήθελα να υπάρχει και η επιλογή του παίκτη να πάρει το αντικείμενο πίσω στα χέρια του. Ενώ αυτό είναι ρεαλιστικό και θυμίζει μία πραγματική σκηνή όπως θα συνέβαινε σε ένα δωμάτιο διαφυγής, στην πράξη σε μερικούς γρίφους η χρήση αυτής της λογικής ήταν αδέξια με αποτελέσματα που δεν με άφηναν ικανοποιημένο. Για αυτόν τον λόγο, αποφάσισα πως θα χρησιμοποιήσω αυτή την τεχνική σε γρίφους που ταίριαζε, ενώ σε άλλους γρίφους θα έπρεπε να βρω μία άλλη λύση. Η λογική όμως στον τομέα του προγραμματισμού παρέμεινε ίδια, αλλάζοντας μόνο διάφορες εντολές. Κατά κύριο λόγο χρησιμοποιήθηκε η δήλωση “if()”. Μέσω του “if()” σε κάθε κατάσταση ο κώδικας ελέγχει αν πληρούνται οι προϋποθέσεις για να ενεργοποιηθεί κάποια κίνηση ή για να ολοκληρωθεί μία διαδικασία. Για παράδειγμα, όταν χρειαζόταν να ανοίξει η πόρτα του κεντρικού δωματίου, οι προϋποθέσεις που θα έπρεπε να πληρούνται για να ενεργοποιηθεί η κίνηση της πόρτας είναι τέσσερις. Πρέπει δηλαδή τέσσερα αντικείμενα να βρίσκονται σε τέσσερα συγκεκριμένα σημεία. Αυτές οι προϋποθέσεις χρησιμοποιούσαν bool, τα οποία ενεργοποιούνταν όταν το αντίστοιχο αντικείμενο βρισκόταν στην αντίστοιχη σωστή θέση. Όταν τα τέσσερα bool του γρίφου ήταν πλέον ενεργοποιημένα, μία δήλωση “if()” ερχόταν σε χρήση και έδινε το αποτέλεσμα της λύσης του γρίφου.

```

if (gunCanOpen && knifeCanOpen && pipeCanOpen && hookCanOpen)
{
    impactFirstDoor.Play(); //plays a sound
    firstDoorAnim.Play("firstdoorrotation", 0, 0f); //door animation
}

```

Πίνακας 7. Τα τέσσερα ενεργοποιημένα bool παίζουν έναν ήχο και την κίνηση της πόρτας.

Παρόμοια χρήση του παραπάνω γίνεται και στη σκάλα του δεύτερου δωματίου του παιχνιδιού. Στον γρίφο αυτόν πρέπει να τοποθετηθούν δύο σανίδες που βρίσκονται μέσα στο παιχνίδι πάνω στην σκάλα για να μπορεί ο παίκτης να σκαρφαλώσει. Οι σανίδες στην προκειμένη περίπτωση δεν τοποθετούνται όπως τα αντικείμενα της πρώτης περίπτωσης, αντιθέτως γίνονται “Destroy()” και ενεργοποιούνται με “SetActive” δύο αόρατες σανίδες που βρίσκονται ήδη πάνω στην σκάλα, μαζί όμως με το “SetActive” ενεργοποιούνται και τα αντίστοιχα bool ή αντικείμενα που ευθύνονται στην λύση του γρίφου.

Για την αποφυγή της χρήσης της συνεχής καταστροφής αντικειμένων, κατά κύριο λόγο χρησιμοποιήθηκαν διάφορα “animations” που δημιουργήθηκαν μέσα στο πρόγραμμα του Unity. Όπως φαίνεται στην παραπάνω εικόνα 59, η χρήση των κινήσεων και των ήχων γίνεται με την εντολή “.Play();”.

Στο τέλος, για να υπάρχει μία πρόκληση σε όλη την εμπειρία δημιουργήθηκε ένα χρονόμετρο που μετράει αντίστροφα. Μέσω της εντολής “Time.deltaTime” στο “Update()” του κώδικα μειώνεται ο χρόνος κάθε δευτερόλεπτο, ενώ στην μέθοδο “DisplayTime()” που δημιουργήθηκε αργότερα, διαμορφώνεται το οπτικό κομμάτι για να φαίνεται το χρονόμετρο στην οθόνη σε λεπτά και δευτερόλεπτα. Αυτό γίνεται με το όρισμα “timeToDisplay”, καθώς μεταβιβάζει τον χρόνο που θα φαίνεται στην οθόνη. Εάν το “timeToDisplay” είναι μικρότερο του 0 ορίζει το “timeToDisplay” σε 0, διαφορετικά εάν είναι μεγαλύτερο αυξάνει το “timeToDisplay” κατά 1. Η μεταβλητή υπολογίζεται διαιρώντας με το νούμερο 60 για τα λεπτά και άλλα 60 για τα δευτερόλεπτα. Αργότερα με μία ιδιότητα “timerText.text” ορίζεται η εμφάνιση των λεπτών και δευτερολέπτων σε μορφή {0:00}:{1:00}. Στη συνέχεια, όταν ο χρόνος φτάνει στο μηδέν τότε εμφανίζεται μία εικόνα με κουμπιά που λέει “GameOver”, δίνοντας την επιλογή να επιστρέψεις στο μενού ή να κλείσεις τελείως το βιντεοπαιχνίδι.

```

void DisplayTime(float timeToDisplay)
{
    /*if (timeToDisplay < 0) //this if is for milliseconds too
    {
        timeToDisplay = 0;
    }*/

    if(timeToDisplay < 0)
    {
        timeToDisplay = 0;
    }
    else if (timeToDisplay > 0)
    {
        timeToDisplay += 1;
    }

    float minutes = Mathf.FloorToInt(timeToDisplay / 60);
    float seconds = Mathf.FloorToInt(timeToDisplay % 60);
    //float milliseconds = timeToDisplay % 1 * 1000; //this if is for milliseconds too

    timerText.text = string.Format("{0:00}:{1:00}", minutes, seconds/*milliseconds*/);
    //this if is for milliseconds too ADD :{2:000} too

```

Πίνακας 8. Ο κώδικας της εμφάνισης του χρονομέτρου.

4. Ήχος και η σχεδίαση ήχου.

Όπως σε μία κινηματογραφική ή σε μία κινουμένων σχεδίων ταινία ο ήχος είναι πλέον απαραίτητος για ένα σωστό αποτέλεσμα, έτσι και στα βιντεοπαιχνίδια είναι εξίσου σημαντικός. Λέγοντας ήχο, πέρα από τον μουσικό τομέα, εννοούμε και τα ηχητικά εφέ που θα χρησιμοποιηθούν μέσα στο έργο του δημιουργού. Η εικόνα και ο ήχος είναι συνδεδεμένα μεταξύ τους σε μία δέσμη που μοιάζει με χορό. Μερικές φορές είναι απαραίτητο αυτά τα δύο να πηγαίνουν μαζί, ενώ πολλές φορές μπορεί και τα δυο να “χορέψουν” λίγο ξεχωριστά πριν

ξανά ενωθούν¹⁵. Σε ένα βιντεοπαιχνίδι ο ήχος παίζει συνεχώς, συνδέοντας τη διάδραση του παίκτη με το περιβάλλον. Όταν βρίσκεσαι σε ένα πλοίο, ο ήχος της ατμόσφαιρας και ο θόρυβος πρέπει να σε προδιαθέτει για το συναίσθημα του συγκεκριμένου επιπέδου. Όταν ο παίκτης προχωρά, πρέπει να ακούγονται τα βήματα του. Όταν προοδεύει μέσα στο βιντεοπαιχνίδι, πρέπει ο ήχος να δείξει με ένα σύνθημα την επιβράβευση του. Όταν μία πόρτα μακριά έχει ανοίξει, πρέπει να δοθεί το σωστό σύνθημα που στα αυτιά του παίκτη θα θυμίζει το άνοιγμα της πόρτας ή μιας πύλης. Για τους παραπάνω λόγους, χωρίς ήχο ένα βιντεοπαιχνίδι, όσο απεγάδιαστες και να είναι οι υπόλοιπες τέχνες, θεωρείται μισό, γεγονός που μπορεί σχεδόν να παραβλεφθεί με τις υπόλοιπες τέχνες.

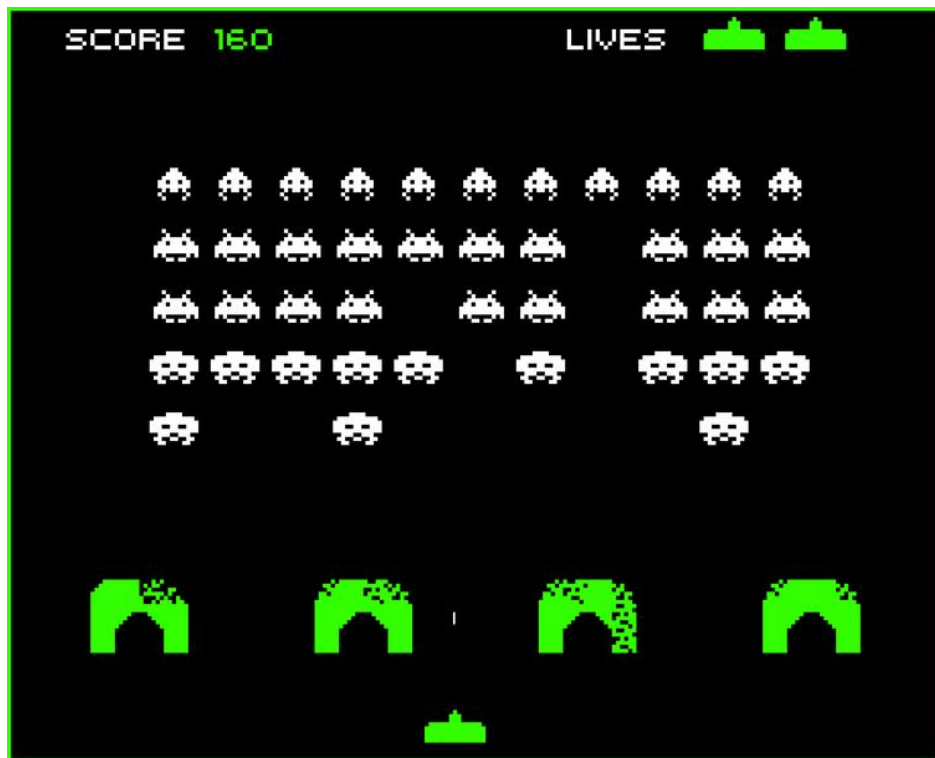
4.1 Ο ήχος στο “Davy Jones’ Locker.

Οι γνώσεις μου στο θέμα του ήχου είναι πολύ βασικές, γνωρίζοντας πως να τους επεξεργαστώ και το πως να κάνω κάποιες αλλαγές για την καλύτερη εμπειρία των παικτών. Στο “Davy Jones’ Locker” δούλεψα στον ήχο χρησιμοποιώντας το πρόγραμμα “Reaper”. Έκοψα και ένωσα ήχους, σε κάποιες περιπτώσεις ηχογράφησα δικούς μου ήχους, τους καθάρισα και άλλαξα τις εντάσεις με σκοπό όλοι να είναι σε ένα παρόμοιο επίπεδο. Σε κάθε διάδραση πρόσθεσα ήχους που ανταμείβουν τις κινήσεις του παίκτη με σκοπό να κατανοεί πως έλυσε τον γρίφο ή πως έκανε μία κίνηση σωστά. Κατά τη διάρκεια της εξερεύνησης έχω φτιάξει και χρησιμοποιήσει ήχους βημάτων που παίζουν σε τυχαία σειρά όσο ο χαρακτήρας κινείται στον χώρο. Επίσης, πρόσθεσα ένα ηχοτόπιο στο βάθος ενός εσωτερικού πλοίου ακούγοντας τον αέρα από το εξωτερικό του και κάποιους θορύβους τριξίματος, καθώς το συναίσθημα που ήθελα να πετύχω ήταν ένα κρύο και σκοτεινό περιβάλλον. Την περισσότερη επεξεργασία υπέστη ο ήχος του ηχοτοπίου αυτού, καθώς ο αέρας και οι πολλοί ήχοι άρχιζαν να γίνονται κουραστικοί μετά από κάποια ώρα παιχνιδιού. Η κύρια επεξεργασία ήταν η χρήση ενός “Equalizer” με σκοπό την σωστή ισορρόπηση των χαμηλών με τον υψηλών συχνοτήτων. Καθώς επίσης, επειδή έλειπε μία χαμηλή συχνότητα ώστε να ξεκουράζει τα αυτιά του παίκτη, πρόσθεσα ένα βουητό μπάσας νότας καθ’ όλη τη διάρκεια του ηχοτοπίου.

¹⁵ Murch, W., & Paine, F. (1981). Sound Design. *Journal of the University Film Association*, 33(4), 15–20.
<http://www.jstor.org/stable/20687577>

5. Βελτιστοποίηση των επιδόσεων του βιντεοπαιχνιδιού.

Ένας πολύ σημαντικός τομέας στην δημιουργία ενός βιντεοπαιχνιδιού, ειδικά στην περίπτωση των τρισδιάστατων γραφικών, είναι η επίδοση του τίτλου στις πλατφόρμες βιντεοπαιχνιδιών. Στις αρχές της βιομηχανίας των βιντεοπαιχνιδιών τα προβλήματα ήταν διαφορετικά από την σημερινή εποχή και ωθούσαν τους δημιουργούς να βρίσκουν λύσεις συνεχώς που στον τομέα πλέον είναι λυμένα εξ' αρχής, συνάμα, στη σημερινή εποχή έχουν δημιουργηθεί σειρές πολλών άλλων που έχουν παράγει πολλές επιπλέον ξεχωριστές θέσεις εργασίας σε άτομα που δουλεύουν πάνω στην βιομηχανία. Παραδείγματος χάρη, οι σκληροί δίσκοι της εποχής των πρώτων βιντεοπαιχνιδιών είχαν ελάχιστη χωρητικότητα σε σχέση με τους σημερινούς δίσκους των οικιακών υπολογιστών. Αυτό, σε συνδυασμό και με την τότε τεχνολογία περιόριζε τους δημιουργούς αναγκάζοντας τους να χρησιμοποιούν εικόνες μικρής ανάλυσης και μουσικές-ήχους απλοϊκούς με λίγα bit ώστε να μην υπερπληρούν τη διαθέσιμη χωρητικότητα των δίσκων της τότε εποχής. Ένα παράδοξο παράδειγμα είναι το βιντεοπαιχνίδι "Space Invaders" του 1978, στο οποίο όσο περισσότερο προόδευε και σκότωνε εξωγήινους ο παίκτης, τόσο περισσότερα δεδομένα διαγραφόταν από τον δίσκο κατά τη διάρκεια του παιχνιδιού, με αποτέλεσμα η πλατφόρμα να καταλήγει να εκτελεί το παιχνίδι σε ταχύτερους ρυθμούς, κάνοντας το σταδιακά δυσκολότερο, χωρίς αυτό να ήταν στα πρωταρχικά πλάνα των δημιουργών. Το παραπάνω παράδειγμα πλέον δεν εφαρμόζεται σε τόσο μεγάλο βαθμό στα τωρινά δεδομένα, όσο όμως εξελίσσεται η τεχνολογία των καρτών γραφικών, οι οθόνες ανεβαίνουν σε ανάλυση και οι απαιτήσεις των καταναλωτών μεγαλώνουν, δημιουργείται η ανάγκη της αύξησης της ανάλυσης των εικόνων που χρησιμοποιούνται για τα γραφικά σε μεγάλο βαθμό, με αποτέλεσμα νέοι τίτλοι βιντεοπαιχνιδιών να καταλήγουν να χρησιμοποιούν τεράστιους όγκους χωρητικότητας. Οι όγκοι αυτοί χρειάζεται στη συνέχεια ο καταναλωτής να τους κατεβάσει και να τους εγκαταστήσει για να μπορέσει να παίξει τον τίτλο. Το γεγονός αυτό αποθαρρύνει πολλούς καταναλωτές από το να αγοράσουν και να παίξουν κάποιους τίτλους, με αποτέλεσμα να είναι απαραίτητη η γνώση της χρυσής τομής αυτού του προβλήματος από τους δημιουργούς των βιντεοπαιχνιδιών της τωρινής εποχής.



Εικόνα 53. “Space Invaders” 1978.

Από την εποχή που ξεκίνησε η βιομηχανία των τρισδιάστατων γραφικών, οι απαιτήσεις των συστημάτων αυξήθηκαν για να μπορέσουν να εκτελέσουν τα καινούρια αυτά βιντεοπαιχνίδια. Ως τώρα έχει παρατηρηθεί πως οι απαιτήσεις των βιντεοπαιχνιδιών, για τις σωστές επιδόσεις στα συστήματα βιντεοπαιχνιδιών, ανεβαίνουν συνεχώς. Πολλά στούντιο προσπαθούν να επιτύχουν φαντασμαγορικά αποτελέσματα μέσα σε σύντομα χρονικά διαστήματα στους νέους τίτλους που δημιουργούν, με αποτέλεσμα να αγνοούν το πρόβλημα της επίδοσης τους και το παιχνίδι να δημοσιεύεται ημιτελές και με προβλήματα που το καθιστούν αδύνατο να ολοκληρωθεί. Οι σημαντικότεροι παράγοντες που επηρεάζουν τις επιδόσεις ενός βιντεοπαιχνιδιού είναι οι εξής. Τα πολύγωνα των τρισδιάστατων μοντέλων, οι ψηφιακές αναλύσεις των εικόνων που χρησιμοποιούνται για τις υφές των αντικειμένων, τα draw calls που χρησιμοποιεί η κάμερα κατά τη διάρκεια που εκτελείται το βιντεοπαιχνίδι και ο προγραμματισμός του κώδικα για τους μηχανισμούς του τίτλου. Όλα τα παραπάνω θα αναλυθούν στα επερχόμενα υποκεφάλαια.



Εικόνα 54. "Cyberpunk 2077" 2020. Ένα παιχνίδι που άφησε ιστορία όσο αναφορά τις κακές επιδόσεις του κατά τη διάρκεια του πρώτου καιρού της δημοσίευσής του.

5.1 Πολύγωνα - Υφές - Levels of Details.

Το πρώτο πράγμα που έρχεται στο μυαλό κάποιου όταν ένα βιντεοπαιχνίδι έχει πολύ ρεαλιστικά γραφικά, είναι το πως θα χρειαστεί ένα πολύ ισχυρό σύστημα για να μπορέσει να το εκτελέσει και να έχει ικανοποιητικές επιδόσεις για την ευχαρίστηση του παίκτη. Ενώ αυτό θεωρητικά εμπεριέχει μία δόση αλήθειας, δεν είναι το παν όσο αναφορά τα γραφικά. Ένα σημαντικό κομμάτι των τρισδιάστατων μοντέλων είναι τα πολύγωνα τους. Τα πολύγωνα επηρεάζουν τις επιδόσεις του επεξεργαστή και στη συνέχεια της κάρτας γραφικών και της μνήμης της. Η δουλειά του επεξεργαστή είναι να παίρνει την πληροφορία των αντικειμένων με τα πολύγωνα και να τη μεταφέρει στην κάρτα γραφικών όπου θα γίνουν και οι περισσότερες επεξεργασίες και θα αποθηκευθεί το αντικείμενο στη μνήμη της μέχρι να διαγραφεί¹⁶. Με βάση τα παραπάνω, ελαχιστοποιώντας τα πολύγωνα των αντικειμένων και γενικά της σκηνής απελευθερώνεις βάρος στη χρήση του επεξεργαστή και της κάρτας γραφικών ταυτόχρονα. Ενώ τα συστήματα την σημερινή εποχή μπορούν να αντέξουν εκατομμύρια πολύγωνα, αφαιρώντας λίγα χιλιάδες πολύγωνα από κάθε αντικείμενο, εν τέλει η διαφορά μπορεί να φέρει τεράστια διαφορά στο αποτέλεσμα. Για αυτόν ακριβώς τον λόγο

¹⁶ Caulfield, B. (2009, December 16). What's the Difference Between a CPU and a GPU? *Nvidia Blog*.

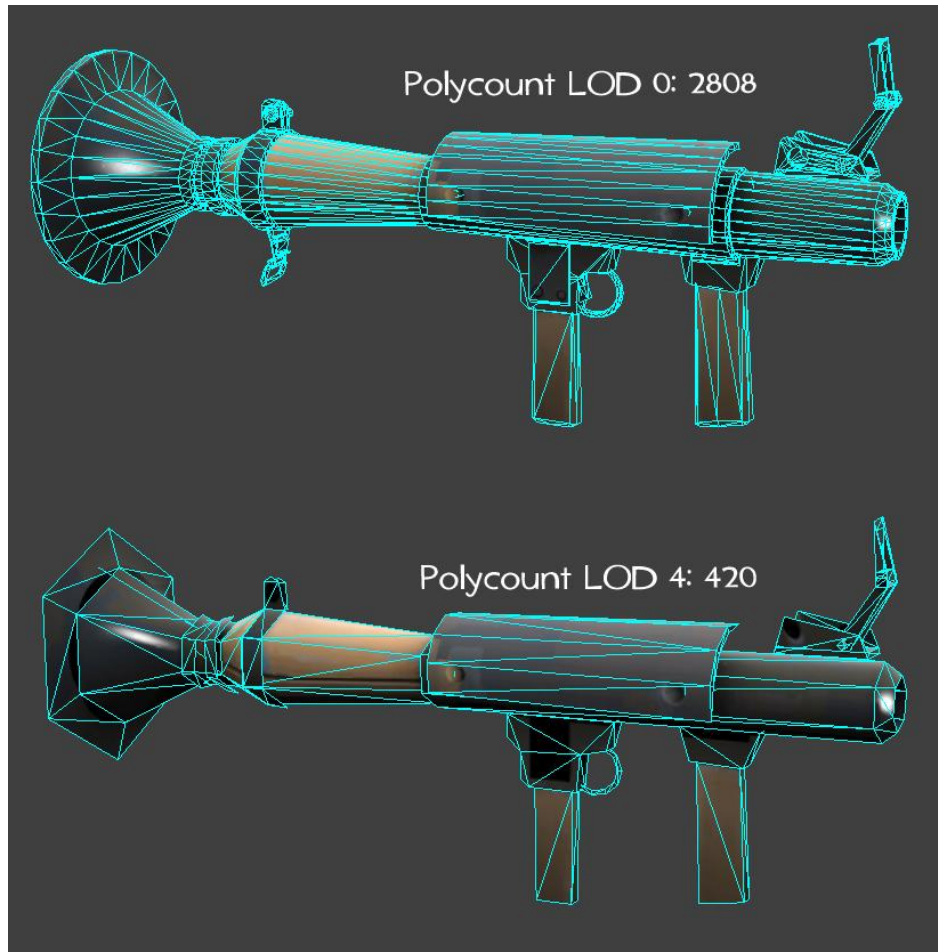
<https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/>

στη δημιουργία του “Davy Jones’ Locker” τα μοντέλα δημιουργήθηκαν έτσι ώστε να έχουν τα λιγότερα πιθανά πολύγωνα χωρίς όμως αυτό να γίνεται αντιληπτό από το μάτι του παίκτη. Ενώ πολύγωνα αντικειμένων που δεν βρίσκονται εύκολα στο οπτικό πεδίο της κάμερας, αποτελούνται από λιγότερα πολύγωνα για να μην επιβαρύνουν τα εξαρτήματα του υπολογιστή.

Όσο αναφορά την ανάλυση των εικόνων που χρησιμοποιούνται για τις υφές των αντικειμένων γίνεται η ίδια διαδικασία. Ο επεξεργαστής μεταφέρει την πληροφορία στην κάρτα γραφικών και στη συνέχεια η κάρτα γραφικών την επεξεργάζεται και την αποθηκεύει. Η διαφορά στη συγκεκριμένη περίπτωση είναι πως όσο μεγαλύτερη ανάλυση η εικόνα, τόσο περισσότερο χρόνο θα χρειαστεί η κάρτα γραφικών να επεξεργαστεί την πληροφορία, πράγμα που ελαφρύνει τον επεξεργαστή αλλά βαραίνει ακόμη περισσότερο την κάρτα γραφικών. Επιπλέον, τα μεγαμπάιτ των εικόνων αυτών αποθηκεύονται και στην μνήμη RAM του υπολογιστή. Η μνήμη RAM στην ουσία αποθηκεύει προσωρινά μεγάλες πληροφορίες από τον σκληρό δίσκο ενός υπολογιστή με σκοπό την άμεση χρήση τους, όταν στη συνέχεια δεν χρειάζονται πλέον αυτές οι πληροφορίες διαγράφονται από τη μνήμη RAM και απελευθερώνεται χώρος για τις επόμενες πληροφορίες. Όταν η εργασία εκείνης της στιγμής έχει ανάγκη μεγάλων ποσοστών μνήμης χωρίς να υπάρχει αυτό το ποσοστό στον υπολογιστή, τότε θα επιβραδύνεται η εκτέλεση της εργασίας μέχρι να απελευθερωθεί ξανά μνήμη από τη RAM. Για τον παραπάνω λόγο, όταν ένας υπολογιστής δεν έχει πλέον χώρο στη μνήμη RAM του, συντρίβεται και στην περίπτωση των windows βλέπουμε μία μπλε οθόνη με μήνυμα σφάλματος και στη συνέχεια το κλείσιμο ή της επανεκκίνησης του υπολογιστή. Όσο αναφορά τις αναλύσεις των εικόνων στο “Davy Jones’ Locker”, ακολουθήθηκε η ίδια λογική με τα πολύγωνα. Τα αντικείμενα που βρίσκονται κοντά στην κάμερα και ο χρήστης βρίσκεται σε συνεχή διάδραση μεταξύ τους έχουν μεγαλύτερη ανάλυση από εικόνες που ο χρήστης δεν δίνει μεγάλη σημασία κατά τη διάρκεια του παιχνιδιού, όπως για παράδειγμα κάποιο αντικείμενο ενός γρίφου έχει περισσότερα πολύγωνα και μεγαλύτερη ανάλυση από ότι τα κεριά που είναι τοποθετημένα στον χώρο.

Για τους παραπάνω λόγους, στα περισσότερα παιχνίδια οι δημιουργοί φτιάχνουν τα λεγόμενα “Levels of Details”, γνωστά και σαν “LOD”. Στην ουσία εξάγουν πολλές εκδοχές των αντικειμένων και των υφών τους. Από εκδοχές πολλών πολυγώνων και υψηλών αναλύσεων μέχρι και εκδοχές λίγων και χαμηλών. Η επιλογή τους γίνεται είτε από τον παίκτη στις ρυθμίσεις του βιντεοπαιχνιδιού, είτε από τους δημιουργούς ανάλογα με το πόσο κοντά ή

μακριά στα αντικείμενα βρίσκεται η κάμερα. Στο “Davy Jones’ Locker” δεν έχει χρησιμοποιηθεί αυτή η διαδικασία, καθώς είναι χρονοβόρα και πολλαπλασιάζει τη δουλειά που θα χρειαζόταν για να ολοκληρωθεί το αποτέλεσμα.



Εικόνα 55. Παράδειγμα κλίμακας LOD από forum της Unity¹⁷.

5.2 Draw Calls και Performance Jumps.

“Draw Calls” είναι ο αριθμός υλικών που τραβάει ο επεξεργαστής κατά τη διάρκεια της λειτουργίας του στο βιντεοπαιχνίδι. Ο αριθμός των draw calls πολλαπλασιάζεται και ανάλογα με τα φώτα και τις σκιές μέσα στη σκηνή. Εάν ένα αντικείμενο για παράδειγμα έχει πέντε υλικά επάνω του, αυτά αντιστοιχούν σε πέντε draw calls μόνο για το συγκεκριμένο αντικείμενο. Υπάρχουν διάφοροι τρόποι που βοηθούν στη μείωση αυτών των αριθμών.

¹⁷ calpolican. (2018). *To understand LOD, imagine you have a very detailed city in your game. There's a huge building in it and [Comment on "what is Lod(Level Of Detail)? why we us it?"]*.

<https://answers.unity.com/questions/1470351/what-is-lodlevel-of-detail-why-we-us-it.html>

Επίσης τα “Performance Jumps” είναι οι μικρές απότομες μειώσεις των επιδόσεων του παιχνιδιού, οι οποίες σχετίζονται άμεσα με τον τρόπο προγραμματισμού του βιντεοπαιχνιδιού¹⁸.

Για τη μείωση του αριθμού των draw calls είναι καλό να ακολουθούνται οι εξής κανόνες. Όταν πολλά αντικείμενα χρησιμοποιούν τα ίδια υλικά μέσα στο λογισμικό, πρέπει να ενώνονται μέσα σε παρτίδες γνωστές ως “batches” ώστε να διαβάζει το ένα υλικό μία φορά και όχι πολλαπλές φορές. Τα δισδιάστατα γραφικά των υλικών να μεταφράζονται σε μία εικόνα όπως αναφέραμε σε προηγούμενο κεφάλαιο μέσω της τεχνικής “baking” και στο παράδειγμα της εικόνας 45. Τα αντικείμενα με διαφορετικά υλικά δεν πρέπει να επικαλύπτουν το ένα στο άλλο. Να μην αλλάζει η κλίμακα ενός αντικειμένου. Και τέλος να χρησιμοποιούνται όσο το δυνατόν λιγότερα σωματίδια στα τρισδιάστατα οπτικά εφέ, καθώς ένα σωματίδιο αντιστοιχεί σε ένα draw call. Στο “Davy Jones’ Locker, όλα τα αντικείμενα που χρησιμοποιούν τα ίδια υλικά και είναι συνεχώς στάσιμα έχουν συγκεντρωθεί μαζί και έχουν χαρακτηριστεί ως στατικά μέσα στην πλατφόρμα του Unity, με σκοπό να μην διαβάζει το πρόγραμμα ξεχωριστά το ίδιο υλικό για την κάθε επιφάνεια. Σωματίδια και ειδικά εφέ δεν χρησιμοποιήθηκαν, καθώς το μόνο εφέ που έχει χρησιμοποιηθεί στον χώρο είναι οι φωτιές από το “Unity Assets Store” με τίτλο “Modular Medieval Lanterns”¹⁹, τα οποία είναι δισδιάστατα γραφικά με animations. Τέλος, χρησιμοποίησα την τεχνική “baking” στα φώτα της σκηνής, πράμα που μείωσε τα batches.

Όσο αναφορά τα performance jumps στον κώδικα του παιχνιδιού, πρέπει κατά κύριο λόγο να χρησιμοποιούνται όσο το δυνατόν λιγότερο οι εντολές “Instantiate()”, ειδικά σε περίπλοκα αντικείμενα και τα functions “Destroy()”, “gameObject.SetActiveRecursively()”, “Object.Find()”, “Resources.UnloadUnusedAssets()” και “GC.Collect()”. Στο “Davy Jones’ Locker” απέφυγα την χρήση των παραπάνω functions, με μόνη εξαίρεση το “Destroy()” για τη λύση κάποιων προβλημάτων και για την απελευθέρωση χώρου, χωρίς όμως αυτό να γίνεται σε πολλά αντικείμενα ταυτόχρονα για να μην επιβαρύνεται το σύστημα.

¹⁸ Karpovich, A. (2019). Improving the Performance of Unity 3d Mobile Games. *Elib.Psu.By*.

<https://elib.psu.by/handle/123456789/31515>

¹⁹ Voevodov, A. (2018). Modular Medieval Lanterns. Unity Asset Store.

<https://assetstore.unity.com/packages/3d/environments/historic/modular-medieval-lanterns-85527>

5.3 Προγραμματισμός και ήχος.

Ο προγραμματισμός ενός βιντεοπαιχνιδιού είναι πάντα μία πολύ χρονοβόρα διαδικασία. Ο κώδικας πρέπει να είναι αποτελεσματικός, χωρίς πολλές επαναλήψεις, χωρίς διαρροές πόρων, και γραμμένος έτσι ώστε να χρησιμοποιούνται εντολές που χρειάζονται την λιγότερη πιθανή προγραμματιστική ισχύ. Ωστόσο, καθώς από τη φύση του ο προγραμματισμός είναι μία πολύ χρονοβόρα διαδικασία, η καλύτερη επιλογή συνήθως είναι το να προχωρήσει η ανάπτυξη βιντεοπαιχνιδιού προγραμματίζοντας το έτσι ώστε να λειτουργεί, και στην συνέχεια να προχωράνε οι προγραμματιστές στο να καθαρογραφούν και να καθαρίζουν τον κώδικα τους από τα παραπάνω προβλήματα. Η εμπειρία ενός προγραμματιστή μπορεί να βοηθήσει στο να περιοριστεί η χρήση κάποιων βλαβερών συνηθειών στον κώδικα, όπως για παράδειγμα τις εντολές που είδαμε πρωτίστως που δημιουργούν performance jumps, αλλά πάντα θα χρειάζεται η διόρθωση στο τέλος. Έτσι, λοιπόν, στο “Davy Jones’ Locker” κατάφερα να εξαλείψω πλήρως κομμάτια που είχαν διαρροές πόρων και στο τελευταίο στάδιο της ολοκλήρωσης, αντικατέστησα εντολές “Destroy()” και “SetActive” με animations που δημιουργήθηκαν μέσα στο Unity.

Όσο αναφορά τον τομέα του ήχου, η χρήση ήχων που χρειάζονται μεγάλο ποσοστό μνήμης χρησιμοποιούν περισσότερους πόρους από το λογισμικό. Για αυτόν τον λόγο, οι ήχοι πρέπει να εξάγονται σε μορφές που διατηρούν μία ικανοποιητική ποιότητα για την εμπειρία του παίκτη, αλλά και την μείωση της μνήμης που θα χρειάζεται να χρησιμοποιηθεί από αυτούς. Για τους ήχους του “Davy Jones’ Locker” χρησιμοποιήθηκε η μορφή .mp3 που είναι επεξεργασμένοι ήχοι με πολύ μικρή χωρητικότητα και ικανοποιητικά αποτελέσματα.

5.4 Έλεγχος των επιδόσεων.

Για τον έλεγχο των επιδόσεων άνοιξα το παιχνίδι με δύο διαφορετικούς μετρητές για τα καρέ ανά δευτερόλεπτο. Το ένα ήταν του προγράμματος “Steam” και το δεύτερο του προγράμματος των Windows που υπάρχει ήδη μέσα στο σύστημα πατώντας τα κουμπιά “Win + G”. Το παιχνίδι λειτουργούσε κλειδωμένο στις 144 εικόνες ανά δευτερόλεπτο και στις δύο περιπτώσεις, καθώς αυτό είναι το όριο στα Hz της οθόνης μου. Επίσης, δεν παρατηρήθηκαν αιχμές αυξομειώσεις των αποδόσεων των εξαρτημάτων του υπολογιστή. Οι προδιαγραφές του προσωπικού μου ηλεκτρονικού υπολογιστή είναι οι εξής. Κάρτα γραφικών “GeForce

GTX 1080 8gb”, επεξεργαστής “Ryzen 5600x”, 32gb RAM στις συχνότητες 2133Hz και σκληρό δίσκο SSD M.2 NVMe.



Εικόνα 56. Ο μετρητής των καρέ ανά δευτερόλεπτο των Windows.

144 FPS Εικόνα 57. Ο μετρητής των καρέ ανά δευτερόλεπτο του “Steam”.

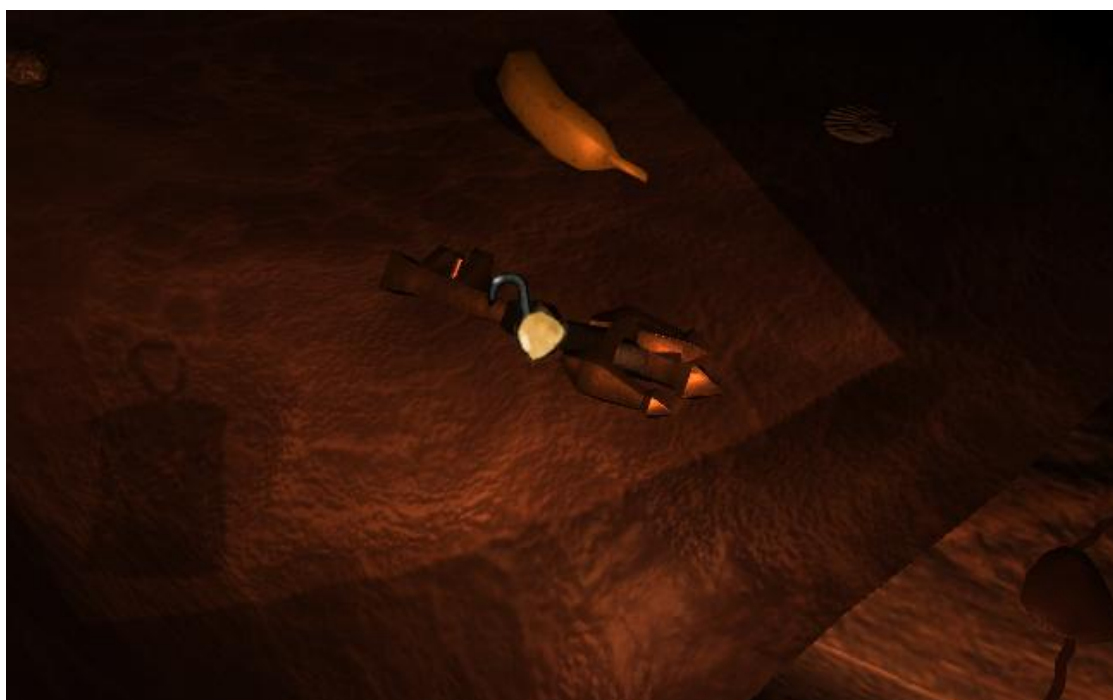
6. Το αποτέλεσμα του “Davy Jones’ Locker”.

Η κίνηση του παίκτη στον χώρο γίνεται με τα κουμπιά “W”, “A”, “S” και “D”, ενώ μπορεί να πραγματοποιήσει μικρά άλματα με το κουμπί “Space” και να σκύψει με το κουμπί “Ctrl”. Ο στόχος του παιχνιδιού είναι η εξερεύνηση στον χώρο και η επίλυση των γρίφων.



Εικόνα 58. Το πρώτο μεγάλο δωμάτιο. Davy Jones' Locker.

Όταν εμφανίζεται ο ζωγραφισμένος γάντζος στην οθόνη του παίκτη αντί της τελείας, αυτό σημαίνει πως ο παίκτης μπορεί με το αριστερό κλικ του ποντικιού να σηκώσει ή να μετακινήσει τα αντικείμενα του χώρου, ενώ με το πάτημα του αριστερού κλικ για δεύτερη φορά, πετάει ή παρατάει το αντικείμενο που κρατάει στα χέρια του.



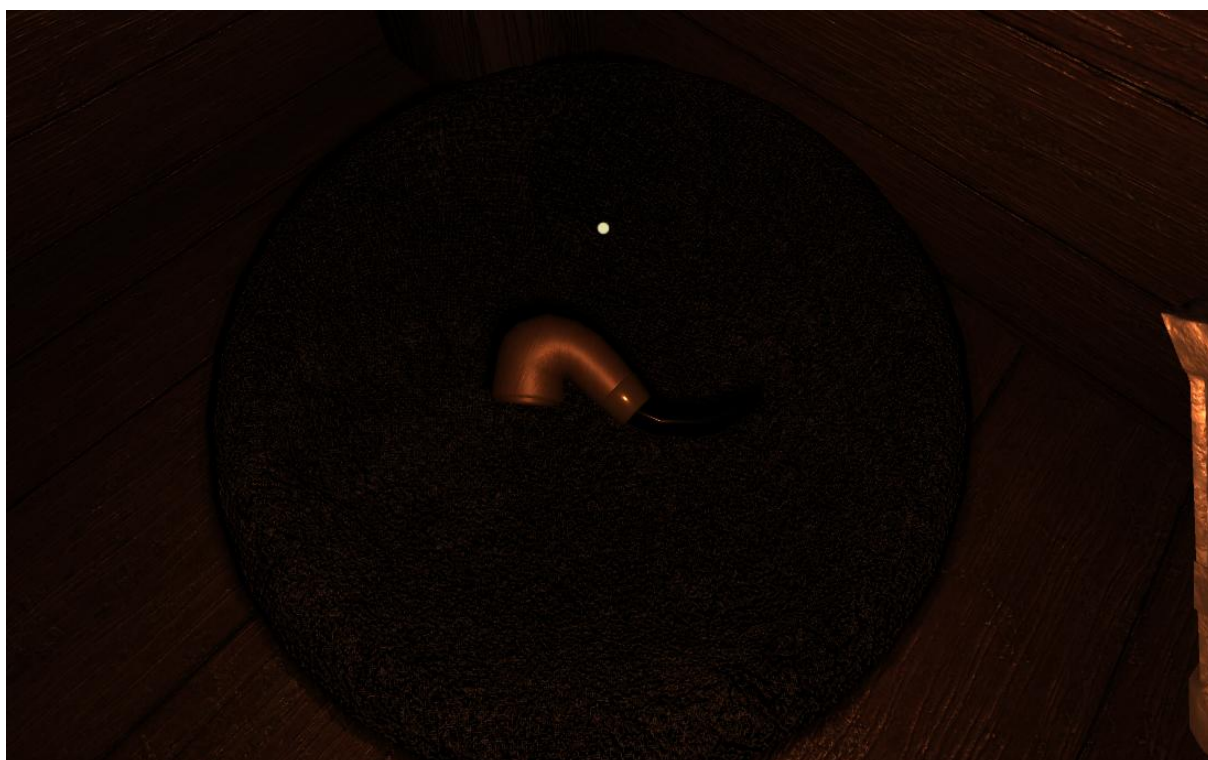
Εικόνα 59. Ο δείκτης "γάντζος". Davy Jones' Locker.

Διάφορες ενδείξεις που μπορούν να εμφανιστούν στην οθόνη του παίκτη είναι κείμενα που εμφανίζονται για να επιδείξουν στον παίκτη τις επιπλέον επιλογές κινήσεων που έχει στην προκειμένη περίπτωση. Για παράδειγμα, όταν ο παίκτης κοιτάει ένα αντικείμενο που μπορεί να εξετασθεί, του εμφανίζεται η ένδειξη “E to Examine”, που σημαίνει πως μπορεί να πατήσει το κουμπί “E” για να εξετάσει το αντικείμενο.



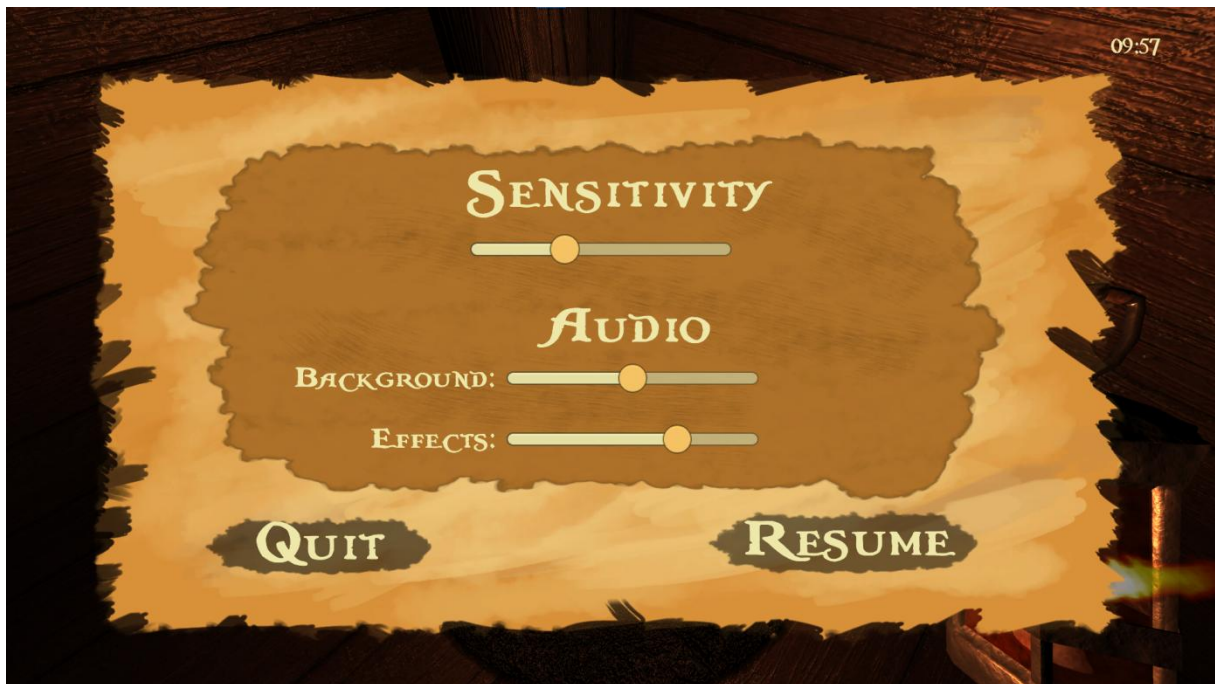
Εικόνα 60. Το κουμπί εξέτασης αντικειμένων. Davy Jones' Locker.

Οι γρίφοι του παιχνιδιού μπορούν να λυθούν με διάφορους τρόπους. Είτε πατώντας το κουμπί που ζητείται στην οθόνη όταν ο παίκτης βρίσκεται στο σωστό σημείο με το σωστό αντικείμενο, είτε αφήνοντας τα σωστά αντικείμενα στην σωστή τοποθεσία.



Εικόνα 61. Τοποθέτηση αντικειμένο στη σωστή τοποθεσία. Davy Jones' Locker.

Εάν ο παίκτης επιθυμεί να κάνει μία μικρή παύση του παιχνιδιού, μπορεί πατώντας το κουμπί “Escape” να ανοίξει το μενού του παιχνιδιού. Στο μενού αυτό πέρα από το πάγωμα του χρόνου, μπορεί ο παίκτης να αλλάξει τις ρυθμίσεις του παιχνιδιού ή και να βγει τελείως από την εφαρμογή.



Εικόνα 62. Το μενού κατά τη διάρκεια του παιχνιδιού.

Με τις παραπάνω γνώσεις, μπορεί πλέον ο παίκτης να περιηγηθεί στον χώρο του “Davy Jones’ Locker” και να επιλύσει τους γρίφους, με μόνο περιορισμό τα τριάντα λεπτά του χρονομέτρου που του μένουν.

Ο σύνδεσμος του “Davy Jones’ Locker”.

https://www.dropbox.com/sh/3lqwzog5b6k7k06/AAACo_H0hMegJImudqbDRxz4a

ΕΠΙΛΟΓΟΣ

7. Η εξέλιξη του “Davy Jones’ Locker”, οι γνώσεις που απέκτησα και το μέλλον.

Το “Davy Jones’ Locker” είναι μία εργασία που σκεφτόμουν πολλά χρόνια πως ήθελα να ολοκληρώσω, καθώς έχω πάθος από μικρός για τα βιντεοπαιχνίδια και είμαι λάτρης των πραγματικών “Escape Rooms”. Μέσω της δημιουργίας του απέκτησα πάρα πολλές γνώσεις πάνω στον τομέα της ανάπτυξης ενός βιντεοπαιχνιδιού που πρωτίστως δεν είχα την παραμικρή ιδέα. Η τεχνική μου της δημιουργίας των αντικειμένων μέχρι και τον προγραμματισμό των μηχανισμών άλλαξε ριζικά από την αρχή της πτυχιακής αυτής εργασίας μέχρι και την ολοκλήρωση της. Το αποτέλεσμα όμως με έχει αφήσει υπερήφανο, καθώς η δημιουργία ενός βιντεοπαιχνιδιού από το μηδέν μέχρι και το τέλος του είναι μία πάρα πολύ δύσκολη διαδικασία και χρονοβόρα, αφήνοντας ένα συναίσθημα πληρότητας.

Για το μέλλον του “Davy Jones’ Locker” σίγουρα θα δουλέψω στον εκσυγχρονισμό όλων όσων έχω κάνει. Θέλω να αναδημιουργήσω κάποια αντικείμενα και ίσως να αναπρογραμματίσω κάποια κομμάτια κώδικα σε μία πιο καθαρή μορφή. Θα ήθελα επίσης να προσθέσω επίπεδα γραφικών σε μορφή LOD, όπως επίσης και περισσότερα δωμάτια ή επίπεδα με επιπλέον διαφορετικές ιδέες γρίφων. Τέλος, θέλω να πειραματιστώ και με το είδος του παιχνιδιού, ίσως να προσθέσω κινδύνους και περισσότερες διαδράσεις που να κάνουν το παιχνίδι πιο ενδιαφέρον για να το παίξει ένα άτομο μόνο του ή και με παρέα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

1. Suits, B. (1967). What Is a Game? *Philosophy of Science*, 34(2), 148–156.
<http://www.jstor.org/stable/186102>
2. Rollings, A., & Adams, E. W. (2003). *Andrew Rollings and Ernest Adams on Game Design*. New Riders.
3. Wolf, M. J. P. (2012). *Encyclopedia of Video Games: M-Z: The Culture, Technology and Art of Gaming* (Vol. 1). Greenwood.
<https://books.google.gr/books?id=deBFx7QAwsQC&printsec=frontcover#v=onepage&q&f=false>
4. Crawford, C., & Crawford, C. (1984). *The Art of Computer Game Design: Reflections of a Master Game Designer*. McGraw-Hill Education.
5. Collins, K. (2008). *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. The MIT Press.
6. Pharr, M., Humphreys, G., & Jakob, W. (2017). *Physically Based Rendering: From Theory to Implementation*. Elsevier.
7. Flavell, L. (2010). *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design* (1st ed.). Springer.

Άλλα

1. Clearwater, D. (2011). What Defines Video Game Genre? Thinking about Genre Study after the Great Divide. *Loading. . .*, 5(8).
<https://journals.sfu.ca/loading/index.php/loading/article/view/67>

2. Huang, D. A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., & Niebles, J. C. (2018). What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/cvpr.2018.00769>

3. Lampton, C. (2018, April 7). *The Birth of 3D Gaming*. Sublime Confusion. <https://sublimeconfusionblog.wordpress.com/2018/04/07/the-birth-of-3d-gaming/>

4. Concept Art Empire. (2018, February 14). What is a Concept Artist? *Concept Art Empire*. <https://conceptartempire.com/what-is-concept-artist/>

5. Schonhorn, M. (1975). Defoe's Four Years Voyages of Capt. George Roberts and Ashton's Memorial. *Texas Studies in Literature and Language*, 17(1), 93–102. <http://www.jstor.org/stable/40754364>

6. HÖLTTÄ, L. (2018). *Effects of Art Styles on Video Game Narratives* [Thesis]. University of Turku. <https://www.utupub.fi/bitstream/handle/10024/145921/Effects%20of%20Art%20Style%20on%20Video%20Game%20Narratives.pdf?sequence=1>

7. Jacobson, J., & Lewis, M. (2002). GAME ENGINES IN SCIENTIFIC RESEARCH. *Communications of the ACM*, 45(1), 27–31. <https://www.cse.unr.edu/~sushil/class/gas/papers/GameAIp27-lewis.pdf>

8. Statt, N., & Roettgers, J. (2022, April 6). With Unreal Engine 5, Epic wants to transform how games and movies get made. *Protocol*. <https://www.protocol.com/newsletters/entertainment/epic-unreal-engine-5>

9. Yeo, A. (2022, March 1). *The secret to the incredible landscapes of "The Mandalorian": Epic Games' Unreal Engine*. Mashable. <https://mashable.com/video/the-mandalorian-epic-games-unreal-engine-set-landscape>

10. Murch, W., & Paine, F. (1981). Sound Design. *Journal of the University Film Association*, 33(4), 15–20. <http://www.jstor.org/stable/20687577>
11. Shiotsu, Y. (2021). C# vs. C++: Which Language is Right for Your Software Project? *Upwork*. <https://www.upwork.com/resources/c-sharp-vs-c-plus-plus#:~:text=C%2B%2B%20is%20much%20more%20lightweight,applications%20where%20performance%20is%20important.>
12. Great Learning. (2022, July 11). *C# vs C++: Differences and Similarities Between C# & C++ - Great Learning*. Great Learning Blog: Free Resources What Matters to Shape Your Career! <https://www.mygreatlearning.com/blog/c-sharp-vs-cpp/>
13. Kratochvil, B. (2015, January 7). Windows Store C++ for C# Developers: Understanding the Domain. *Microsoft*. <https://learn.microsoft.com/en-us/archive/msdn-magazine/2014/april/windows-store-windows-store-c-for-csharp-developers-understanding-the-domain>
14. Krassnigg, J., Born, F., Meder, J., Coulter, D., & Krassnigg, J. (2023, January 10). *PBR materials - Azure Remote Rendering*. Microsoft Learn. <https://learn.microsoft.com/en-us/azure/remote-rendering/overview/features/pbr-materials>
15. Wagner, A. (2020, April 8). *The Evolution of the Video Game Industry, Part 1: Hardware*. Wagner Road Capital Management. <https://www.wagnerroadcm.com/blog/2020/4/7/the-evolution-of-the-video-game-industry-part-1-hardware>
16. The Graphics Technology of Fallout 4. (2015, November 4). *Bethesda*. <https://bethesda.net/en/article/2Y37xeRPeUW0EgkgaKW8oA/the-graphics-technology-of-fallout-4>

17. Osborn, A. (2017, September 16). Why Ubisoft Ditched Unreal in Favor of Its Own Proprietary Tech. *IGN*. <https://www.ign.com/articles/2017/09/15/why-ubisoft-ditched-unreal-in-favor-of-its-own-proprietary-tech>
18. Prisco, J. (2017, November 13). How Japan changed video games forever. *CNN*. <https://edition.cnn.com/2017/11/12/asia/future-japan-videogame-landmarks/index.html>
19. Kochetov, K. (2018). *Modern approach to hard-surface modeling for games* [Bachelor's Thesis]. University of Applied Sciences. <https://www.theseus.fi/handle/10024/155459>
20. Marshall, D., Coyle, D., Wilson, S., & Callaghan, M. (2013). Games, Gameplay, and BCI: The State of the Art. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(2), 82–99. <https://doi.org/10.1109/tciaig.2013.2263555>
21. What Is Low Poly? Learn About Polygon Art for Video Games and More. (2021). *Domestika*. <https://www.domestika.org/en/blog/6206-what-is-low-poly-learn-about-polygon-art-for-video-games-and-more>
22. *The PBR Guide - Part 2 on Substance 3D Tutorials*. (n.d.). <https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-2>
23. Schäfer, H., Süßmuth, J., Denk, C., & Stamminger, M. (2012). Memory efficient light baking. *Computers & Graphics*, 36(3), 193–200. <https://doi.org/10.1016/j.cag.2011.12.001>
24. Stepp, E. (2020, March 2). Polycount? Understanding model efficiency. *Linkedin*. <https://www.linkedin.com/pulse/polycount-understanding-model-efficiency-emerson-stepp/>
25. *Modeling — Blender Manual*. (n.d.-b). <https://docs.blender.org/manual/en/latest/modeling/index.html>

26. Caulfield, B. (2009, December 16). What's the Difference Between a CPU and a GPU?
Nvidia Blog. <https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/>
27. Karpovich, A. (2019). Improving the Performance of Unity 3d Mobile Games.
Elib.Psu.By. <https://elib.psu.by/handle/123456789/31515>
28. Voevodov, A. (2018). Modular Medieval Lanterns. Unity Asset Store.
<https://assetstore.unity.com/packages/3d/environments/historic/modular-medieval-lanterns-85527>