

Γραφικά για videogames

Σχεδιασμός και μοντελοποίηση γραφικών για ηλεκτρονικά παιχνίδια

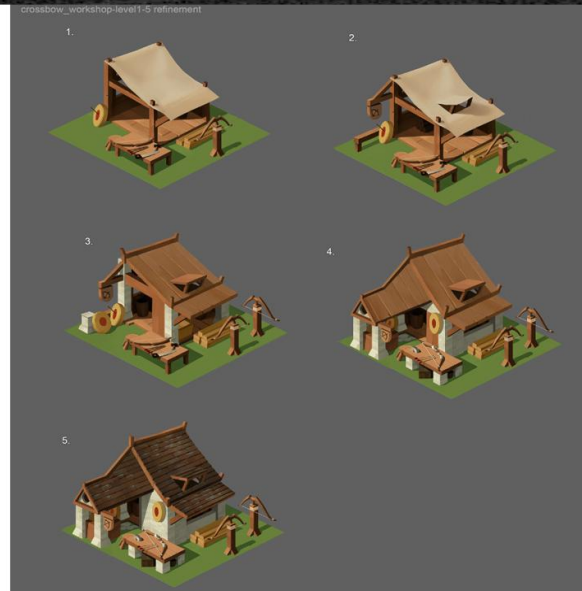
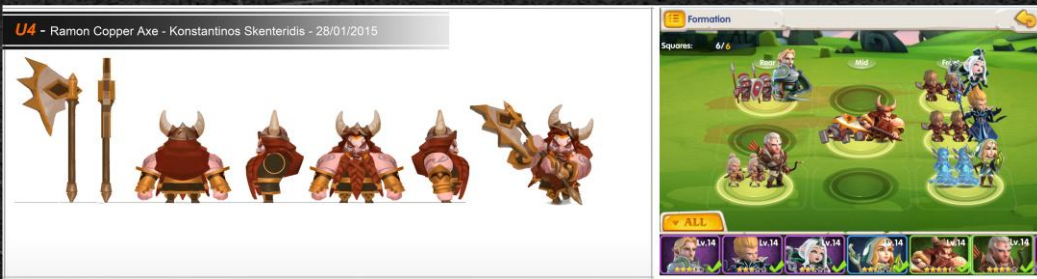
Σκεντερίδης Κωνσταντίνος

Concept art/illustration και 3d

-Τα γραφικά αποτελούν το μεγαλύτερο όγκο δουλειάς σε ένα παιχνίδι.

-Εννοείται πως και ο προγραμματισμός έχει δουλειά, αλλά εάν τα διάφορα υποσυστήματα έχουν υλοποιηθεί, μετά επαναλαμβάνονται, ενώ από γραφικά, υπάρχουν ανάγκες για πολλά διαφορετικά assets μέσα στο game.

-Η solo ανάπτυξη videogame συνήθως φαίνεται να αποτυγχάνει λόγο μεγαλεπήβολου score. Για παράδειγμα με αρκετή δουλειά και υπομονή δημιουργείται το πρώτο επίπεδο, και ενώ όλα τα mechanics προγραμματιστικά είναι υλοποιημένα, τα asset και το design που πρέπει να δημιουργηθεί για το υπόλοιπο content του παιχνιδιού είναι αποκαρδιωτικό.



Λογισμικά - Εργαλεία

-Τα βασικά εργαλεία/λογισμικά που χρησιμοποιώ στην δουλειά είναι:

- Photoshop
- Substance Painter
- Blender
- Unity game engine
- Zbrush

-Υπάρχουν και άλλα εργαλεία που χρησιμοποιώ, άλλα open source και free εναλλακτικές, αλλά κυρίως δουλεύω με αυτά



Βασικό Workflow δημιουργίας 3d asset

-Αρχικά κάνουμε την μοντελοποίηση σε ένα πρόγραμμα, πχ στο blender

-Υπάρχουν πολλές μέθοδοι αλλά το βασικό στοιχείο είναι πως εάν θέλουμε να καταλήξουμε με ένα μοντέλο χαμηλής πολυπλοκότητας πολυγώνων αλλά που να έχει πολλές λεπτομέρειες, τότε θα πρέπει να έχουμε μια εκδοχή του μοντέλου σε **low-poly** κατάσταση, και μια άλλη σε **high-poly** κατάσταση.

-Έχοντας αυτές τις 2 εκδοχές, μπορούμε σε ένα πρόγραμμα όπως το substance painter, να μεταφέρουμε [baking] τις λεπτομέρειες του high-poly στο low-poly μοντέλο, κάνοντας generate ένα normal map, το οποίο αποθηκεύει σε ένα 2d texture τις κατευθύνσεις που κοιτάει κάθε πολύγωνο[του high poly] [δίνοντας έτσι την ψευδαίσθηση ότι το μοντέλο μας έχει παραπάνω λεπτομέρεια/πολύγωνα , χωρίς να ισχύει κάτι τέτοιο [δεν είναι displacement]

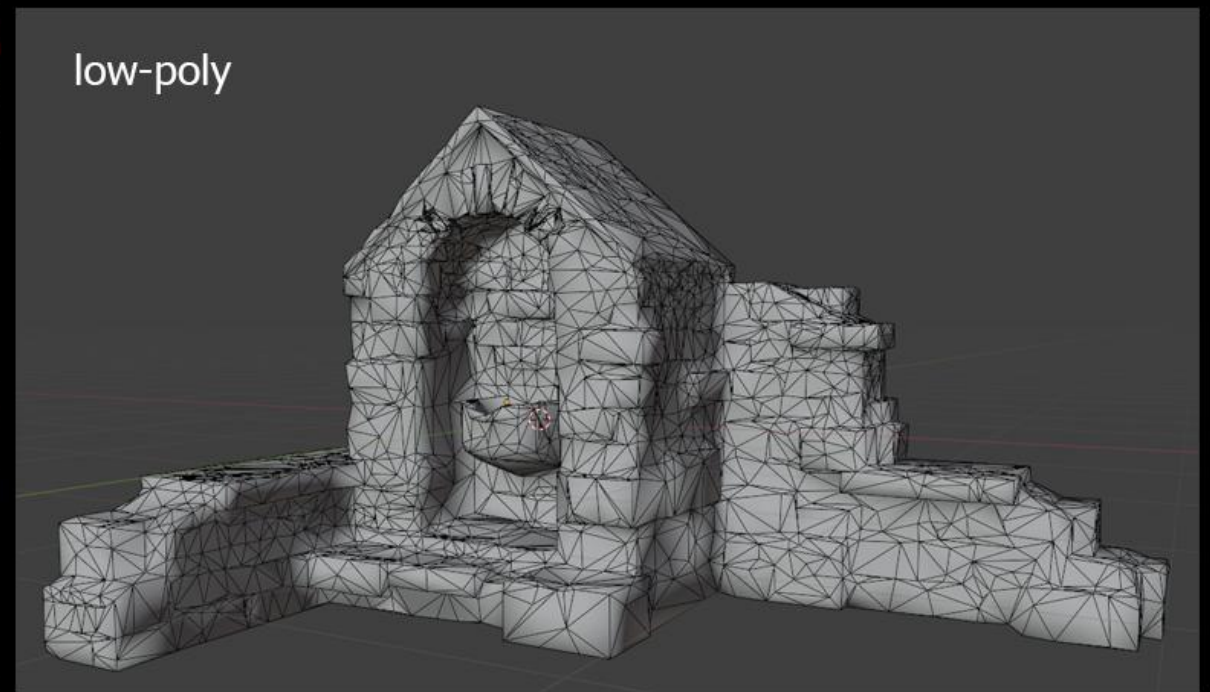
-Προγράμματα όπως το substance painter η ακόμα και το blender με το Texture painting υποσύστημά του, επιτρέπουν πλέον πολύ μεγαλύτερη χαλαρότητα στο πώς θα γίνει το un unwrapping. Στην ουσία για αντικείμενα στατικά που δεν πρόκειται να γίνουν animated, ένα απλό smart un project [αυτόματο unwrap στην ουσία] μας εξυπηρετεί μια χαρά

-Φυσικά γενικά χρειάζεται κάποια στιγμή κατανόηση του τι σημαίνει un unwrap ώστε όταν προκύψουν προβλήματα η χρειαζόμαστε να κάνουμε κάτι συγκεκριμένο, να μπορούμε να ρυθμίσουμε την κατάσταση και να αντιμετωπίσουμε το πρόβλημα με την αντίστοιχη λύση.

Βασικό Workflow δημιουργίας 3d asset

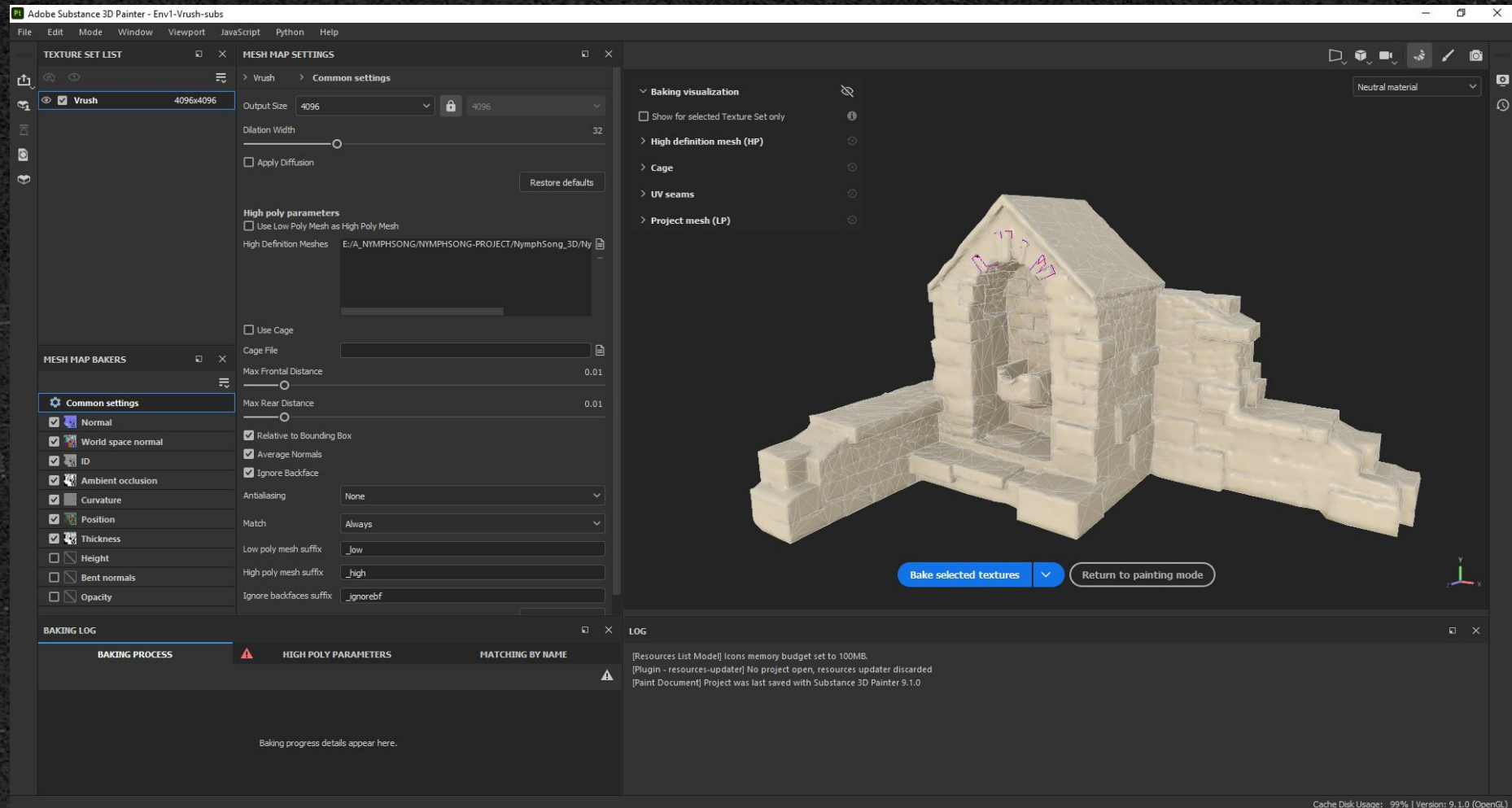
-Ένα παράδειγμα με μια παλιά βρύση

-Στο συγκεκριμένο παράδειγμα δούλεψα πρώτα την βρύση με λεπτομέρειες ώστε να φτιάξω το high-poly μοντέλο, και μετά χρησιμοποίησα έναν decimation modifier για να του μειώσω την πολυπλοκότητα, διατηρώντας αρκετά καλά τις άκρες/μεγάλες λεπτομέρειες. Το unwrap στην συνέχεια γίνεται στο low poly μοντέλο, το high είναι μόνο να χρησιμοποιηθεί κατά το baking [όπου μεταφέρουμε τις λεπτομέρειες από το ένα στο άλλο, δημιουργώντας το Normal map]



Βασικό Workflow δημιουργίας 3d asset

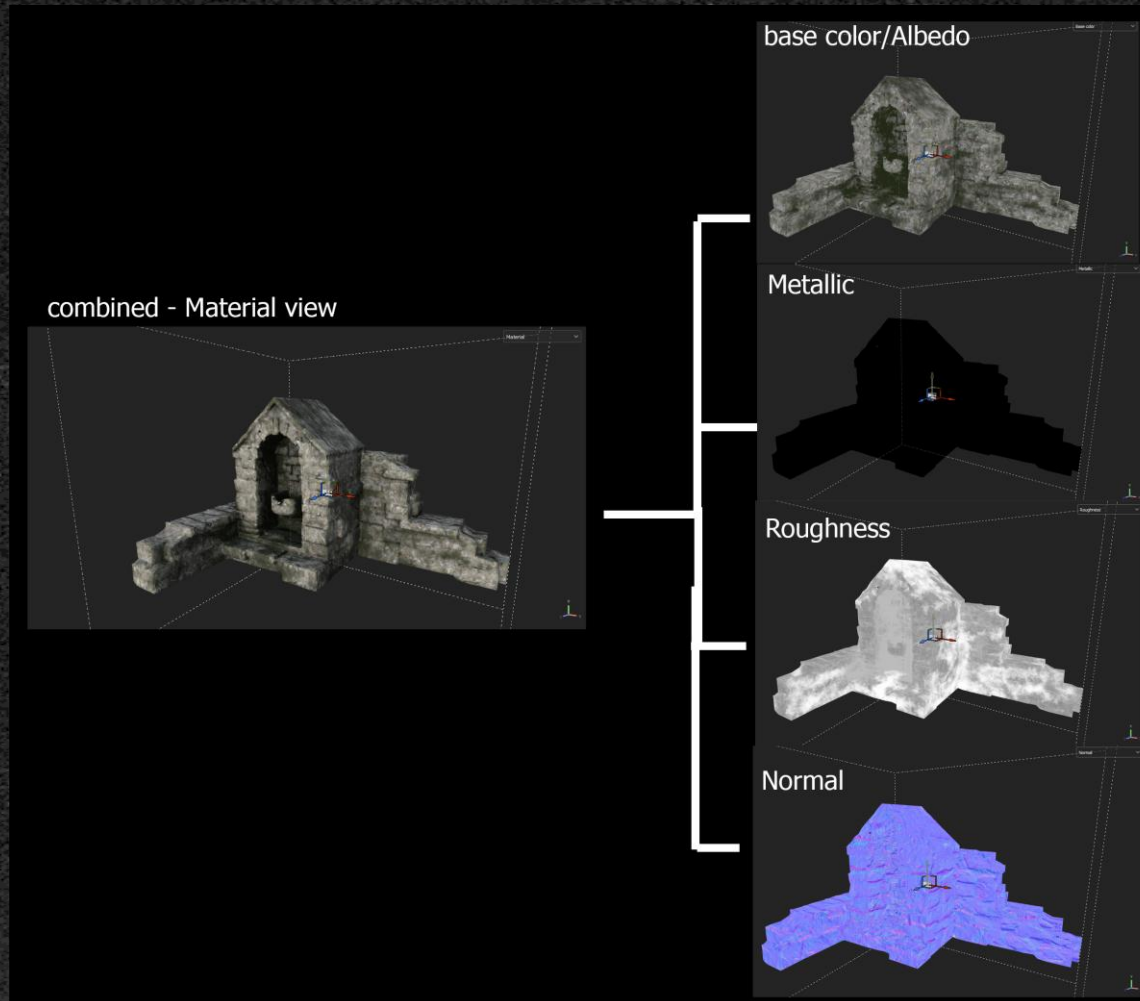
-Χρησιμοποιώντας τις 2 αυτές εκδοχές μέσα στο substance painter για παραγωγή normal map και άλλων map [Ambient occlusion, curvature etc.]. Όλα αυτά τα υπόλοιπα texture maps βοηθάνε πολύ στο texturing.



Βασικό Workflow δημιουργίας 3d asset

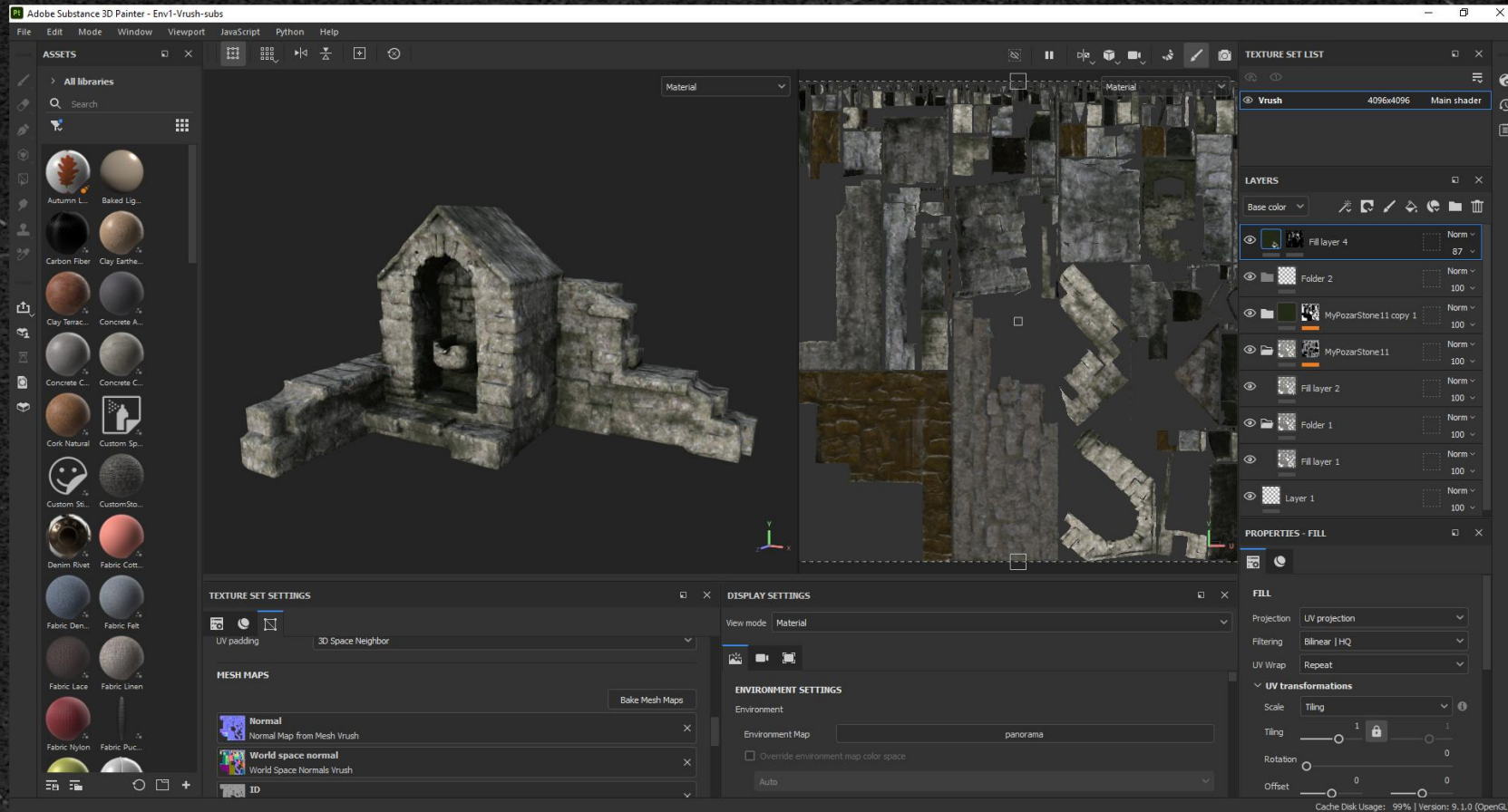
-Κατά το texturing στο substance painter δημιουργούμε στην ουσία μια σειρά από textures που θα χρησιμοποιηθούν για να στήσουμε το PBR material μας [Physically based rendering material]. Για ένα PBR material χρειαζόμαστε κυρίως:

- Base color
- Metallic
- Roughness
- Normal



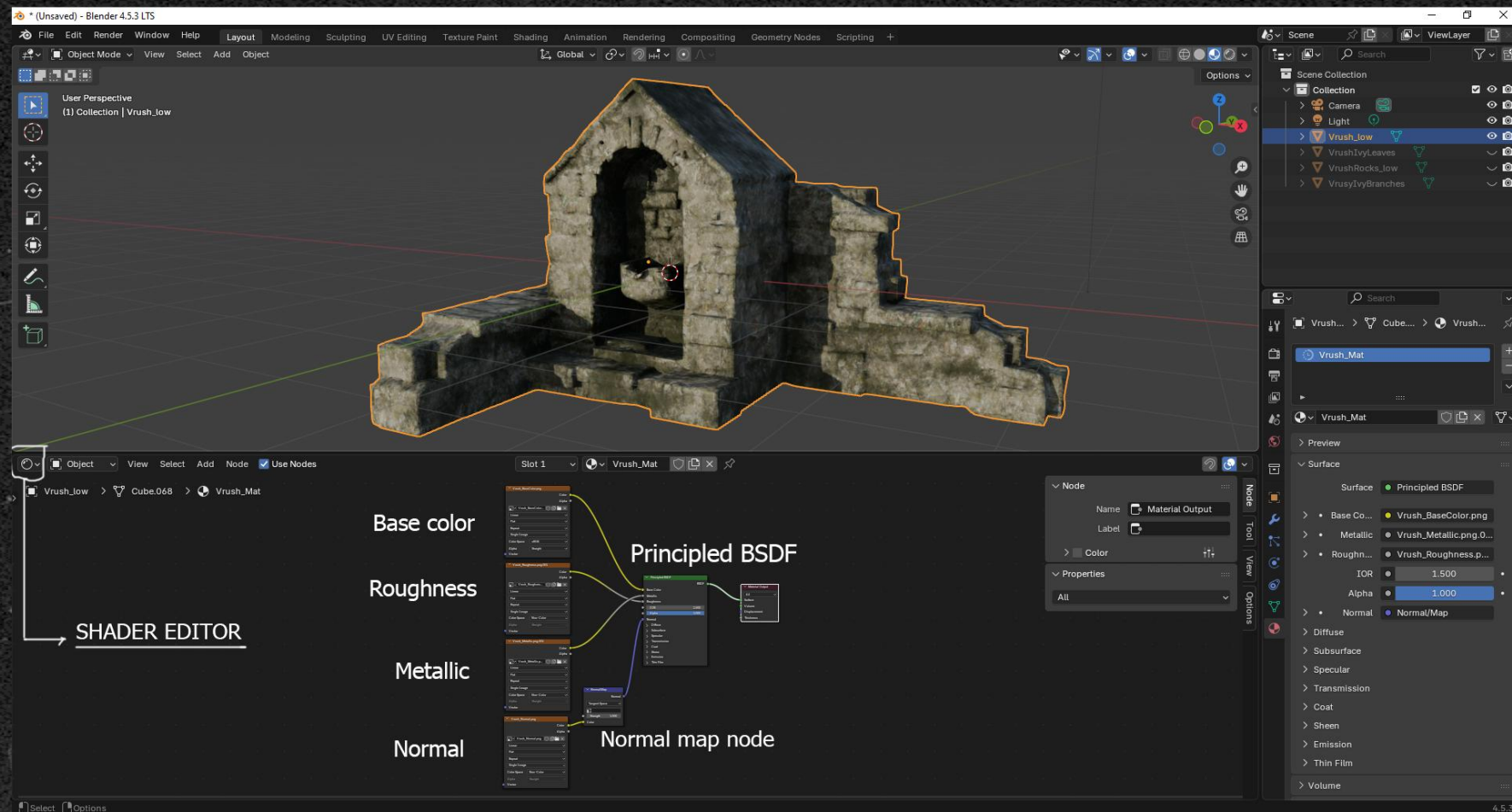
Βασικό Workflow δημιουργίας 3d asset

-Το texturing στο substance painter έχει την λογική που έχουν και προγράμματα όπως το Photoshop. Δουλεύουμε δηλαδή σε layers. Εδώ όμως μπορούμε να χρησιμοποιήσουμε στοιχεία που μας παρέχονται από το 3d μοντέλο, για να κάνουμε την διαδικασία του texturing πιο εύκολη και αποδοτική. Για παράδειγμα χρησιμοποιώντας το curvature map που δημιουργήθηκε κατά το baking [από high -> low] μπορούμε να βάψουμε βρωμιά/Dirt στις εγκοπές και στα σκοτεινά σημεία.



Βασικό Workflow δημιουργίας 3d asset

-Αφού τελειώσουμε το texturing, μετά μπορούμε να κάνουμε export τα textures του μοντέλου. Μπορούμε να επιλέξουμε κάποιο preset που εξάγει τα κατάλληλα texture για την εφαρμογή στην οποία θέλουμε να σετάρουμε το μοντέλο. Στην συγκεκριμένη περίπτωση εξάγουμε για το Blender [base color, metallic, roughness, normal].

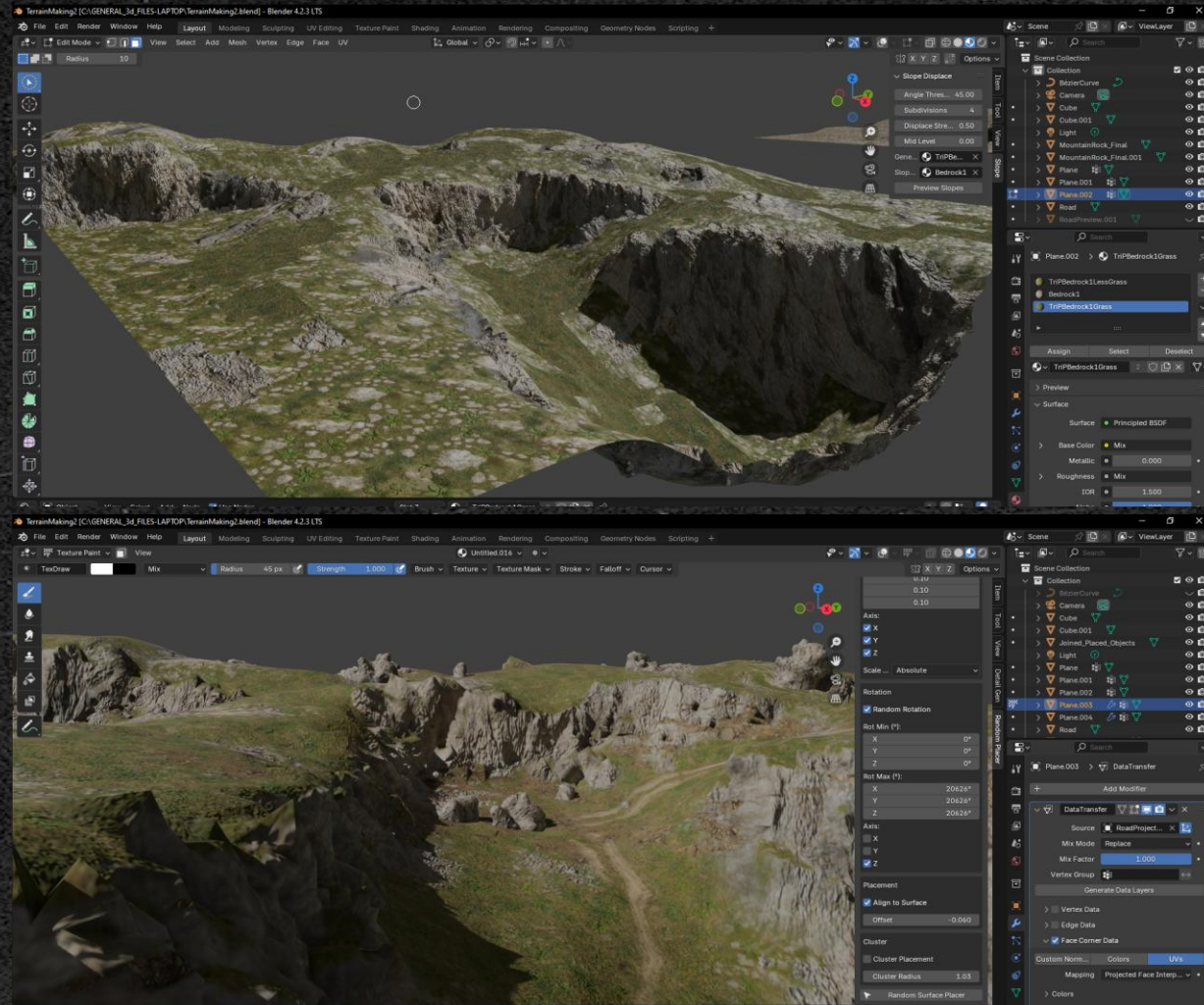


Πώς η γνώση των Γραφικών με Υπολογιστές βοηθάει στις βασικές καλλιτεχνικές ικανότητες

- Μαθαίνοντας τα βήματα που ακολουθεί ο υπολογιστής για να προσομοιώσει την αλληλεπίδραση φωτός με διάφορα υλικά [3d rendering] , μαθαίνουμε πώς και εμείς αντίστοιχα μπορούμε να προσεγγίσουμε την φωτοσκίαση [shading] κατά την ζωγραφική.
- Κατανοώντας πώς ρυθμίζονται τα materials και τις βασικές τους παραμέτρους, μπορούμε μετά κατά τον σχεδιασμό να κάνουμε ενσυνείδητες επιλογές στα αντικείμενα που σχεδιάζουμε [Roughness, Metallic, Base Color]
- Για παράδειγμα κατανοώντας πώς επηρεάζεται το υλικό όταν έχει περισσότερο η λιγότερο Roughness, μπορούμε να το αποδώσουμε αντίστοιχα και στο ζωγραφικό έργο. Είναι δύσκολο να κάνουμε κάτι κατά τον σχεδιασμό όταν δεν το γνωρίζουμε.
- Με την ίδια λογική, μαθαίνοντας πώς αναλύεται μια εικόνα σε διάφορα βασικά Render passes, μπορούμε να δημιουργήσουμε ένα workflow ζωγραφικής που μας παρέχει πιο ρεαλιστικά αποτελέσματα η ακόμα και να μας επιτρέψει να κάνουμε κάποιο ιδιαίτερο στυλ πειράζοντας κάποιο βήμα της διαδικασίας, εν γνώση μας όμως.
- Το πιο απλό σενάριο φωτισμού, θα ήταν αυτό ενός λευκού αντικειμένου [σφαίρα, κύβος] μέσα σε έναν απόλυτα μαύρο χώρο, φωτισμένου από 1 φωτεινή πηγή. Σε αυτήν την περίπτωση όσα πολύγωνα δεν λαμβάνουν φως από την φωτεινή πηγή, θα είναι μαύρα [γιατί δεν παίρνουν καθόλου φως]

Technical artist και εργαλεία για γρηγορότερη και αποδοτικότερη υλοποίηση γραφικών

Παραδείγματα εργαλείων



-Ενας Technical artist συνήθως αναλαμβάνει και την δημιουργία εργαλείων με τα οποία οι υπόλοιποι non-technical artists θα μπορέσουν να δημιουργήσουν πιο γρήγορα και αποδοτικότερα.

-Στο παράδειγμα δίπλα, φαίνεται ένα εργαλείο που καταλαβαίνει σε ένα μοντέλο ποιες περιοχές είναι απότομες πλαγιές/γρεμοί , και με το πάτημα ενός κουμπιού επιλέγει αυτές τις περιοχές [τα vertices τους] , εφαρμόζει subdivision ώστε να έχουν περισσότερα πολύγωνα εκεί τοπικά, και εφαρμόζει επάνω σε αυτά, displacement modifier με κάποιο noise κατάλληλο ώστε να κάνει τις περιοχές αυτές να μοιάζουν με βράχια.

-Επιπλέον κάνει assign σε αυτές τις περιοχές κάποια material που έχουν προετοιμαστεί, και έτσι πολύ γρήγορα έχουμε ένα αρκετά αληθοφανές terrain.

Technical artist και εργαλεία για γρηγορότερη και αποδοτικότερη υλοποίηση γραφικών

-Άλλα εργαλεία μπορεί να είναι:

-Scattering αντικειμένων επάνω σε επιφάνεια, στα σημεία όπου ο χρήστης κάνει κλίκ στην επιφάνεια

-Randomness στην τοποθέτηση αυτών των αντικειμένων [random scale, rotation]

-Procedural generation κτιρίων

-Procedural generation terrain

-Όλα τα procedural generation ανήκουν πάνω κάτω στην κατηγορία των technical artists

-Ειδικά για μικρές ομάδες development ,διάφορα τέτοια εργαλεία βοηθούν πολύ να μειωθεί ο όγκος δουλειάς

-Ακόμα και το απλό εργαλείο του τυχαίου scattering αντικειμένων σε μια επιφάνεια κάνει την δημιουργία ενός τοπίου πολύ πιο εύκολη. Θα ήθελε πάρα πολλές ώρες να κάτσει κάποιος manually να βάζει μια μία πέτρα σε ένα τοπίο.

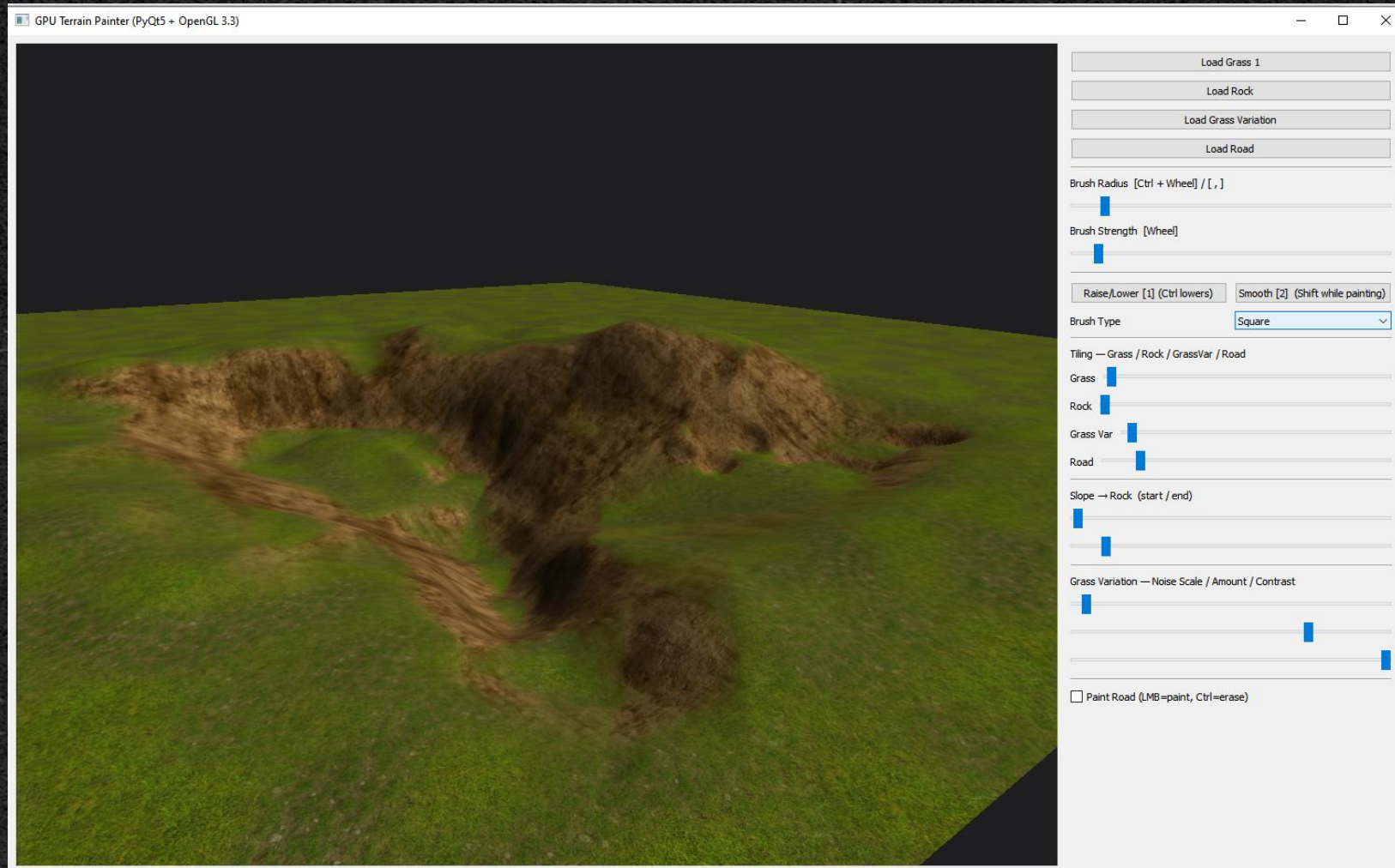
-Εργαλεία που βοηθούν στην δημιουργία αντικειμένων που είναι επαναλαμβανόμενα

-Staircase generator

-Complex Roof generator

Technical artist και εργαλεία για γρηγορότερη και αποδοτικότερη υλοποίηση γραφικών

-Stand alone εργαλείο terrain sculpting υλοποιημένο με Python



Basic optimization techniques

- Επειδή μας ενδιαφέρουν συνήθως real-time εφαρμογές [όπως τα παιχνίδια] είναι απαραίτητο να γνωρίζουμε και κάποιες βασικές τεχνικές για βελτιστοποίηση της απόδοσης
- Εν συντομία, το ζητούμενο που μας απασχολεί περισσότερο , είναι η μείωση των DRAW CALLS
- Τα draw calls μπορούμε να τα φανταστούμε ως οι εντολές που δίνει η CPU στην GPU ώστε να σχεδιάσει κάτι
- Όσο περισσότερα draw calls έχουμε, τόσο περισσότερο πέφτουν τα fps και γενικά απαιτεί χρόνο να γίνει render η σκηνή.
- Σκοπός είναι να συγχωνεύσουμε διάφορα στοιχεία ώστε με λίγα draw calls να έχουμε παρόμοιο αποτέλεσμα

ΤΙ ΑΥΞΑΝΕΙ ΤΑ DRAW CALLS

- Ξεχωριστά αντικείμενα
- Ξεχωριστά materials [ακόμα και αν ένα αντικείμενο έχει πολλά material -> draw calls όσα είναι τα materials]
- Διαφορετικοί shaders
- Ξεχωριστά textures

ΠΩΣ ΤΑ ΜΕΙΩΝΟΥΜΕ

- Mesh combining(manual or in-engine), Texture atlases, simple shaders, Material combining (πάει μαζί με το texture atlasing)

Επαγγελματικές προοπτικές

- Μια αναζήτηση να κάνει κάποιος στο ίντερνετ σχετικά με δουλειές στην βιομηχανία των videogame, είναι σίγουρο πως θα βρει πολλές αγγελίες.
- Με την εμφάνιση της τεχνητής νοημοσύνης όμως, φάνηκε ο πήχης να ανέβηκε ακόμα περισσότερο.
- Οι περισσότερες θέσεις φαίνεται να προσφέρονται σε άτομα με εμπειρία στον χώρο και λιγότερο σε junior
- Στον τομέα του concept art/illustration φαίνεται πως τουλάχιστον στην remote κατάσταση [remote freelance] δίνεται έμφαση στα art outsourcing studios [τα οποία έχουν πληθύνει]
- Ο ανταγωνισμός είναι ακόμα μεγαλύτερος από παλιότερα, επομένως η ανάγκη κάποιος να είναι καλός σε επαγγελματικό βαθμό και να έχει γερές γνώσεις σε διάφορα εργαλεία [blender, substance painter, unity , unreal κτλ] είναι πλέον αναμενόμενο.
- Το πάν στην βιομηχανία είναι το portfolio και όχι τόσο τα πτυχία
- Ακόμα και εργασιακή εμπειρία να μην έχει κάποιος, αν φαίνεται ξεκάθαρα από το portfolio του ότι μπορεί να κάνει την δουλειά όπως την κάνει ένας επαγγελματίας που ήδη έχουν στην ομάδα, τότε οι πιθανότητες να ενδιαφερθούν είναι μεγάλες.
- Ψάχνουν άτομα που να μπορούν να μπουν και να πάρουν μέρος στην παραγωγή χωρίς πολύ εκπαίδευση [ειδικά στις remote δουλειές]. Επομένως το portfolio πρέπει να δείχνει όσο το δυνατόν πιο κοντά σε portfolio άλλων επαγγελματιών [από θέμα ποιότητας και ικανοτήτων [ασχέτως του στυλ]]

Σύνδεσμοι

ARTSTATION - www.artstation.com/kingkostasart

INSTAGRAM - www.instagram.com/konstantinos_skenteridis_art/

EMAIL - kostaking@gmail.com