

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σχεδίαση και ανάπτυξη δισδιάστατου παιχνιδιού
δράσης με προοπτική, με την χρήση του περιβάλλοντος
Unity»



Του φοιτητή
Θωμά Ράτζιχα
Αρ. Μητρώου: 134008

Επιβλέπων
Παναγιώτης Αδαμίδης
Καθηγητής

Ημερομηνία
Φεβρουάριος 2022

Τίτλος Δ.Ε. Σχεδίαση και ανάπτυξη δισδιάστατου παιχνιδιού δράσης με προοπτική, με την χρήση του περιβάλλοντος Unity

Κωδικός Δ.Ε.: 20232

Ονοματεπώνυμο φοιτητή: Θωμάς Ράτζιχα

Ονοματεπώνυμο εισηγητή: Παναγιώτης Αδαμίδης

Ημερομηνία ανάληψης Δ.Ε. 08-11-2020

Ημερομηνία περάτωσης Δ.Ε. Φεβρουάριος 2022

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Θωμά Ράτζιχα που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

1 Πρόλογος

Η ανάπτυξη βιντεοπαιχνιδιών είναι μια πολύπλοκη διαδικασία στην οποία εμπλέκονται πολλές διαφορετικές ομάδες για κάθε διαφορετικό κομμάτι του παιχνιδιού (γραφικά, μουσική, λειτουργίες). Σκοπός της πτυχιακής αυτής είναι να δημιουργηθεί ένα παιχνίδι από ένα άτομο χρησιμοποιώντας διάφορα διαθέσιμα λογισμικά (Unity, Aseprite, Tilesetter) και να εξηγήσουμε τις ιδιαιτερότητες που μπορεί να έχει η διαδικασία αυτή. Μέσα από αυτήν την Πτυχιακή Εργασία ο αναγνώστης μπορεί να μάθει πως με την χρήση κάποιων παραπάνω προγραμμάτων ενισχύουμε την λειτουργικότητα του Unity και κάνουμε την διαδικασία υλοποίησης πιο εύκολη. Επίσης με μικρές παραμετροποιήσεις, το script για την δημιουργία τυχαίων επιπέδων μέσω κυτταρικού αυτόματου μπορεί να χρησιμοποιηθεί από τον οποιονδήποτε για να καλύψει τις ανάγκες του παιχνιδιού του.

2 Περίληψη

Η παρούσα πτυχιική εργασία ασχολείται με τον σχεδιασμό και την ανάπτυξη δισδιάστατου παιχνιδιού δράσης με προοπτική, με χρήση της μηχανής ανάπτυξης παιχνιδιών Unity. Χρησιμοποιείται η γλώσσα προγραμματισμού C# και τα λογισμικά Unity, Aseprite και Tilesetter. Η εργασία ξεκινάει με μια ιστορική αναδρομή για τα βιντεοπαιχνίδια, τους τύπους τους καθώς και τις ιδιαιτερότητες στην διαδικασία δημιουργίας τους με λίγες λεπτομέρειες περί των τεχνικών περιορισμών ανά τις δεκαετίες. Έπειτα αναλύονται τα λογισμικά τα οποία χρησιμοποιήθηκαν στην δημιουργία του παιχνιδιού με μερικές πληροφορίες για το περιβάλλον χρήσης και την λειτουργικότητα τους. Το παιχνίδι ακολουθεί έναν συγκεκριμένο χαρακτήρα ο οποίος δέχεται είσοδο από πληκτρολόγιο για την κίνηση του και την επίθεση προς τους αντιπάλους. Παράλληλα, εκμεταλλεύεται τον αλγόριθμο Game of Life του Conway για την δημιουργία τυχαία και δυναμικά εξελισσόμενων χαρτών. Αναλύεται η δημιουργία του παιχνιδιού και πιο συγκεκριμένα οι ξεχωριστές λειτουργίες, δίνοντας περισσότερη βάση στην υλοποίηση του κυτταρικού αυτόματου Game of Life του Conway και περιγράφοντας τις υπόλοιπες διαδικασίες και προσθήκες.

Design and development of a 2D Top-Down perspective video game using Unity environment.

Thomas Ratzicha

3 Abstract

This thesis focuses on the design and development of a 2D Top-Down perspective action video game using the Unity environment. C# is used as a programming language along with Unity, Aseprite and Tiled software tools. The thesis begins with a historical overview of video games, their genres and the peculiarities of the process of their creation with a few details about the technical limitations over the decades. Then, the software used to create the game is analyzed with some information about the user interface and their functionality. The game follows a specific character who receives input from the keyboard regarding his movement and attack on opponents. At the same time, it utilizes Conway's Game of Life algorithm to create random and dynamic evolving dungeon maps. The creation of the game and more specifically the individual functions are analyzed, further focusing on the implementation of Conway's cellular automata Game of Life as well as the other functions and additions to the game.

4 Περιεχόμενα

1	Πρόλογος.....	iii
2	Περίληψη.....	iv
3	Abstract	v
4	Περιεχόμενα	vi
	Κατάλογος Σχημάτων	viii
	Συντομογραφίες.....	x
	Κεφάλαιο 1ο: Εισαγωγή.....	1
	Κεφάλαιο 2ο: Ιστορία των Βιντεοπαιχνιδιών.....	2
2.1	Εισαγωγή.....	2
2.2	Ιστορική Αναδρομή.....	2
2.2.1	Δεκαετία του 1970.....	2
2.2.2	Δεκαετία του 1980.....	3
2.2.3	Δεκαετία του 1990.....	4
2.2.4	Δεκαετία του 2000.....	5
2.2.5	Δεκαετία του 2010.....	6
2.3	Τύποι παιχνιδιών	6
2.3.1	Παιχνίδια Δράσης.....	7
2.3.2	Παιχνίδια Δράσης-Περιπέτειας	7
2.3.3	Παιχνίδια Περιπέτειας.....	8
2.3.4	Παιχνίδια Ρόλων.....	8
2.3.5	Παιχνίδια Προσομοίωσης.....	9
2.3.6	Παιχνίδια Στρατηγικής.....	10
2.3.7	Παιχνίδια Αθλημάτων	10
	Κεφάλαιο 3ο: Εργαλεία που Χρησιμοποιήθηκαν.....	12
3.1	Unity.....	12
3.1.1	Πλεονεκτήματα του Unity.....	13
3.1.2	Το περιβάλλον του Unity	13
3.2	Aseprite	27
3.2.1	Διαδικασία σχεδιασμού.....	27

3.2.2	Animation.....	29
3.2.3	Αποθήκευση και εξαγωγή	29
3.3	Tilesetter.....	30
3.3.1	Tiles και Sources	30
3.3.2	Επεξεργασία πλακιδίων.....	31
3.3.3	Δημιουργία πλακιδίων.....	32
3.3.4	Συμπεριφορά πλακιδίων.....	32
3.3.5	Επεξεργασία χάρτη.....	34
3.3.6	Εξαγωγή project	34
Κεφάλαιο 4ο:	Διαδικασία Ανάπτυξης.....	36
4.1	Επεξήγηση Παιχνιδιού	36
4.2	Ανάλυση λειτουργιών	37
6	Συμπεράσματα και προτάσεις για βελτίωση	48
7	Βιβλιογραφία.....	49

Κατάλογος Σχημάτων

Σχήμα 1: Richter F 2020, Gaming: The Most Lucrative Entertainment Industry By Far.	2
Σχήμα 2: Pong Arcade Machine.....	3
Σχήμα 3: Nintendo Entertainment System, NES.....	4
Σχήμα 4: Σύγκριση μεταξύ Final Fantasy VII(1998, PS1) και Final Fantasy X(2001,PS2).....	5
Σχήμα 5: Το λογότυπο του Unity	12
Σχήμα 6: Το Unity στην παγκόσμια αγορά μηχανών παιχνιδιών (2019) [4]	13
Σχήμα 7: Περιβάλλον του Unity	14
Σχήμα 8: Scene View	14
Σχήμα 9: Game View	15
Σχήμα 10: Προκαθορισμένο Παράθυρο Ιεραρχίας	15
Σχήμα 11:Αναπαράσταση Parenting	16
Σχήμα 12: Inspector window.....	16
Σχήμα 13: Inspector window για ένα GameObject	17
Σχήμα 14: Animator window	18
Σχήμα 15: Animation System.....	18
Σχήμα 16: Μενού δημιουργία Sprites	19
Σχήμα 17: Rig Μενού.....	20
Σχήμα 18: Bone Renderer	21
Σχήμα 19: Blend Tree.....	22
Σχήμα 20: Μενού κόμβων Blend Tree	22
Σχήμα 21: 2D Blend Tree.....	22
Σχήμα 22: Δημιουργία Tilemap	23
Σχήμα 23: Tilemap window	23
Σχήμα 24: Grid window	23
Σχήμα 25: Tile Palette window	24
Σχήμα 26: Active Brush Inspector	24
Σχήμα 27: Ιδιότητες των Rule Tiles	26
Σχήμα 28: Animated Tile	27
Σχήμα 29: Αρχική σελίδα Sprite	28
Σχήμα 30: Εισαγωγή παραμέτρων New Sprite	28
Σχήμα 31: Aseprite Timeline	29
Σχήμα 32: Αποθήκευση αρχείου	29
Σχήμα 33: Διάκριση και συνεργασία των Tiles και Sources.....	30
Σχήμα 34: Λειτουργία Tiles και Sources στη δημιουργία σετ πλακιδίων.....	30
Σχήμα 35: Δημιουργία πλακιδίων	32
Σχήμα 36: Λειτουργία Relations	33
Σχήμα 37: Λειτουργία Relations 2	33
Σχήμα 38: Λειτουργία Relations 3	33
Σχήμα 39: Επεξεργασία χάρτη	34
Σχήμα 40: Εξαγωγή Tiles.....	35
Σχήμα 41: Εξαγωγή Maps.....	35
Σχήμα 42 Use Case Scenario μιας εκτέλεσης του παιχνιδιού	36
Σχήμα 43 Αρχική οθόνη παιχνιδιού	37
Σχήμα 44: Παράμετροι του Tile Automator.....	38

Σχήμα 45: Παράδειγμα χάρτη με Ini Chance 30, Birth 3 Death 3 και 7 επαναλήψεις(Αριστερά) και Ini chance 50, Birth 4, Death 3 και 3 επαναλήψεις(Δεξιά).....	39
Σχήμα 46: Idle Blend Tree Rules	41
Σχήμα 47: Αντίπαλοι.....	42
Σχήμα 48: Αριστερά, κλειστό collider κεντραρισμένο στον χαρακτήρα, Δεξιά αναμμένο collider με φορά προς την φορά επίθεσης.....	42
Σχήμα 49 Aseprite tiles	43
Σχήμα 50 Tilesetter setup	43
Σχήμα 51 Διαδικασία Tilesetter	44
Σχήμα 52 Διαδικασία Tilesetter 2	44
Σχήμα 53 Εξαγωγή σε Unity	44
Σχήμα 54 Τελικοί αυτόματοι κανόνες Tiles.....	45
Σχήμα 55 Πίστα χωρίς ενισχύσεις γραφικών	45
Σχήμα 56 Πίστα με ενισχύσεις γραφικών	46
Σχήμα 57 Μια εικόνα μαζί με το αντίστοιχο normal map της	46
Σχήμα 58 Normal Sphere	47
Σχήμα 59 Χωρίς χρήση Normal Maps (Αριστερά), Με χρήση normal maps (Δεξιά)	47

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή
M.O.B.A	Multiplayer Online Battle Arena
R.P.G	Role Playing Game
R.T.S	Real Time Strategy
4X	Explore, Expand, Exploit, Exterminate (Τύπος παιχνιδιού)

Κεφάλαιο 1ο: Εισαγωγή

Η παρούσα πτυχιακή έχει ως σκοπό να αναλύσει τους τρόπους, τις ιδιαιτερότητες και τις δυσκολίες δημιουργίας ενός παιχνιδιού από ένα άτομο το οποίο είναι υπεύθυνο τόσο για το κομμάτι του κώδικα και των λειτουργιών όσο και για το κομμάτι των γραφικών.

Ξεκινώντας, γίνεται μια ανάλυση πορείας της βιομηχανίας των βιντεοπαιχνιδιών στο χρόνο (ανά δεκαετία) αναφέροντας σημαντικά γεγονότα μέσα στο χρονοόπως είναι σήμερα. Ακολουθεί μια εξήγηση των διαφόρων τύπων παιχνιδιών που υπάρχουν δίνοντας μερικά παραδείγματα για το ποιο είναι το κύριο στοιχείο τους καθώς και μερικούς αντιπροσωπευτικούς τίτλους παιχνιδιών.

Το επόμενο κεφάλαιο δίνει βάση στην ανάλυση των προγραμμάτων τα οποία χρησιμοποιήθηκαν (Unity, Aseprite, Tilesetter) για την δημιουργία του παιχνιδιού. Μεγαλύτερη βάση δίνεται στο Unity καθώς είναι και το λογισμικό που χρησιμοποιήθηκε εκτενέστερα. Δίνεται βάθος στο περιβάλλον και τις λειτουργίες που παρέχει. Αντιστοίχως αλλά με λιγότερη έμφαση αναλύεται και η λειτουργία του Aseprite και του Tilesetter, κυρίως για τις λειτουργίες για τις οποίες χρησιμοποιήθηκαν.

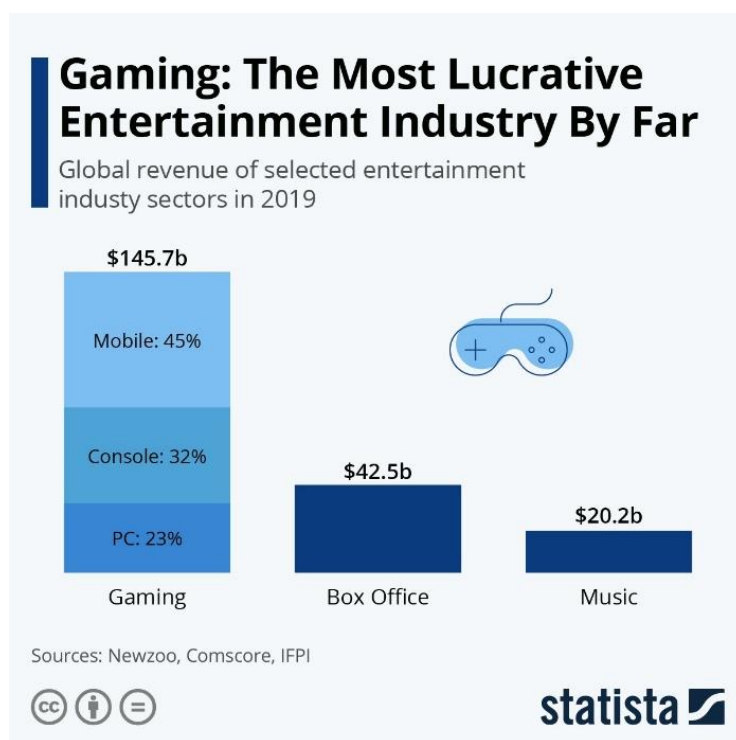
Το 4ο κεφάλαιο εμβαθύνει στην δημιουργία του ίδιου του παιχνιδιού. Δίνεται ένα διάγραμμα ροής του παιχνιδιού μαζί με μια ανάλυση του σκοπού και των περιοχών που υπάρχουν. Αναλύονται στην συνέχεια οι κύριες λειτουργίες του παιχνιδιού με κύρια έμφαση στο κομμάτι της δημιουργίας τυχαίων σπηλιών με την χρήση του αλγορίθμου Game of Life του Conway. Εξηγείται η παραμετροποίηση της λειτουργίας του αλγορίθμου για να επιτευχθεί ο σκοπός και ύστερα γίνεται ανάλυση των υπολοίπων λειτουργιών καθώς και το πως υλοποιήθηκαν. Ένα ακόμα σημαντικό κομμάτι είναι η ανάλυση της χρήσης των γραφικών και των ενισχύσεων που εφαρμόστηκαν με την χρήση των λειτουργιών του Unity.

Τέλος γίνονται προτάσεις για περαιτέρω βελτίωση, αναλύοντας τι μπορεί να επεκταθεί, όπως ο διαφορετικός τρόπος δημιουργίας χαρτών, προηγμένη νοημοσύνη για τους αντιπάλους ή και προσθήκη ιστορίας και διαλόγων για την δημιουργία ενός πραγματικά ολοκληρωμένου παιχνιδιού.

Κεφάλαιο 2ο: Ιστορία των Βιντεοπαιχνιδιών

2.1 Εισαγωγή

Η βιομηχανία των βιντεοπαιχνιδιών είναι αδιαμφισβήτητα ένας τομέας του οποίου η εξέλιξη είναι εκθετική μέσα στο χρόνο. Από τη γέννησή τους, τη δεκαετία του 1950 που ήταν απλά ένα έκθεμα σε μια επιστημονική έκθεση, με τη χρήση ενός καθοδικού σωλήνα και ενός παλμοσκοπίου, έχει εξελιχθεί στην πιο κερδοφόρα βιομηχανία διασκέδασης (Σχήμα 2.1). Ξεκινώντας από το πολύ γνωστό Pong το 1972 και καταλήγοντας στην δημιουργία παιχνιδιών αρκετά ρεαλιστικών ώστε να μην αναγνωρίζει κάποιος αν αυτό που βλέπει είναι παιχνίδι ή πραγματικότητα. Άρχισε την δεκαετία του 1950 σαν ιδέα αλλά δεν έγινε γνωστό για διάφορους λόγους- κυρίως θέματα μάρκετινγκ και το γεγονός ότι η ενασχόληση με βιντεοπαιχνίδια ήταν μια ξένη έννοια για εκείνη την εποχή- μέχρι τις αρχές της δεκαετίας του 1970 που δημιουργήθηκε το Pong από την Magnavox και ακολούθησε η Atari που δημιούργησε το πρώτο Pong Arcade (δηλαδή την δημιουργία μηχανήματος εξ ολοκλήρου φτιαγμένο για να παίζει κάποιος ένα παιχνίδι), που ήταν και η αφορμή για την άνοδο της βιομηχανίας.



Σχήμα 1: Richter F 2020, Gaming: The Most Lucrative Entertainment Industry By Far.

<<https://www.statista.com/chart/22392/global-revenue-of-selected-entertainment-industry-sectors/>>

2.2 Ιστορική Αναδρομή

2.2.1 Δεκαετία του 1970

Η αρχή της βιομηχανίας - ή τουλάχιστον της ανοδικής πορείας της - εντοπίζεται το 1972 με την κυκλοφορία του παιχνιδιού Pong (Σχήμα 2) από την Atari. Εμπνευσμένο από το αντίστοιχο παιχνίδι που είχε δημιουργηθεί για την πρώτη οικιακή κονσόλα Magnavox Odyssey, το Pong.

Ήταν η πρώτη εμπορική επιτυχία για την βιομηχανία [1], ανοίγοντας τον δρόμο για την εγκαθίδρυση του τομέα. Η επιτυχία του ήταν τόσο διαδεδομένη που υπήρχε κόσμος που πήγαινε σε μαγαζιά, μπαρ και εμπορικά κέντρα, μόνο για να παίξει το παιχνίδι αυτό. Λόγω της επιτυχίας αυτής, η αγορά γέμισε με κλώνους του Pong, και ως αποτέλεσμα το 1977 οι δημιουργοί των κλώνων αυτών πούλησαν τις παλαιότερες συσκευές τους σε μικρότερη τιμή για να αδειάσουν το απόθεμα τους [2]. Αυτό προκάλεσε τον κορεσμό στην αγορά μειώνοντας τα κέρδη της βιομηχανίας αφήνοντας στην αγορά μόνο την Atari και την Magnavox. Η επιτυχία του Space Invaders, παιχνίδι της Taito, το 1978 ήταν αυτό που έθεσε τον δρόμο για την χρυσή εποχή των βιντεοπαιχνιδιών. Το Atari VCS (Video Computer System), ύστερα γνωστό ως Atari 2600, πήρε τα δικαιώματα για το Space Invaders και έγινε έτσι η επόμενη μεγάλη επιτυχία σε βαθμό που τετραπλασίασε τα κέρδη της κονσόλας. Αυτό βοήθησε την Atari να ανακάμψει από τις προηγούμενες απώλειες και εν συνεχεία το Atari 2600 αναβίωσε την βιομηχανία των κόνσολών για το σπίτι.



Σχήμα 2: Pong Arcade Machine

<https://artsandculture.google.com/asset/pong-arcade-game-atari/5QGMET_i-9Qlww>

2.2.2 Δεκαετία του 1980

Η δεκαετία του 1980 θεωρείται η χρυσή εποχή των βιντεοπαιχνιδιών καθώς και το ζενίθ της αγοράς. Οι πωλήσεις των Arcade είχαν αυξηθεί από 50 εκατομμύρια το 1978, σε 900 εκατομμύρια το 1982 με την μεγαλύτερη επιτυχία να είναι το Pac-Man της Namco που οι πωλήσεις του έφτασαν πάνω από 350 χιλιάδες μηχανήματα καθώς και την εισφορά 1 δισεκατομμυρίου σε νομίσματα. Το αποκορύφωμα της αγοράς ήταν το 1982 όπου τα έσοδα της βιομηχανίας έφτασαν τα 8 δισεκατομμύρια σε νομίσματα (αντίστοιχο με 21,6 δισεκατομμύρια το 2020). Είχε μεγαλύτερη επιτυχία από την βιομηχανία του Hollywood και της Pop μουσικής μαζί (3 και 4 δισεκατομμύρια αντίστοιχα).

Το 1983 ήταν η χρονιά που έγινε η δεύτερη μεγάλη κατάρρευση της αγοράς παιχνιδομηχανών, μεγαλύτερη από αυτήν του 1977, καθώς υπήρχε ένας υπερκορεσμός βιντεοπαιχνιδιών αμφιβόλου ποιότητας επειδή η παραγωγή τους ήταν απλή. Το γεγονός αυτό σε συνδυασμό με την άνοδο των προσωπικών υπολογιστών είχε ως αποτέλεσμα την έλλειψη τόσο εσόδων όσο και της εμπιστοσύνης από τους καταναλωτές οι οποίοι θεώρησαν ότι η προσοχή τους θα έπρεπε να είναι στραμμένη στους προσωπικούς υπολογιστές και ότι οι κονσόλες ήταν ακόμα μια μόδα. Αυτό άλλαξε με την κυκλοφορία της κονσόλας Famicom (ευρέως γνωστή ως Nintendo Entertainment System, NES), που αναβίωσε την βιομηχανία των κόνσολων το 1985. Η κονσόλα αυτή έφερε μεγάλη ανάπτυξη καθώς πολλά από τα παιχνίδια που έχουν δημιουργηθεί για αυτήν έχουν δημιουργήσει παγκοσμίως γνωστές και επιτυχημένες σειρές παιχνιδιών έως και σήμερα (Final Fantasy, Dragon Quest, Legend of Zelda). Μέχρι το 1989 τα έσοδα των παιχνιδιών κασέτας είχαν αυξηθεί στα 2 δισεκατομμύρια σε αντίθεση με τα παιχνίδια δίσκου φτιαγμένα για προσωπικούς υπολογιστές που είχαν λιγότερο από 300 εκατομμύρια σε έσοδα.

Στα τέλη της δεκαετίας άρχισε και η κυκλοφορία των MUD (Multi User Dungeon) παιχνιδιών που ήταν συνήθως παιχνίδια βασισμένα σε κείμενο που εισήγαγαν για πρώτη φορά την έννοια του Multiplayer Online Game επιτρέποντας σε κόσμο να παίζει μεταξύ τους με την χρήση telnet clients ή ειδικά φτιαγμένων MUD clients που ήταν φτιαγμένα με σκοπό την βελτίωση της εμπειρίας για τον χρήστη.



Σχήμα 3: Nintendo Entertainment System, NES

https://en.wikipedia.org/wiki/Nintendo_Entertainment_System

2.2.3 Δεκαετία του 1990

Η δεκαετία του 1990 σηματοδότησε αξιοσημείωτες καινοτομίες στον τομέα των βιντεοπαιχνιδιών. Η στροφή από γραφικά τύπου ράστερ σε 3D ήταν μια από αυτές και έδωσε πάτημα για την δημιουργία περισσότερων ειδών παιχνιδιών όπως τα FPS (First Person Shooter), RTS (Real Time Strategy) καθώς και λόγω της εισόδου του ίντερνετ στα σπίτια ευρέως άρχισαν να έχουν άνοδο και τα MMOG (Massive Multiplayer Online games), η εξέλιξη των MUD, με την κυκλοφορία παιχνιδιών όπως Underlight, EverQuest, Meridian 59.

Λόγω της κυκλοφορίας του GameBoy το 1989 από την Nintendo, τα παιχνίδια χειρός είδαν μεγάλη άνοδο την δεκαετία του 90 επιτρέποντας στον χρήστη να παίζει όπου αυτός θέλει. Παρόλο που παρέμειναν διάσημα, τα Arcade είχαν μια πτώση αυτή τη δεκαετία καθώς ο κόσμος στρέφεται όλο και περισσότερο στις κονσόλες και τους ηλεκτρονικούς υπολογιστές. Η άνοδος αυτή προκάλεσε τον

τομέα των Βιντεοπαιχνιδιών να γίνει δημοφιλής και επιτυχημένος τομέας παγκοσμίως αποβάλλοντας την έννοια της μόδας από τον ορισμό τους. Αυτό οδήγησε στην δημιουργία ακόμα περισσότερων εταιρειών με μεγαλύτερες ομάδες παραγωγής και συνεργασία με τους τομείς τόσο του κινηματογράφου όσο και της μουσικής. Μια ακόμα σημαντική δημιουργία ήταν η γέννηση των Sim (Simulation Games) που προσομοιώνουν πτυχές του πραγματικού κόσμου ξεκινώντας με την Maxis που δημιούργησε αρχικά το SimCity το 1989 και αποκορυφώθηκε με το The Sims που κυκλοφόρησε το 2000 που είναι εάν από τα παιχνίδια με τις καλύτερες πωλήσεις μέχρι και σήμερα.

Τη δεκαετία του 90 έγινε και η στροφή από κασέτες σε CD, κυρίως λόγω της μειωμένης πολυπλοκότητας, κόστους παραγωγής και ευκολίας καθώς οι κασέτες είχαν μέσα εάν ολόκληρο ηλεκτρικό κύκλωμα σε αντίθεση με τα CD που ήταν απλά ένα μέσο αποθήκευσης. Αυτό επέτρεψε την δημιουργία σημαντικά μεγαλύτερων και πιο περίπλοκων παιχνιδιών αφού το μεγαλύτερο μέγεθος που μπορούσε να υποστηρίξει μια κασέτα για την κονσόλα Nintendo 64 ήταν 64mb σε αντίθεση με τα CD που από την αρχή τους υποστηρίζουν 650mb, 10 φορές δηλαδή μεγαλύτερη χωρητικότητα. Παρόλη τη μεγάλη διαφορά στην χωρητικότητα όμως τα παιχνίδια και των δυο κόνσολων ήταν εξίσου ποιοτικά λόγω τεχνικών που χρησιμοποιούσαν για να μειώσουν τους τεχνικούς περιορισμούς της κάθε κονσόλας αντίστοιχα. Στα τέλη της δεκαετίας οι κυρίαρχες κονσόλες ήταν το Playstation 1 και το Nintendo 64.

2.2.4 Δεκαετία του 2000

Με τις αρχές της νέας χιλιετίας έρχεται και η μοντέρνα εποχή των βιντεοπαιχνιδιών. Λόγω των ραγδαίων τεχνολογικών αλλαγών και την εισαγωγή του διαδικτύου στον περισσότερο κόσμο, έρχεται μια νέα γενιά κόνσολων που αμέσως κάνει την προηγούμενη να φαίνεται απαρχαιωμένη. Η Sony με την κυκλοφορία του Playstation 2 χρησιμοποιεί DVD για τα παιχνίδια εκτοξεύοντας την χωρητικότητα από 650MB σε 4.7 GB και βάζοντας λειτουργίες αναπαραγωγής ταινίας και μουσικής κάνοντας το Playstation 2 εάν ολοκληρωμένο σύστημα ψυχαγωγίας. Σε αυτήν την έκτη γενιά κόνσολων υπάρχει επίσης το Microsoft Xbox, το Sega Dreamcast και το Nintendo Gamecube. Η στροφή στο 3D πλέον αρχίζει να είναι μονόδρομος και εισάγει σε πολλά είδη παιχνιδιών έναν νεοσύστατο ρεαλισμό.



Σχήμα 4: Σύγκριση μεταξύ Final Fantasy VII(1998, PS1) και Final Fantasy X(2001,PS2)

Λόγω της προόδου της τεχνολογίας, και πιο συγκεκριμένα, την βελτίωση των επεξεργαστών και της αύξησης δυνατοτήτων του διαδικτύου, άρχισαν να ανθίζουν τα online καταστήματα στις κονσόλες όπως το Playstation Network το Microsoft Store και το Wii Shop Channel. Με την κυκλοφορία του

Playstation 3 του Xbox 360 και του Wii το 2006 έχουν αρχίσει να παράγονται παιχνίδια υψηλής ευκρίνειας (HD και Full-HD), επιπροσθέτως, ενσωματώνονται στις ίδιες τις κονσόλες η δυνατότητα σύνδεσης στο διαδίκτυο καθώς και τα αντίστοιχα καταστήματα τους, αλλάζοντας έτσι τον τρόπο με τον οποίο επικοινωνούν οι παίκτες καθώς και τον τρόπο με τον οποίο λαμβάνουν παιχνίδια και αναβαθμίσεις για τις κονσόλες τους.

Προς τα τέλη της δεκαετίας είδε μεγάλη αύξηση και ο τομέας του mobile gaming λόγω της κυκλοφορίας αντίστοιχων online καταστημάτων όπως αυτά των κόνσολών, στα κινητά. Το 2008 με το τότε Android Market και πλέον Google Play Store καθώς και το App Store της Apple άρχισε η δημιουργία και διανομή παιχνιδιών σε κινητά χωρίς ανάγκη για εξωτερικό παράγοντα με το πιο διάσημο τότε να είναι το Angry Birds που κυκλοφόρησε το 2009.

2.2.5 Δεκαετία του 2010

Υπάρχουν διάφοροι λόγοι οι οποίοι καθιστούν την δεκαετία του 2010 αυτήν που έφερε το gaming στο προσκήνιο. Οι προσωπικοί υπολογιστές και οι κονσόλες, πλέον ικανές να προβάλλουν απaráμιλλο ρεαλισμό και βύθιση στα παιχνίδια τους αρχίζουν να προσελκύουν πλέον έναν τεράστιο αριθμό κοινού. Με την κυκλοφορία του Playstation 4 και Xbox One το 2013 και έχοντας πλέον το Full-HD ως μονόδρομο οι κονσόλες γίνονται ο πιο γρήγορος και ταυτόχρονα προσιτός τρόπος για να μπει κάποιος στον κόσμο των παιχνιδιών. Τα είδη των παιχνιδιών, αφού έχει εξαλειφθεί το μεγαλύτερο ποσοστό τεχνολογικών περιορισμών, έχουν αυξηθεί και ταυτόχρονα εξελιχθεί για προσφέρουν μια πολύ πιο ολοκληρωμένη εμπειρία ανοίγοντας τον δρόμο για την δημιουργία παιχνιδιών.

Στα Μέσα της δεκαετίας του 2010 άρχισε να έχει ανάπτυξη και η εικονική πραγματικότητα εφόσον φεύγουν οι τεχνολογικοί περιορισμοί. Έκανε την αρχή του με την κυκλοφορία του Oculus Rift για τον υπολογιστή προσφέροντας ασύγκριτη ποιότητα σε σχέση με οποιαδήποτε άλλη προσπάθεια. Από τότε έχουν κυκλοφορήσει και άλλα μοντέλα όπως το HTC Vive ή το PSVR, το καθένα με τα θετικά και αρνητικά που τα διαφοροποιούν αναμεταξύ τους.

Ένας ακόμα τομέας που είδε τεράστια αύξηση σε αυτήν την δεκαετία είναι το mobile gaming. Με τα κινητά πλέον να έχουν πολύ δυνατά χαρακτηριστικά αλλά και με εξελίξεις στο software τόσο των κινητών όσο και των εργαλείων ανάπτυξης εφαρμογών για κινητά, τα παιχνίδια στο κινητό έχουν εξελιχθεί και συμπεριλαμβάνουν σχεδόν όλα τα είδη παιχνιδιών που υπάρχουν στους υπολογιστές και τις κονσόλες. Ένας από τους λόγους που έχουν γίνει τόσο διάσημα είναι και το ότι ένα μεγάλο ποσοστό αυτών είναι δωρεάν δίνοντας την επιλογή στον οποιονδήποτε να μπορεί να τα παίξει.

Ένας από αυτούς είναι το twitch, μια πλατφόρμα που επιτρέπει σε οποιονδήποτε να κάνει ζωντανή μετάδοση και δίνει στους χρήστες του την επιλογή να παρακολουθήσουν οποιονδήποτε επιθυμούν. Αυτό σε συνδυασμό με την κυκλοφορία του DOTA 2 (Defence Of The Ancients) και του League of Legends, παιχνίδια τύπου M.O.B.A. (Multiplayer Online Battle Arena), έθεσε τα θεμέλια για την εδραίωση της σκηνής των e-sport. Τα e-sports όπως φαίνεται και από το όνομα του είναι ηλεκτρονικά αθλήματα στα οποία είτε μεμονωμένα άτομα αλλά και ομάδες παίζουν ανταγωνιστικά σε πρωταθλήματα με το έπαθλο να είναι συνήθως ένα χρηματικό ποσό.

2.3 Τύποι παιχνιδιών

Οι τύποι των παιχνιδιών που υπάρχουν είναι πολυάριθμοι, αλλά οι μεγαλύτερες κατηγορίες που μπορούν να διακριθούν είναι οι εξής [3]:

2.3.1 Παιχνίδια Δράσης

Τα παιχνίδια δράσης έχουν το χαρακτηριστικό ότι ο χαρακτήρας έχει κάποιες φυσικές προκλήσεις και δίνει έμφαση στον συντονισμό χεριού-ματιού καθώς και στον χρόνο αντίδρασης. Είναι αρκετά ευκολά για να ξεκινήσει κάποιος να παίζει αμέσως και για αυτό συνήθως είναι και τα πιο διάσημα. Ορισμένες μεγάλες υποκατηγορίες είναι:

- **Πλατφόρμας**

Τα παιχνίδια πλατφόρμας παίρνουν το όνομα τους από την διαμόρφωση της πίστας η οποία συνήθως περιλαμβάνει πλατφόρμες στις οποίες ο χαρακτήρας τρέχει πηδάει η πέφτει. Κάποια από τα πιο διάσημα παιχνίδια πλατφόρμας είναι η σειρά παιχνιδιών Mario

- **Shooter**

Τα Shooter παιχνίδια επιτρέπουν στον χαρακτήρα να χρησιμοποιεί όπλα για να αντιμετωπίσει αντιπάλους ή άλλους παίκτες. Χωρίζονται σε Shooter πρώτου προσώπου, τρίτου προσώπου και από πάνω προς τα κάτω ανάλογα με το ποια οπτική του χαρακτήρα έχει ο παίκτης. Διάσημα παιχνίδια περιλαμβάνουν Call of Duty, Fortnite

- **Fighting**

Τα Fighting παιχνίδια δίνουν έμφαση στην μάχη και στις περισσότερες φορές την άοπλη μάχη μεταξύ 2 χαρακτήρων. Ως επί το πλείστον περιλαμβάνουν μια γκάμα από συγκεκριμένους χαρακτήρες με συγκεκριμένες κινήσεις ο καθένας όπου έρχονται αντιμέτωποι ο ένας με τον άλλον. Τα πιο γνωστά στην κατηγορία αυτή είναι Street Fighter, Mortal Kombat, Tekken.

- **Επιβίωσης**

Τα παιχνίδια επιβίωσης συνήθως έχουν έναν σχετικά ανοιχτό κόσμο στον οποίο ο παίκτης πρέπει να αντέξει όσο περισσότερο γίνεται από εξωτερικούς παράγοντες ή αντιπάλους δίνοντας του διαφορά αντικείμενα μέσα στον χάρτη που μπορούν να τον βοηθήσουν να φτιάξει οτιδήποτε απαραίτητο για την επιβίωση του.

- **Beat-em Up**

Τα Beat-em up παιχνίδια μοιάζουν με τα Fighting αλλά η διαφορά τους είναι ότι ο παίκτης έχει τον έλεγχο ενός χαρακτήρα και παλεύει ορδές από αντιπάλους αντί για έναν συγκεκριμένο την φορά. Διάσημα παιχνίδια περιλαμβάνουν God of War, Bayonetta, Double Dragon.

- **Rhythm**

Τα παιχνίδια ρυθμού ξεφεύγουν από την κλασική έννοια της δράσης και δίνουν περισσότερο έμφαση στα ανταντακλαστικά του παίκτη βάζοντας τον να πατήσει συγκεκριμένα κουμπιά σε συγκεκριμένη χρονική στιγμή ακολουθώντας τον ρυθμό ενός τραγουδιού. Παραδείγματα περιλαμβάνουν Guitar Hero, Dance Dance Revolution, Osu.

2.3.2 Παιχνίδια Δράσης-Περιπέτειας

Τα παιχνίδια Δράσης-Περιπέτειας συνήθως συνδυάζουν συστατικά στοιχεία και από τους δυο τύπους, πιο συγκεκριμένα έχουν μια μεγάλη αποστολή που πρέπει να ολοκληρωθεί. Δίνουν συνήθως περισσότερη έμφαση στην εξερεύνηση του χάρτη, χρησιμοποιώντας για παράδειγμα αντικείμενα που βρίσκονται στον χάρτη που είναι απαραίτητα για την πρόοδο, όπως κλειδιά όπλα κλπ. Οι υποκατηγορίες του είναι οι εξής:

- **Metroidvania**

Τα Metroidvania παιχνίδια έχουν ως πηγή έμπνευσης τα παιχνίδια Metroid και Castlevania. Αυτός ο τύπος παιχνιδιού μοιάζει με τα παιχνίδια πλατφόρμας και τα γενικότερα παιχνίδια δράσης περιπέτειας με τις διαφορές να είναι ότι ενώ συνήθως είναι αρκετά γραμμικά τα επίπεδα τους, στα Metroidvania είναι απαραίτητη πολλές φορές η υπαναχώρηση σε προηγούμενα σημεία αφού έχουν ολοκληρωθεί κάποιες προϋποθέσεις για την πρόοδο στο παιχνίδι πχ κάποιο κλειδί η κάποιο απαραίτητο αντικείμενο. Επίσης διαφοροποιούνται από τα υπόλοιπα καθώς δίνουν περισσότερη έμφαση και στην μάχη. Παραδείγματα άλλων Metroidvania παιχνιδιών περιλαμβάνουν τα, Hollow Knight, Ori and the Blind Forest, Dead Cells.

- **Επιβίωσης Τρόμου.**

Τα παιχνίδια επιβίωσης τρόμου έχουν συνήθως σύνθεση που είναι πιο κατάλληλη για ενήλικους δείχνοντας στον παίκτη πολλές φορές σκηνές βίας και τρόμου και έχουν ως σκοπό την επιβίωση του χαρακτήρα με περιορισμένους πόρους που βρίσκουν στον χάρτη. Παραδείγματα παιχνιδιών περιλαμβάνουν τα Resident Evil, Silent Hill, Outlast, Amnesia

2.3.3 Παιχνίδια Περιπέτειας

Τα παιχνίδια περιπέτειας ορίζονται ως παιχνίδια τα οποία δεν έχουν στοιχεία δράσης μέσα τους και σκοπός είναι η αλληλεπίδραση με το περιβάλλον και τους χαρακτήρες ή η επίλυση ορισμένων παζλ για την πρόοδο της ιστορίας. Είναι τύπος παιχνιδιού που δεν έχει εξελιχθεί πολύ από τότε που δημιουργήθηκε που το κάνει και λιγότερο διάσημο σαν είδος.

Ορισμένες υποκατηγορίες είναι οι εξής:

- **Περιπέτειες κειμένου**

Οι περιπέτειες κειμένου όπως δείχνει και το όνομα τους βασίζονται απλά στην ανάγνωση κειμένου και την χρήση πληκτρολογίου για την εγγραφή εντολών όπως για παράδειγμα “Πήγαινε Βόρεια”, “Άνοιξε την πόρτα”. Μεγάλος χρόνος στην παραγωγή τους αφιερώνεται στις διάφορες απαντήσεις που μπορεί να δώσει ο παίκτης. Παραδείγματα παιχνιδιών είναι Zork, The Dreamhold.

- **Γραφικές Περιπέτειες**

Με την εξέλιξη των υπολογιστών οι περιπέτειες κειμένου εξελίχθηκαν και εισήγαγαν εικόνες κυρίως με τις οποίες ο χρήστης αλληλοεπιδρά με την χρήση του ποντικιού. Παραδείγματα παιχνιδιών περιλαμβάνουν τα Broken Sword, Agatha Christie: And then there were none.

- **Τρισδιάστατα Πραγματικού Χρόνου**

Η τελευταία εξέλιξη στο είδος αυτό ξεφεύγει από την χρήση προκαθορισμένων εικόνων ως γραφικά και έχει τον χαρακτήρα να εξερευνά έναν ανοιχτό χάρτη και στοιχεία μέσα σε αυτόν για την πρόοδο της ιστορίας. Κατά κόρων οι επιλογές που παίρνει ο παίκτης στο παιχνίδι επηρεάζουν δραστικά την εξέλιξη και την έκβαση της ιστορίας. Μερικά παραδείγματα τίτλων είναι Detroit Become Human, Heavy Rain,

2.3.4 Παιχνίδια Ρόλων

Τα παιχνίδια Ρόλων (RPG) είναι ένα από τα μεγαλύτερα και πιο διάσημα είδη παιχνιδιών. Συνήθως διαδραματίζονται σε μεσαιωνική εποχή και έχουν έντονο το στοιχείο της φαντασίας. Σκοπός τους είναι συνήθως η εκπόνηση μιας ιστορίας καθώς και άλλων παράπλευρων αποστολών που βοηθούν

στην ανάπτυξη του χαρακτήρα, των ικανοτήτων ή ακόμα και της ιστορίας του παιχνιδιού. Συνήθως το σύστημα μάχης είναι τακτικό και γίνεται με εναλλαγή σειράς παίκτη και αντιπάλου. Σε πολλά παιχνίδια ρόλων επίσης, δίνεται η δυνατότητα να έχει ο παίκτης επίδραση στην ιστορία με τις επιλογές του, γεγονός που κάνει την εμπειρία πιο καθηλωτική. Γνωστές υποκατηγορίες του είδους είναι οι εξής:

- **RPG Δράσης**

Τα RPG δράσης έχουν το χαρακτηριστικό ότι το σύστημα μάχης αλλάζει από τακτική σε πραγματικού χρόνου, οπότε δίνει περισσότερη έμφαση στην ικανότητα και τα αντανakλαστικά του ιδίου του παίκτη πάρα στα στατιστικά του εκάστοτε χαρακτήρα.

- **MMORPG**

Τα MMORPG (Massively Multiplayer Online Role Playing Games) είναι παιχνίδια πολλαπλών παικτών με την χρήση του ιντερνέτ. Ο κάθε παίκτης έχει τον δικό του χαρακτήρα με αρκετές συνήθως επιλογές από εξωτερική εμφάνιση και ικανότητες, δίνοντας μια μοναδικότητα στον κάθε παίκτη ξεχωριστά. Είναι η εξέλιξη των Multi User Dungeon παιχνιδιών που είχαν κυκλοφορήσει την δεκαετία του 1970. Ο παίκτης μπορεί να ολοκληρώσει το περιεχόμενο του παιχνιδιού μόνος, με παρέα ή με αγνώστους και έχει ο καθένας την δικιά του ξεχωριστή πρόοδο στην ιστορία του παιχνιδιού.

- **Roguelikes**

Τα Roguelikes είναι ένα είδος του οποίου ο τίτλος είναι εμπνευσμένος από το παιχνίδι Rogue. Κύριο χαρακτηριστικό τους είναι η επανάληψη ενός επιπέδου που είναι τυχαίο κάθε φορά με σκοπό την αναβάθμιση των στατιστικών του χαρακτήρα προχωρώντας σταδιακά περισσότερο μέχρι την νίκη εναντίων του τελευταίου αντίπαλου.

- **Sandbox RPG**

Τα Sandbox RPG είναι παιχνίδια τα οποία έχουν ανοιχτό κόσμο και αφήνουν τον παίκτη να επιλέξει τι θα κάνει. Έχουν προφανώς μια κύρια αποστολή αλλά δεν είναι δεσμευμένοι σε αυτήν. Η ελευθερία αυτή βγάζει το στοιχείο της γραμμικότητας που έχουν τα περισσότερα παιχνίδια και αφήνει τον παίκτη να εξερευνήσει με τον τρόπο που θέλει αυτός και τους δημιουργούς του παιχνιδιού να φτιάχνουν παιχνίδια με μεγάλο επίπεδο ζωντάνιας. Παραδείγματα περιλαμβάνουν The Elder Scrolls: Skyrim, Grand Theft Auto, Borderlands.

2.3.5 Παιχνίδια Προσομοίωσης

Τα παιχνίδια προσομοίωσης έχουν έναν κοινό παράγοντα, την προσομοίωση της πραγματικής ή ιδεατής ζωής. Τα είδη παιχνιδιών προσομοίωσης που υπάρχουν είναι τα εξής:

- **Προσομοίωση Ζωής**

Τα παιχνίδια της κατηγορίας αυτής επιτρέπουν τον χειρισμό των επιλογών, χαρακτηριστικών μέχρι και αντιδράσεων του χαρακτήρα. Δίνουν έμφαση στην διαχείριση πόρων, σχέσεων, χρημάτων και είναι μια γενικότερη προσομοίωση της πραγματικής ζωής. Το πιο διάσημο και χαρακτηριστικό παιχνίδι στην κατηγορία αυτή είναι το The Sims

- **Προσομοίωση Οχημάτων.**

Οι προσομοιωτές οχημάτων περιλαμβάνουν όλα τα παιχνίδια τα οποία έχουν ως σκοπό τον χειρισμό ενός οχήματος βασισμένου στην πραγματικότητα όπως για παράδειγμα αμαξιού ή αεροπλάνου. Ο χειρισμός τους ξεφεύγει συνήθως από τον απλό έλεγχο με το πληκτρολόγιο και δίνει περισσότερη

έμφαση στην πολυπλοκότητα του κάθε οχήματος ξεχωριστά. Μπορεί να έχει μέσα και σύστημα μάχης στην περίπτωση που είναι προσομοιωτής πολεμικών αεροσκαφών. Παραδείγματα περιλαμβάνουν Microsoft Flight Simulator, Gran Turismo.

2.3.6 Παιχνίδια Στρατηγικής

Τα παιχνίδια στρατηγικής χρειάζονται περισσότερη σκέψη για το πως θα φέρουν τον παίκτη στην νίκη και όλο το παιχνίδι εξαρτάται από την σωστή χρήση και διαχείριση των πόρων του παίκτη (Μονάδες, πρώτες ύλες κλπ.). Στις αρχές παιζόταν με χρήση σειράς παίκτη αντιπάλου αλλά τα τελευταία χρόνια μετά από απαιτήσεις των παικτών έχουν στραφεί σε στρατηγική πραγματικού χρόνου. Μερικά από τα είδη παιχνιδιών στρατηγικής είναι τα εξής:

- **Στρατηγικής Πραγματικού Χρόνου**

Στα παιχνίδια πραγματικού χρόνου σκοπός είναι να συλλέξουν και να διατηρήσουν πόρους όπως βάσεις ενώ ταυτόχρονα προωθούν και αναπτύσσουν τόσο τους πόρους αυτούς όσο και των μονάδων μάχης τους. Διάσημα παραδείγματα περιλαμβάνουν: Age of Empires, Command and Conquer.

- **4X**

Σε αυτήν την υποκατηγορία κατατάσσονται τα παιχνίδια των οποίων οι στόχοι είναι οι εξής 4: Εξερεύνηση, Επέκταση, Εκμετάλλευση, Εξολόθρευση. Παιχνίδια αυτής της κατηγορίας συνήθως αφήνουν τον παίκτη σε μια εποχή στην οποία έχουν περιορισμένους πόρους και στρατεύματα και σκοπός τους είναι η εξερεύνηση του υπόλοιπου χάρτη με σκοπό να εκμεταλλευτούν τους πόρους που βρίσκονται σε αυτόν καθώς και τον προβιβάσμό σε επόμενη εποχή με συνεχώς εξελισσόμενη τεχνολογία και στρατεύματα.

- **Tower Defense Παιχνίδια**

Στα Tower Defense παιχνίδια, οι παίκτες πρέπει να αποκρούσουν ορδές από αντιπάλους που πλησιάζουν από προκαθορισμένα μονοπάτια για να καταστρέψουν την βάση, πλαισιώνοντας μέσα στον χάρτη χαρακτήρες ή κτίσματα που χτυπάνε τους αντιπάλους όταν περάσουν από την εμβέλεια τους. Οι αντίπαλοι έρχονται σε κύματα και στο τέλος κάθε κύματος ο παίκτης ανταμείβεται με χρυσό τον οποίο μπορεί να χρησιμοποιήσει για να αναβαθμιστεί ή να προσθέσει περισσότερους χαρακτήρες για την ερχόμενη επίθεση. Διάσημα παιχνίδια περιλαμβάνουν: Orcs must Die, Plants vs Zombies, Dungeon Defenders.

- **MOBA**

Τα MOBA (Multiplayer Online Battle Arena) παιχνίδια συνδυάζουν παιχνίδια δράσης, ρόλων και στρατηγικής μαζί. Ο παίκτης ελέγχει έναν χαρακτήρα σε έναν χάρτη μαζί με άλλους 9 παίκτες μέσω διαδικτύου. Χωρισμένοι σε 2 ομάδες έχουν ως σκοπό την καταστροφή της αντίπαλης βάσης. Ο παίκτης αναβαθμίζει τις ικανότητες και τα αντικείμενα που έχει ο χαρακτήρας του με χρυσό που αποκτά από την εξόντωση των αντίπαλων παικτών καθώς και τεράτων που βρίσκονται μέσα στον χάρτη ελεγχόμενα από τον υπολογιστή.

2.3.7 Παιχνίδια Αθλημάτων

Τα παιχνίδια Αθλημάτων είναι προσομοιωτές πραγματικών αθλημάτων όπως μπάσκετ, ποδοσφαίρου, γκολφ και μπιέζμπολ ή και αθλημάτων φτιαγμένων για το παιχνίδι συγκεκριμένα. Ο παίκτης μπορεί να είναι αντιμέτωπος με άλλους παίκτες ή με αντίπαλο τον υπολογιστή σε διάφορους βαθμούς

δυσκολίας. Τα παιχνίδια αυτά μπορεί να είναι ακριβή μίμηση των πραγματικών ομόλογων τους ή να διαφοροποιούν προσθέτοντας ή αφαιρώντας χαρακτηριστικά και κανόνες από το παιχνίδι. Διάσημα παραδείγματα περιλαμβάνουν: FIFA, NBA, Madden NFL, WWE.

Κεφάλαιο 3ο: Εργαλεία που Χρησιμοποιήθηκαν

3.1 Unity

Το Unity (Σχήμα 5) αναπτύχθηκε από την Unity Technologies και πρόκειται για μια μηχανή δημιουργίας παιχνιδιών για διαφορετικές πλατφόρμες. Η πρώτη ανακοίνωση και κυκλοφορία της έγινε τον Ιούνιο του 2005 στο Worldwide Developers Conference της Apple Inc. ως αποκλειστική μηχανή παιχνιδιών Mac OS X. Από το 2018, έχει επεκταθεί για να υποστηρίξει περισσότερες από 25 πλατφόρμες και έχει υιοθετηθεί από βιομηχανίες εκτός βιντεοπαιχνιδιών, όπως φιλμ, αυτοκίνητα, αρχιτεκτονική, μηχανική και κατασκευές.

Κατά τη διάρκεια των πρώτων δέκα ετών της κυκλοφορίας του προϊόντος, οι πληρωμένες εκδόσεις του Unity πωλήθηκαν εντελώς και το 2016, η εταιρεία άλλαξε την πολιτική της μετατρέποντας το σε μοντέλο συνδρομής.[1] Η Unity έχει δωρεάν και επί πληρωμή επιλογές αδειοδότησης. Η δωρεάν άδεια προορίζεται για προσωπική χρήση ή μικρότερες εταιρείες με λιγότερα από 100.000\$ ετησίως και οι συνδρομές βασίζονται σε έσοδα που δημιουργούνται από τα παιχνίδια που χρησιμοποιούν το Unity.



Σχήμα 5: Το λογότυπο του Unity

<https://commons.wikimedia.org/wiki/File:Unity_Technologies_logo.svg>

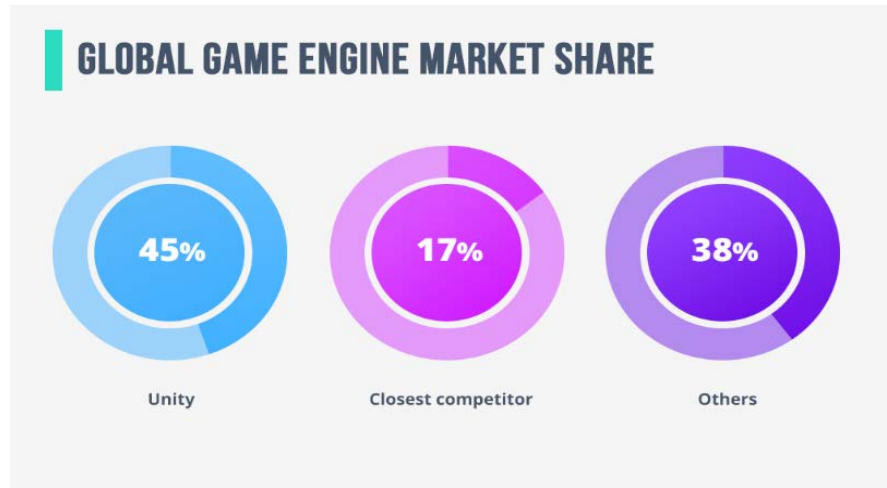
Το Unity δίνει στους χρήστες τη δυνατότητα για δημιουργία 3D, 2D, εικονικής πραγματικότητας και παιχνιδιών επαυξημένης πραγματικότητας, καθώς και προσομοιώσεων και άλλων εμπειριών. Επίσης, προσφέρει ένα πρωταρχικό API δέσμης ενεργειών σε C#, τόσο για τον επεξεργαστή Unity με τη μορφή προσθηκών, όσο και για τα ίδια τα παιχνίδια, καθώς και τη λειτουργία drag and drop. Παράλληλα, υποστηρίζει scripts και στις γλώσσες προγραμματισμού C++ και Java, ενώ παρέχει στο χρήστη ένα σημαντικό αριθμό έτοιμων materials που προσδιορίζουν πώς θα φαίνεται ένα 3D μοντέλο στην οθόνη, καθιστώντας το λογισμικό πρακτικά πιο εύχρηστο. Τέλος, ο ενδιαφερόμενος έχει τη δυνατότητα να προσθέσει δικούς του shaders σε Cg και GLSL για πλήρη έλεγχο του μετασχηματισμού και της παρουσίασης ή απεικόνισης αντίστοιχα. [2]

Ο επεξεργαστής Unity υποστηρίζεται σε Windows, macOS και στην πλατφόρμα Linux, ενώ η ίδια η μηχανή υποστηρίζει την κατασκευή παιχνιδιών για περισσότερες από 25 διαφορετικές πλατφόρμες, όπως φορητές συσκευές, επιτραπέζιους υπολογιστές, κονσόλες και εικονική πραγματικότητα. Στον κατάλογο με τις περιλαμβάνονται οι iOS, Android, Tizen, Windows, Universal Windows Platform, Mac, Linux, WebGL, PlayStation 4, PlayStation Vita, Xbox One, 3DS, Oculus Rift, Google Cardboard, Steam VR, PlayStation VR, Gear VR, Windows Mixed Reality, Daydream, Android TV,

Samsung Smart TV, tvOS, Nintendo Switch, Fire OS, Facebook Gameroom, Apple ARKit, ARCore της Google, Vuforia, και Magic Leap.

3.1.1 Πλεονεκτήματα του Unity

Το Unity αποτελεί μία μηχανή δημιουργίας παιχνιδιών με πολλά πλεονεκτήματα κι αυτός είναι ο κύριος λόγος που κατέχει ένα μεγάλο μερίδιο στην αγορά των μηχανών ανάπτυξης παιχνιδιών.



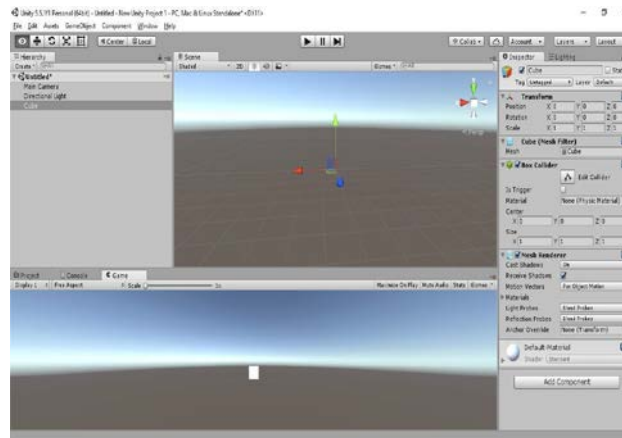
Σχήμα 6: Το Unity στην παγκόσμια αγορά μηχανών παιχνιδιών (2019) [4]

Παρακάτω παρατίθενται οι βασικοί λόγοι επιλογής της συγκεκριμένης πλατφόρμας [4]:

- Το περιβάλλον του είναι εύχρηστο προς το χρήστη και εξαιρετικά αποτελεσματικό κατά την απόδοση 2D και 3D σκηνών. Η ποιότητα που προσφέρεται είναι επίσης πολύ καλή σε σύγκριση με άλλες εφαρμογές.
- Το Unity είναι εξαιρετικό για cross-platform παιχνίδια. Η ανάπτυξη παιχνιδιών για διαφορετικές πλατφόρμες είναι πολύ δημοφιλής πλέον, ωστόσο η δημιουργία τους είναι πραγματικά χρονοβόρα και απαιτεί πολλές προσπάθειες για την ανάπτυξη εγγενών εφαρμογών. Μηχανές όπως το Unity έχουν κάνει τη διαδικασία αρκετά απλή και εύχρηστη σε σύγκριση με προηγούμενες εποχές.
- Η τεχνική υποστήριξη για το Unity είναι αποτελεσματική. Τα μέλη της τεχνικής υποστήριξης είναι πολύ εξειδικευμένα και οι χρήστες μπορούν να απευθύνονται εκεί σε περίπτωση τεχνικών ζητημάτων.
- Το Asset-Store που παρέχεται προσφέρει δωρεάν ορισμένα γραφικά μοντέλα-scripting code.
- Παρά το γεγονός ότι δεν είναι ανοιχτού κώδικα, το κόστος είναι πολύ μικρότερο σε σύγκριση με τους ανταγωνιστές του. Η αποδοτικότητα κόστους είναι ένα από τα κύρια πλεονεκτήματα της χρήσης του Unity.

3.1.2 Το περιβάλλον του Unity

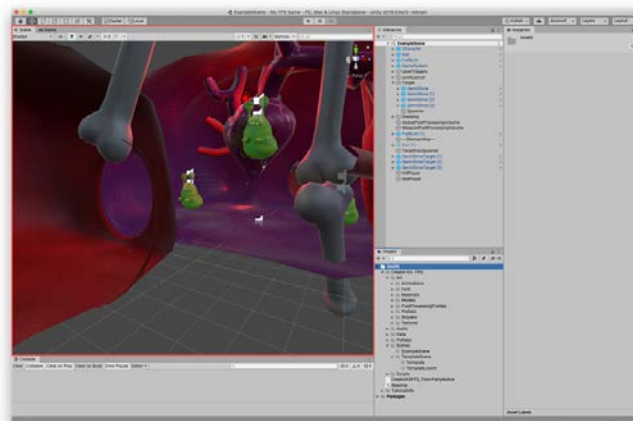
Όπως έχει αναφερθεί ήδη, το Unity προσφέρει στο χρήστη αρκετές δυνατότητες ώστε να καταστεί ευκολότερη η δημιουργία του παιχνιδιού. Στις παρακάτω ενότητες παρουσιάζονται τα βασικά εργαλεία που συναντάμε στο περιβάλλον του Unity(Σχήμα 7).



Σχήμα 7: Περιβάλλον του Unity

Scene View

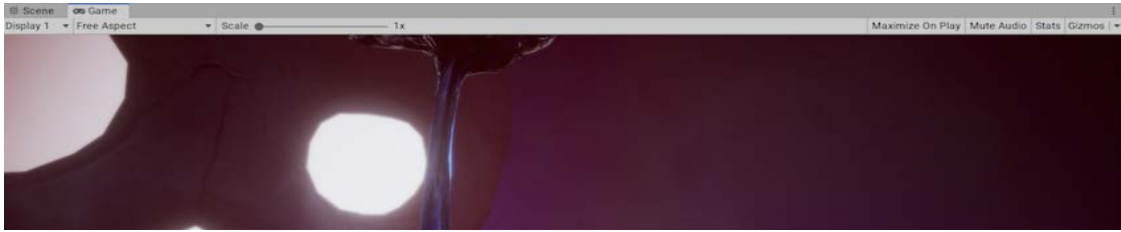
Το Scene View (Σχήμα 8) είναι η διαδραστική προβολή του χρήστη στον κόσμο που δημιουργεί. Μπορεί να χρησιμοποιείται για να επιλεγούν και να τοποθετηθούν τοπία, χαρακτήρες, κάμερες, φώτα και όλοι οι άλλοι τύποι GameObjects. Η επιλογή, ο χειρισμός και η τροποποίηση του GameObjects στο Scene View είναι μερικές από τις πρώτες δεξιότητες που πρέπει να μάθει όποιος επιθυμεί να εργαστεί στο Unity.



Σχήμα 8: Scene View

Game View

Η επιλογή Game View (Σχήμα 9) προβάλλει την εικόνα του παιχνιδιού από την Κάμερα (εξ) στην εφαρμογή. Αντιπροσωπεύει την τελική, εν δυνάμει, δημοσιεύσιμη έκδοση της εφαρμογής. Για να είναι δυνατός ο έλεγχος των όσων βλέπει ο παίκτης όταν χρησιμοποιεί όσα έχουν δημιουργηθεί πρέπει να χρησιμοποιούνται μία ή περισσότερες κάμερες.



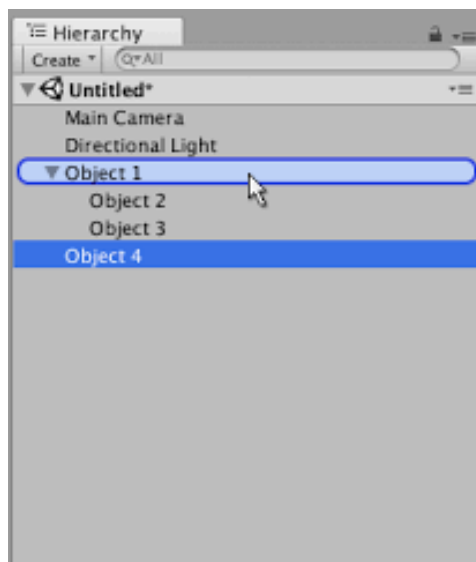
Σχήμα 9: Game View

Χρησιμοποιώντας τα κουμπιά στη Γραμμή εργαλείων είναι δυνατός ο έλεγχος της λειτουργίας «Επεξεργασία αναπαραγωγής» ώστε ο χρήστης να μπορεί να δει πώς παίζει η εφαρμογή που έχει αναπτύξει. Στη λειτουργία αυτή, οι όποιες αλλαγές είναι προσωρινές και επαναφέρονται με την έξοδο από αυτή. Το UI του προγράμματος επεξεργασίας σκουραίνει ως υπενθύμιση.

Hierarchy window

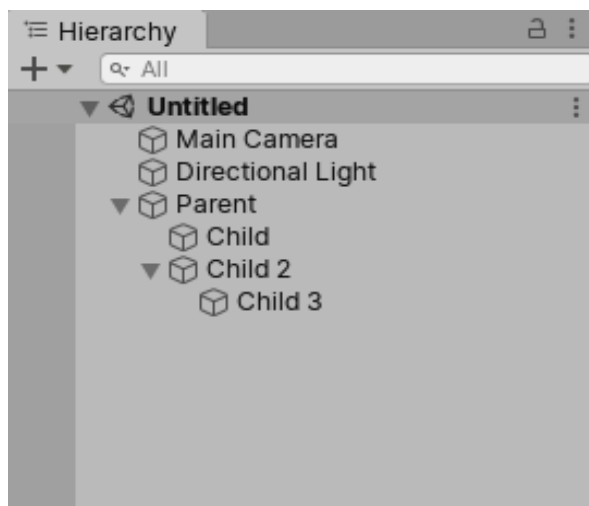
Το Hierarchy window περιέχει μια λίστα με κάθε GameObject στην τρέχουσα σκηνή. Ορισμένες από αυτές είναι άμεσες παρουσίες αρχείων Asset (όπως μοντέλα 3D) και άλλες είναι περιπτώσεις Prefabs, τα οποία είναι προσαρμοσμένα GameObjects που αποτελούν το μεγαλύτερο μέρος του παιχνιδιού. Με την προσθήκη ή κατάργηση GameObjects, αυτά εμφανίζονται και εξαφανίζονται επίσης από το Hierarchy window.

Από προεπιλογή, το Hierarchy window παραθέτει τα GameObjects κατά σειρά δημιουργίας, με τα πιο πρόσφατα δημιουργημένα GameObjects στο κάτω μέρος. Τα GameObjects μπορούν να αναδιαταχθούν σύροντάς τα προς τα πάνω ή προς τα κάτω ή κάνοντάς τα "παιδί" ή "γονέα" GameObjects.



Σχήμα 10: Προκαθορισμένο Παράθυρο Ιεραρχίας

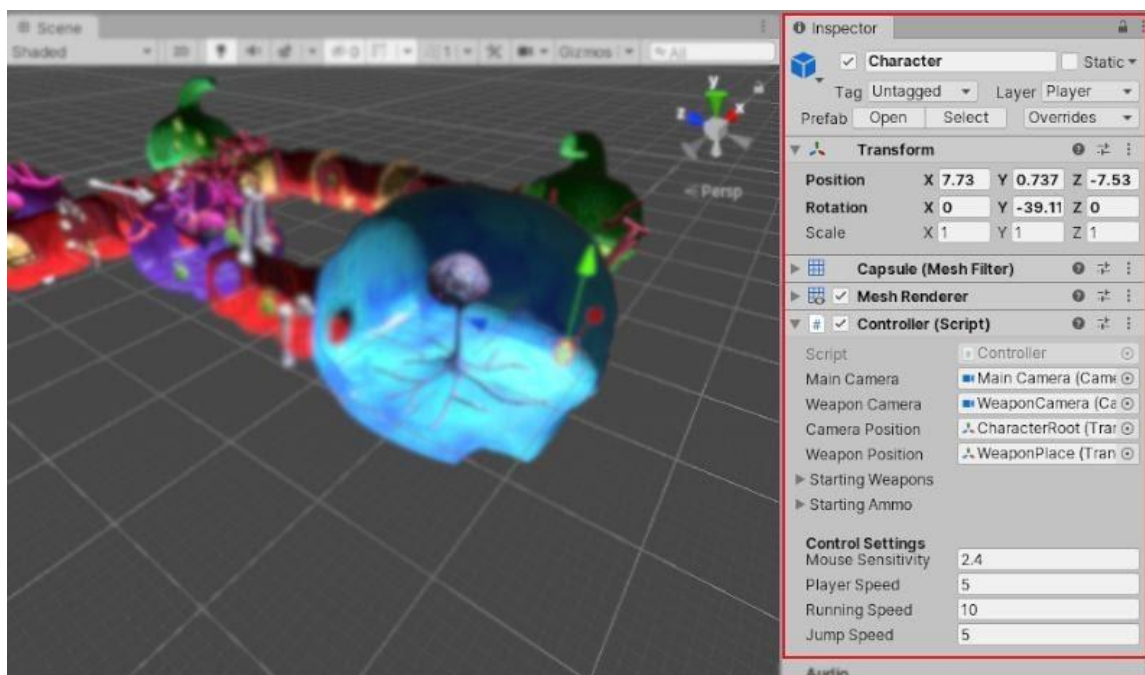
Στο Σχήμα 11, το Child και Child 2 είναι παιδιά του GameObjects Parent. Το Child 3 είναι ένα παιδί GameObject του Child 2 και ένας απόγονος του GameObject Parent.



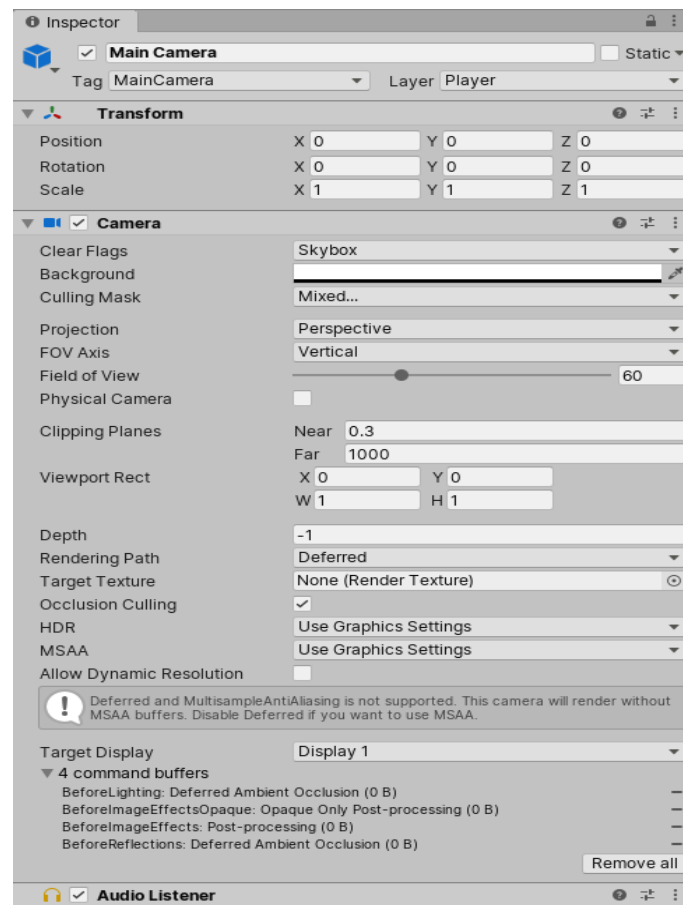
Σχήμα 11: Αναπαράσταση Parenting

Inspector window

Τα project στο πρόγραμμα επεξεργασίας του Unity αποτελούνται από πολλά GameObjects που περιέχουν σενάρια, ήχους και άλλα γραφικά στοιχεία. Το Inspector window (μερικές φορές αναφέρεται ως "The Inspector") εμφανίζει λεπτομερείς πληροφορίες σχετικά με το τρέχον επιλεγμένο GameObject, συμπεριλαμβανομένων όλων των συνημμένων στοιχείων και των ιδιοτήτων τους, και επιτρέπει την τροποποίηση της λειτουργικότητας του GameObjects στη σκηνή σας.



Σχήμα 12: Inspector window



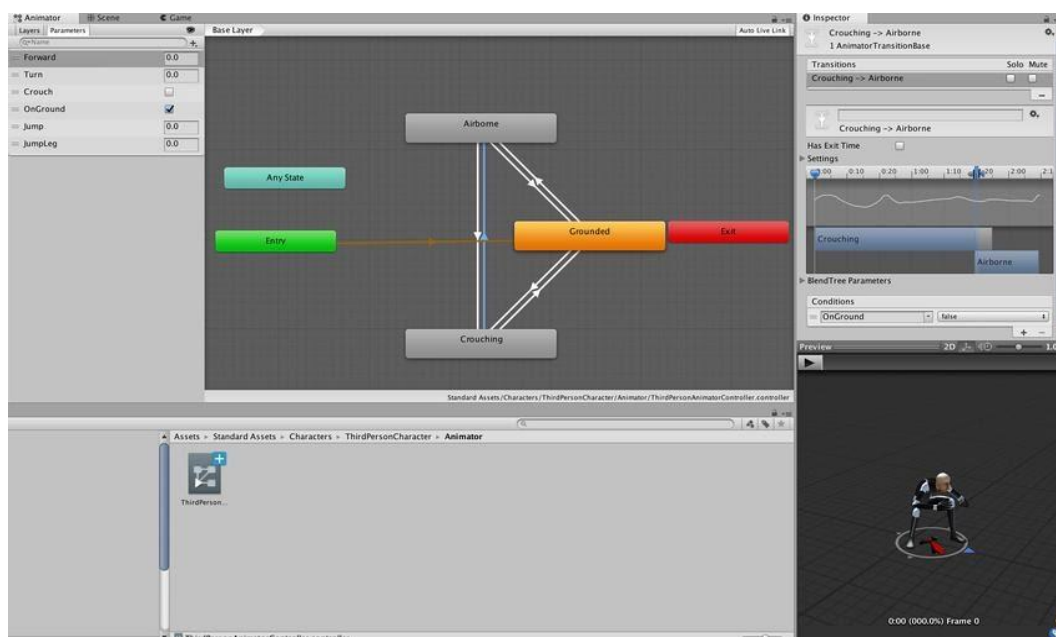
Σχήμα 13: Inspector window για ένα GameObject

Animator view

Οι Animation λειτουργίες της Unity περιλαμβάνουν επαναλαμβανόμενα κινούμενα σχέδια, πλήρη έλεγχο των animation weights κατά το χρόνο εκτέλεσης, κλήση συμβάντων από την αναπαραγωγή animation, εξελιγμένες ιεραρχίες και μεταβάσεις State Machine και πολλά άλλα.

Animations

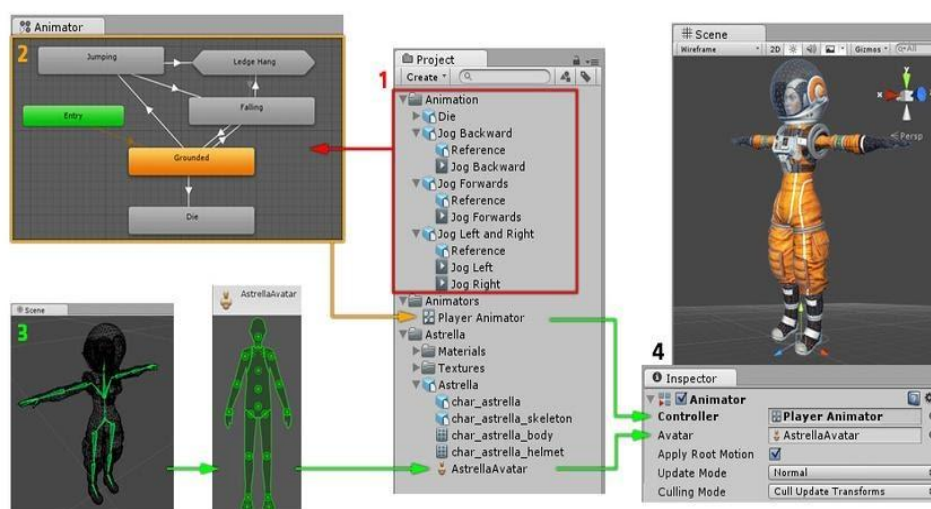
Το animation σύστημα της Unity βασίζεται στην έννοια των Animation Clips, τα οποία περιέχουν πληροφορίες σχετικά με το πώς ορισμένα αντικείμενα πρέπει να αλλάζουν τη θέση τους, την περιστροφή τους ή άλλες ιδιότητες με την πάροδο του χρόνου. Κάθε clip μπορεί να θεωρηθεί ως μία γραμμική εγγραφή και οργανώνονται σε ένα δομημένο σύστημα ροής που ονομάζεται Animator Controller. Ο Animator Controller ενεργεί ως "State Machine" που παρακολουθεί ποιο clip πρέπει να παίζει αυτήν τη στιγμή και τότε οι κινούμενες εικόνες πρέπει να αλλάζουν ή να συνδυάζονται μεταξύ τους. [5]



Σχήμα 14: Animator window

Ένας πολύ απλός ελεγκτής Animator μπορεί να περιέχει μόνο ένα ή δύο clips, για παράδειγμα για τον έλεγχο της περιστροφής και της αναπήδησης του powerup, ή για την κίνηση ενός ανοίγματος και κλεισίματος μιας πόρτας τη σωστή στιγμή. Ένας πιο προηγμένος ελεγκτής Animator μπορεί να περιέχει δεκάδες κινούμενες εικόνες ανθρωποειδών για όλες τις ενέργειες του κύριου χαρακτήρα και μπορεί να συνδυάζεται μεταξύ πολλαπλών clips ταυτόχρονα για να παρέχει μια ρευστή κίνηση καθώς η συσκευή αναπαραγωγής κινείται γύρω από τη σκηνή.

Το σύστημα Animation της Unity διαθέτει επίσης πολλές ειδικές δυνατότητες για το χειρισμό ανθρωποειδών χαρακτήρων που επιτρέπουν τον επαναπροσδιορισμό της ανθρωποειδούς κινούμενης εικόνας από οποιαδήποτε πηγή (για παράδειγμα: κίνηση λήψης, το Asset Store ή κάποια άλλη βιβλιοθήκη κινούμενων σχεδίων τρίτων) στο μοντέλο χαρακτήρων του χρήστη, καθώς και την προσαρμογή των ορισμών των μυών. Αυτές οι ειδικές λειτουργίες ενεργοποιούνται από το σύστημα Avatar της Unity, όπου οι ανθρωποειδείς χαρακτήρες αντιστοιχίζονται σε μια κοινή εσωτερική μορφή.



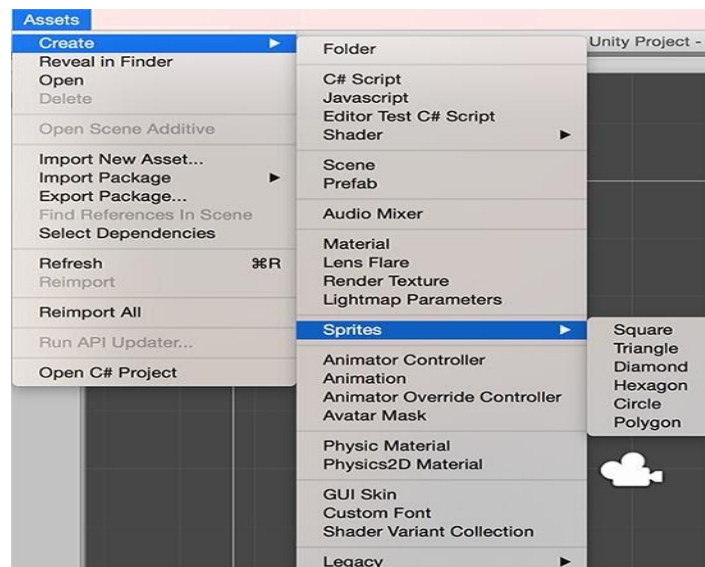
Σχήμα 15: Animation System

Κάθε ένα από αυτά τα κομμάτια (Animation Clips, Animator Controller και Avatar), συγκεντρώνονται σε ένα GameObject μέσω του Animator Component. Αυτό το στοιχείο έχει αναφορά σε έναν Animator Controller και (εάν απαιτείται) το Avatar για αυτό το μοντέλο. Ο Animator Controller, με τη σειρά του, περιέχει τις αναφορές στα Animation Clips που χρησιμοποιεί.

Sprite Editor 2D

Τα sprites είναι 2D γραφικά αντικείμενα. Για τη δημιουργία και διαχείρισή τους η Unity παρέχει ένα placeholder Sprite Creator, έναν ενσωματωμένο Sprite Editor, ένα Sprite Renderer και ένα Sprite Packer. [6]

- **Sprite Creator:** Επιτρέπει τη δημιουργία sprite placeholder στην εφαρμογή.
- **Sprite Editor:** Επιτρέπει την εξαγωγή sprite graphics από μια μεγαλύτερη εικόνα και την επεξεργασία ενός αριθμού εικόνων συνιστωσών σε μία texture στον επεξεργαστή εικόνων.
- **Sprite Renderer:** Τα Sprites αποδίδονται με ένα στοιχείο Sprite Renderer και όχι με το Mesh Renderer που χρησιμοποιείται με αντικείμενα 3D. Με αυτό ο χρήστης μπορεί να εμφανίσει εικόνες ως Sprites για χρήση σε σκηνές 2D και 3D.
- **Sprite Packer:** Χρησιμοποιείται για τη βελτιστοποίηση της χρήσης και της απόδοσης της μνήμης βίντεο στο project.



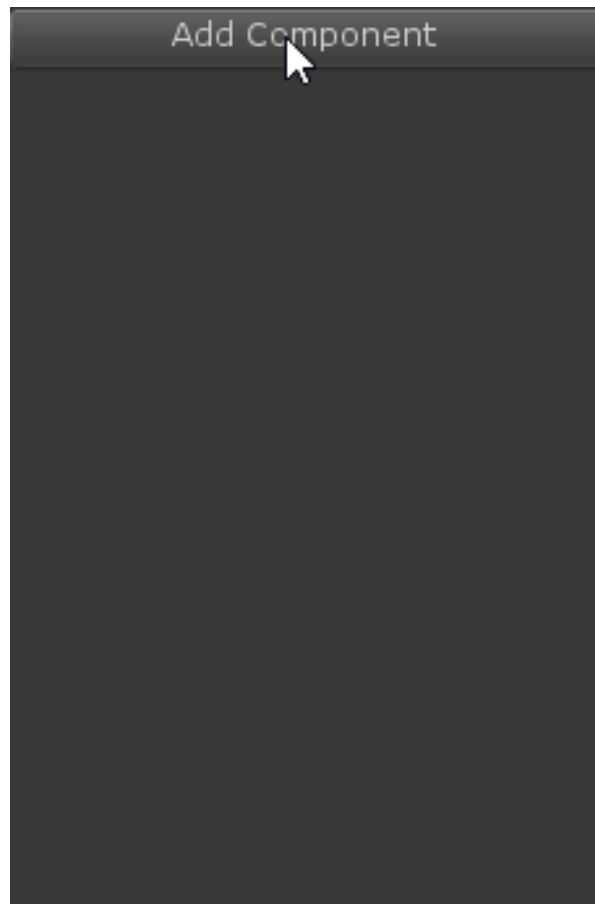
Σχήμα 16: Μενού δημιουργία Sprites

Για τη δημιουργία sprites επιλέγουμε Assets>Create>Sprites και, στη συνέχεια, το sprite placeholder που θέλουμε να κάνουμε (τετράγωνο, τρίγωνο, διαμάντι, εξαγωνο ή πολύγωνο). (Σχήμα 16) Το νέο sprite placeholder εμφανίζεται ως λευκό σχήμα στο φάκελο στοιχείων που έχουμε ανοίξει. Το όνομα του νέου sprite είναι από προεπιλογή στο όνομα του σχήματος, αλλά έχουμε την επιλογή να το μετονομάσουμε όταν δημιουργείται για πρώτη φορά.

Animation Rigging

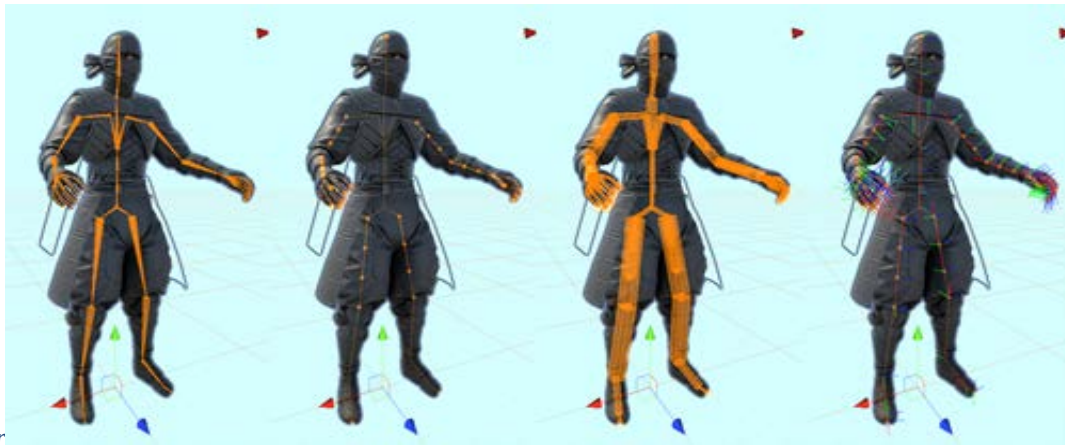
Το Animation Rigging επιτρέπει στο χρήστη να δημιουργεί και να οργανώνει διαφορετικά πακέτα χαρακτηριστικών που εφαρμόζονται στο project του με στόχο τη δημιουργία δια δραστικών sprites.

Το Rig Builder λειτουργεί παράλληλα με το Animator και δημιουργεί ένα γράφημα με δυνατότητα αναπαραγωγής που επισυνάπτεται στο υπάρχον Animator State Machine. Χρησιμοποιώντας το Rig Layers, το Rig Builder επιτρέπει τη στοίβαξη πολλαπλών Rigs, τα οποία μπορούν να ενεργοποιηθούν/απενεργοποιηθούν ανά πάσα στιγμή. [7]



Σχήμα 17: Rig Μενού

Μέσω του Bone Renderer επιτρέπεται στο χρήστη να προχωρήσει στο σχεδιασμό οστών για οπτικοποίηση και επιλογή κατά τη διάρκεια του rigging. Αυτά τα οστά δεν είναι ορατά στο Game view, η εμφάνισή τους μπορεί να προσαρμοστεί και το μέγεθος, το σχήμα και το χρώμα των οστών μπορούν να τροποποιηθούν. Επίσης, μπορούν να εμφανιστούν τρίποδα τοπικών αξόνων και το μέγεθός τους να προσαρμόζεται ώστε να ανταποκρίνεται στις προτιμήσεις των χρηστών. Ο χρήστης μπορεί να επιλέξει μία από τις προεπιλεγμένες εμφανίσεις: Pyramid, Line or Box.



Σχήμα 18: Bone Renderer

Animation Parameters

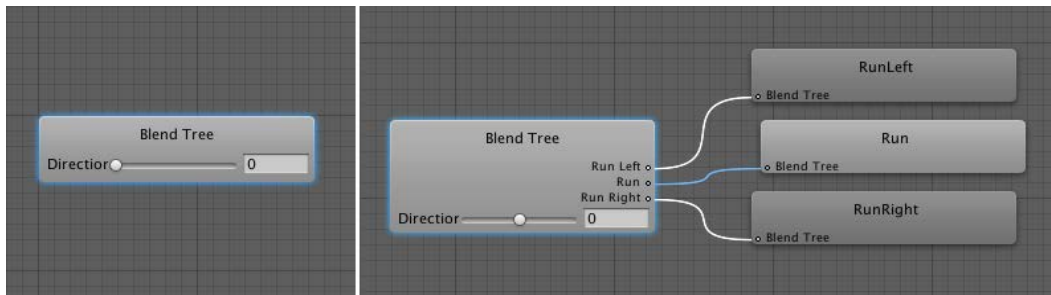
Οι Animation Parameters είναι μεταβλητές που ορίζονται σε έναν Animator Controller, στις οποίες ο χρήστης έχει πρόσβαση και μπορεί να εκχωρήσει τιμές από scripts. Για παράδειγμα, η τιμή μιας παραμέτρου μπορεί να ενημερωθεί με μια animation curve και στη συνέχεια να προσπελαστεί από ένα script έτσι ώστε, το βήμα ενός ηχητικού εφέ να μπορεί να μεταβάλλεται σαν να ήταν ένα κομμάτι animation. Μπορεί να πάρει έναν από τους τέσσερις βασικούς τύπους παραμέτρων: [8]

- **Integer:** ένας ακέραιος αριθμός
- **Float :** ένας αριθμός με κλασματικό μέρος
- **Bool:** πραγματική ή ψευδής τιμή (αντιπροσωπεύεται από ένα πλαίσιο ελέγχου)
- **Trigger:** μια boolean παράμετρος που επαναφέρεται από τον controller όταν καταναλώνεται από μια μετάβαση.

Blend Trees

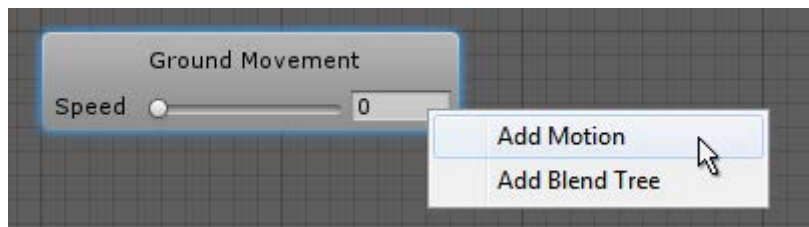
Ένα κοινό task στα animation ενός παιχνιδιού είναι ο συνδυασμός μεταξύ δύο ή περισσότερων παρόμοιων κινήσεων. Τα Blend Trees χρησιμοποιούνται για να επιτρέπουν την ομαλή ανάμειξη πολλών κινούμενων σχεδίων ενσωματώνοντας τμήματα όλων αυτών σε διαφορετικούς βαθμούς. Το ποσοστό που κάθε μία από τις κινήσεις συμβάλλει στο τελικό αποτέλεσμα ελέγχεται χρησιμοποιώντας μια blending parameter. Για να έχει νόημα η ανάμειξη, οι κινήσεις που αναμιγνύονται πρέπει να έχουν παρόμοια φύση και χρονισμό. [9]

Στο Σχήμα 19 φαίνεται το Animator window που δείχνει ένα γράφημα ολόκληρου του Blend Tree. Στα αριστερά βρίσκεται ένα Blend Tree με μόνο τον root Blend Node (δεν έχουν προστεθεί ακόμη θυγατρικοί κόμβοι). Στα δεξιά βρίσκεται ένα Blend Tree με ρίζα Blend Node και τρία Animation Clips ως θυγατρικοί κόμβοι.



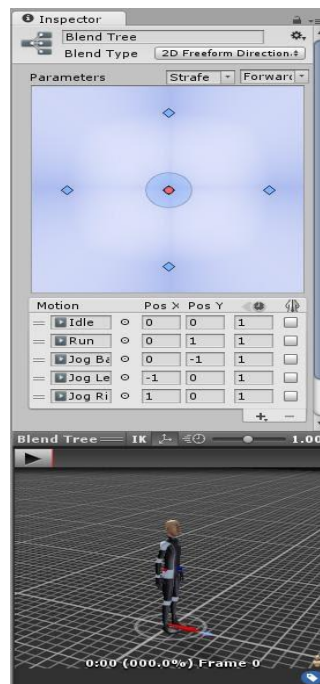
Σχήμα 19: Blend Tree

Στο Σχήμα 20 παρουσιάζεται το μενού περιβάλλοντος όταν κάνουμε δεξί κλικ σε έναν κόμβο Blend Tree.



Σχήμα 20: Μενού κόμβων Blend Tree

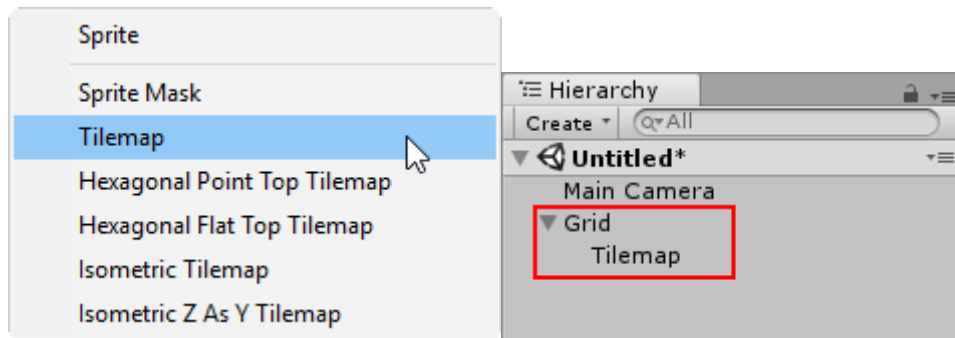
Στο Σχήμα 21 έχουμε ένα 2D Blend Tree που έχει δημιουργηθεί με πέντε animation clips, με προεπισκόπηση στον inspector.



Σχήμα 21: 2D Blend Tree

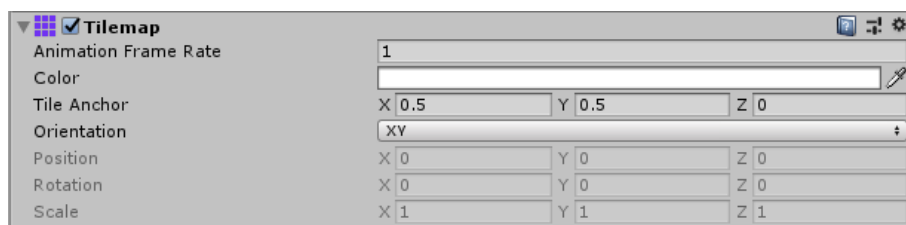
1. Tilemaps

Το Tilemap είναι ένα σύστημα που αποθηκεύει και διαχειρίζεται Tile Assets για τη δημιουργία επιπέδων 2D. Κατά τη δημιουργία ενός Tilemap (μενού: GameObject> 2D Object> Tilemap), το Unity δημιουργεί αυτόματα ένα Grid GameObject και το ορίζει ως γονέα ενός παιδιού Tilemap GameObject. Για τροποποίηση και τοποθέτηση Tiles σε χάρτη Tilemap, επιλέγεται το Tile Palette (μενού: Window> 2D> Tile Palette). (Σχήμα 22) [10]



Σχήμα 22: Δημιουργία Tilemap

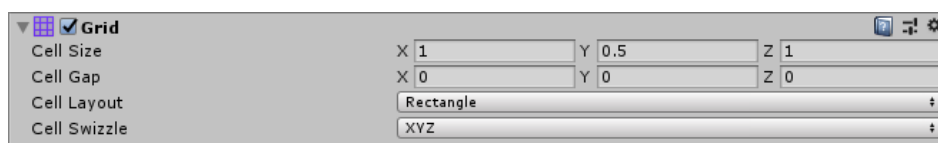
Στο Σχήμα 23 φαίνονται οι διάφορες ιδιότητες των Tilemaps που μπορούν να τροποποιηθούν από το χρήστη. Στο Animation Frame Rate μπορούμε να αλλάξουμε το ρυθμό με τον οποίο η Unity παίζει τα Tiles animation (π.χ. η επιλογή 2 ορίζει διπλάσια ταχύτητα). Στο Color γίνεται ο χρωματισμός του χάρτη με το επιλεγμένο χρώμα, ενώ στο Tile Anchor δίνονται οι συντεταγμένες θέσης ώστε να αντιστοιχηθεί η θέση του κελιού του Tile με το αντίστοιχο Grid. Τέλος, το Orientation καθορίζει τον προσανατολισμό των Tiles στο Tilemap.



Σχήμα 23: Tilemap window

1. Grid

Το Grid είναι ένας οδηγός που βοηθά στην ευθυγράμμιση των GameObjects, όπως τα Tiles, με βάση μια επιλεγμένη διάταξη. Μετατρέπει τις θέσεις του πλέγματος στις αντίστοιχες τοπικές συντεταγμένες του GameObject. Στη συνέχεια το Transform μετατρέπει αυτές τις τοπικές συντεταγμένες σε παγκόσμιο χώρο ή σε παγκόσμιες συντεταγμένες. [11]



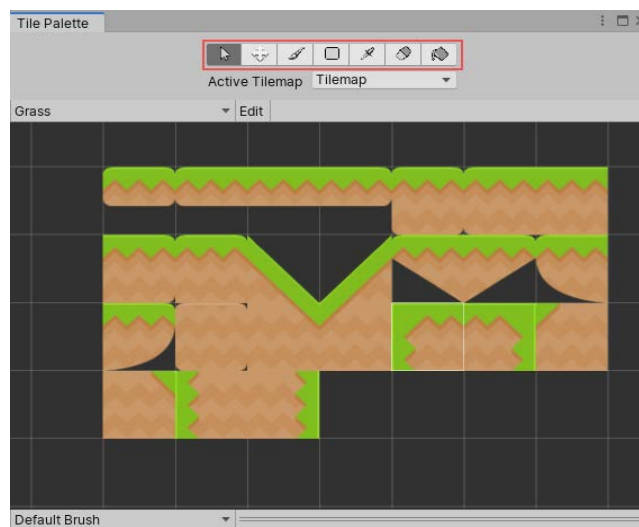
Σχήμα 24: Grid window

Στο Σχήμα 24 μπορούμε να δούμε τις ιδιότητες του Grid που μπορεί να τροποποιήσει ο χρήστης ώστε να το προσαρμόσει στο project του. Το Cell Size ορίζει το μέγεθος του κάθε κελιού στο Grid, το Cell Gap το κενό μεταξύ των κελιών και δεν μπορεί να πάρει αρνητική τιμή μικρότερη του Cell Size, το Cell Layout το σχήμα και τη διάταξη των κελιών και στο Cell Swizzle η Unity αναδιατάσσει τις συντεταγμένες των κελιών.

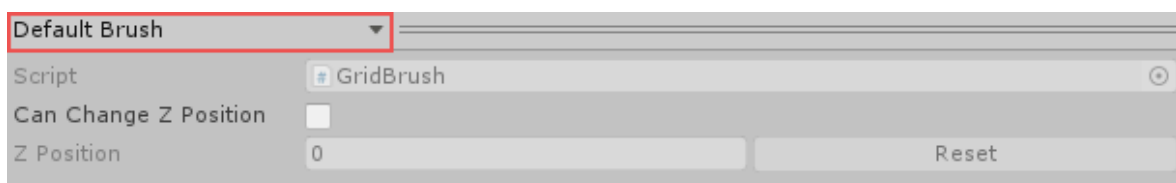
1. Painting

Σ' αυτήν την ενότητα περιγράφονται τα βήματα και τα εργαλεία για τη ζωγραφική των Tiles στο Tilemap. Για να ξεκινήσουμε, επιλέγουμε το Tilemap που θέλουμε να βάψουμε από το αναπτυσσόμενο μενού Active Tilemap στο παράθυρο Tilemap Palette. Τα Tilemap που δημιουργούνται στο Scene 's hierarchy προστίθενται αυτόματα στο αναπτυσσόμενο μενού. Μόνο τα τρέχοντα επιλεγμένα Active Tilemap επηρεάζονται από τα εργαλεία ζωγραφικής Tile Palette. [12]

Τα εργαλεία ζωγραφικής Tilemap βρίσκονται στη σειρά στο πάνω μέρος του παραθύρου Tile Palette. (Σχήμα 25) Το Active Brush Inspector βρίσκεται στο κάτω μέρος του παραθύρου του Tile Palette και επιτρέπει την αλλαγή του τρέχοντος ενεργού πινέλου και των ιδιοτήτων του. (Σχήμα 26)



Σχήμα 25: Tile Palette window



Σχήμα 26: Active Brush Inspector

1. Scriptable Brushes

Με τη χρήση script, ο δημιουργός του project μπορεί δημιουργήσει πινέλα για να βάψει αντικείμενα με βάση τη θέση και τις συνθήκες του κελιού. Η συμπεριφορά του πινέλου μπορεί να τροποποιηθεί περαιτέρω από το επιλεγμένο Εργαλείο επεξεργασίας, όπως το Erase ή το Floodfill. [13]

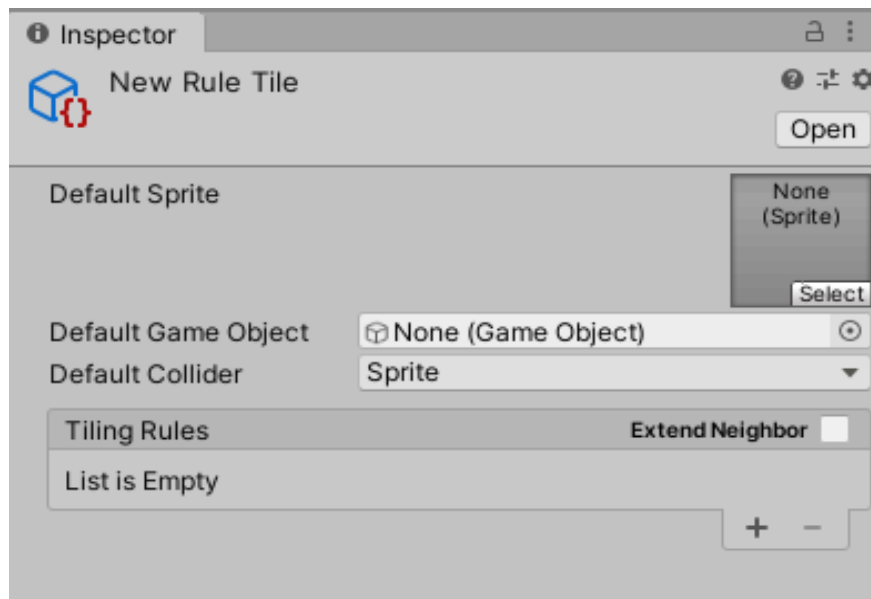
Τα είδη πινέλων που μπορούμε να αναπτύξουμε μέσω script είναι:

- **GameObject Brush:** Εμφανίζει, τοποθετεί και χειρίζεται το GameObjects στη σκηνή. Μπορεί, επίσης, να χρησιμοποιηθεί ως παράδειγμα για τη δημιουργία custom πινέλων που μπορούν να στοχεύσουν και να χειριστούν άλλα GameObjects ανάμεσα στα Tiles.
- **Group Brush:** Χρησιμοποιείται για την επιλογή Tiles που ομαδοποιούνται ανά θέση. Ορίζοντας το Gap value προσδιορίζεται ποια Tiles ανήκουν σε κάθε ομάδα και το Limit value κατοχυρώνει ότι η επιλεγμένη ομάδα παραμένει εντός του επιθυμητού μεγέθους.
- **Line Brush:** Φέρει μια γραμμή πάνω σε Tiles στο Tilemap.
- **Random Brush:** Τοποθετεί τυχαία Tiles πάνω στο Tilemap.

1. Rule Tiles

Το Rule Tile Asset είναι ένας αποτελεσματικός και προσαρμοστικός τρόπος βαφής Tiles σε ένα Scene. Συνήθως, όταν χρησιμοποιείται Tilemap, η βαφή των αντίστοιχων Tiles είναι σχετικά γρήγορη. Ωστόσο, κάποιες φορές η διαδικασία αυτή μπορεί να είναι επαναλαμβανόμενη και δυσκίνητη, ειδικά εάν πρέπει να γίνονται αλλαγές σ' αυτά. Τα Rule Tiles μπορούν να ταιριάξουν και να προσαρμοστούν στα Tiles που τα περιβάλλουν, προσφέροντας έναν γρήγορο, βολικό τρόπο προσθήκης ή τροποποίησης τους. Για παράδειγμα, όταν ζωγραφίζεται ένα Rule Tile στο Scene δίπλα σε υπάρχοντα Tiles, τα τελευταία θα ενημερώνονται αυτόματα. Ταυτόχρονα, μπορεί να ρυθμιστεί για να χειρίζεται animating Tiles, όρια και συγκρούσεις εν κινήσει, γεγονός που βοηθά στην επιτάχυνση της διαδικασίας δημιουργίας. [14]

Για τη δημιουργία ενός Rule Tile επιλέγουμε από το μενού Animating Assets > Create > Tiles > Rule Tile. Στη συνέχεια επιλέγουμε New Rule Tile για να τροποποιήσουμε τις ιδιότητές του στο παράθυρο Inspector. Οι ιδιότητες (Σχήμα 27) περιλαμβάνουν το Default Sprite ώστε να οριστεί το προεπιλεγμένο Sprite σε ένα από τα Tiles που θα επηρεάσει το Rule Tile, το Default Game Object το οποίο μπορεί να παραμείνει κενό, καθώς τα Rule Tiles θα οριστούν αργότερα, το Default Collider που καθορίζει τις φυσικές αλληλεπιδράσεις των Tiles και το Tilling Rules όπου ο χρήστης μπορεί να προσθέσει τόσους κανόνες όσα είναι και τα υπάρχοντα Tiles στο tileset.

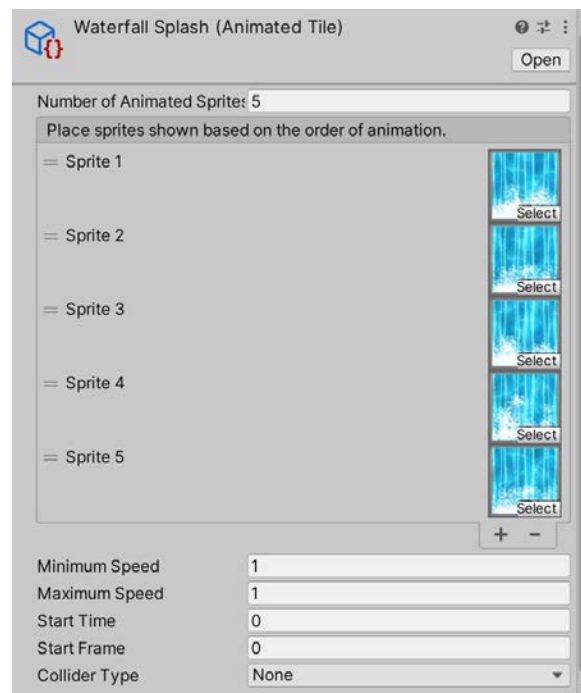


Σχήμα 27: Ιδιότητες των Rule Tiles

1. Animated Tiles

Τα Animated Tiles είναι Tiles, τα οποία έχουν εξ αρχής animation. Σ' αυτά υπάρχει μία λίστα από sprites που μπορούν να τροποποιηθούν, όπως φαίνονται στο Σχήμα 28: [15]

- **Αριθμός animated sprites:** Αριθμός animated sprites στο Animated Tile.
- **Sprite:** Το Sprite set για το Animated Tile, το οποίο θα παίζει διαδοχικά.
- **Minimum Speed:** Η ελάχιστη δυνατή ταχύτητα με την οποία θα παίζεται το Animation στο Tile. Μια τυχαία τιμή ταχύτητας θα επιλεγεί μεταξύ της ελάχιστης και της μέγιστης ταχύτητας.
- **Maximum Speed:** Η μέγιστη δυνατή ταχύτητα με την οποία θα παίζεται το Animation στο Tile.
- **Start Time:** Η ώρα έναρξης αυτού του Animated Tile.
- **Start Frame:** Το αρχικό frame του συγκεκριμένου Animated Tile. Η επιλογή αυτή επιτρέπει στο χρήστη να ξεκινήσει το Animation από ένα συγκεκριμένο Sprite στη λίστα των Animated Sprites.
- **Collider Type:** Το Collider Shape δημιουργείται από το Tile.



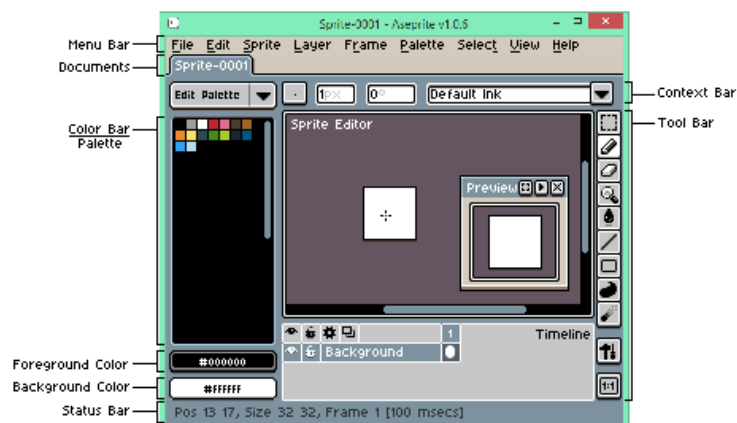
Σχήμα 28: Animated Tile

3.2 Aseprite

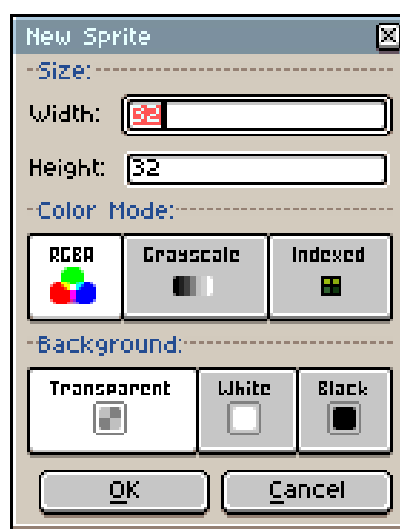
Το Aseprite είναι ένα πρόγραμμα που επιτρέπει στον χρήστη την δημιουργία 2D γραφικών και Animation για βιντεοπαιχνίδια. Από sprites μέχρι και pixel-art, καλύπτει τα πάντα από την εποχή των 8 και 16 bit. Η χρήση του στην πτυχιακή αυτή ήταν για την δημιουργία των γραφικών περιβάλλοντος του παιχνιδιού, καθώς και πιο συγκεκριμένα την βάση των tiles που θα χρησιμοποιηθούν για την αυτόματη δημιουργία πιστών. Ενώ υπάρχουν διάφορα άλλα προγράμματα για την δημιουργία γραφικών Pixel art, το Aseprite είναι εξειδικευμένο πάνω σε αυτό και παρέχει συγκεντρωμένα όλες τις λειτουργίες που μπορεί να χρειαστεί κάποιος σε ένα πρόγραμμα.

3.2.1 Διαδικασία σχεδιασμού

Ανοίγοντας το πρόγραμμα ερχόμαστε αντιμέτωποι με την αρχική σελίδα του (Σχήμα 29), από όπου διαλέγουμε από το Menu Bar: File->New για την δημιουργία καινούργιου sprite και εισάγουμε τις παραμέτρους που θέλουμε για το αρχείο μας (Σχήμα 30).



Σχήμα 29: Αρχική σελίδα Sprite



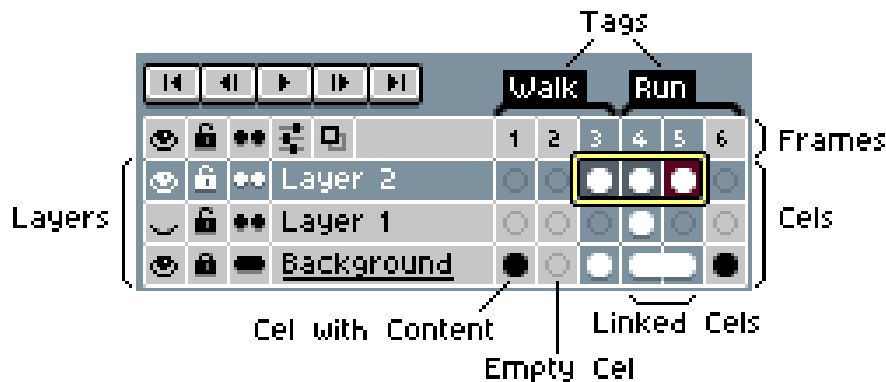
Σχήμα 30: Εισαγωγή παραμέτρων Νέου Sprite

Χρησιμοποιώντας την παλέτα εργαλείων που μας δίνεται δεξιά προχωράμε στον σχεδιασμό του γραφικού μας. Τα εργαλεία τα οποία μας παρέχονται είναι τα εξής:

- Pencil
- Line
- Curve
- Rectangle
- Ellipse
- Countour
- Polygon
- Eraser
- Eyedropper
- Rectangular Marquee
- Move Cel
- Slice

3.2.2 Animation

Το animation είναι η ψευδαίσθηση κίνησης που δημιουργείται όταν εναλλάσσονται με μια συγκεκριμένη σειρά εικόνες που αναπαριστούν τις προοδευτικές φάσεις της κίνησης αυτής. Το Aseprite το πετυχαίνει αυτό με την χρήση των frames που είναι ξεχωριστά sprites στο ίδιο αρχείο μέσα. Τον έλεγχο αυτών τον πετυχαίνουμε με την χρήση του timeline (Σχήμα 31) Από εκεί μπορούμε να διαλέξουμε την σειρά και την ταχύτητα αναπαραγωγής του κάθε frame καθώς και τον διαχωρισμό τους σε κάθε διαφορετική κίνηση.



Σχήμα 31: Aseprite Timeline

3.2.3 Αποθήκευση και εξαγωγή

Αφού φτάσουμε στο επιθυμητό αποτέλεσμα η αποθήκευση του αρχείου μπορεί να γίνει με διάφορες μεθόδους. Η αποθήκευση σε τύπο .ase που είναι αρχείο μόνο αναγνώρισιμο από το aseprite μας επιτρέπει να κρατήσουμε πληροφορίες όπως animation, layers και διάφορες άλλες παραμέτρους που επιτρέπουν την συνέχιση της επεξεργασίας. Σε περίπτωση που θέλουμε να εξάγουμε το αρχείο σε κάποιο άλλο πρόγραμμα, είναι δυνατή η αποθήκευση ως jpg ή png και ως gif για τα animations (Σχήμα 32). Αποθηκεύοντας τα με κάποιον από αυτούς τους τύπους είναι δυνατή η επεξεργασία τους στο aseprite μετά, αλλά χάνονται οι περαιτέρω πληροφορίες.



Σχήμα 32: Αποθήκευση αρχείου

3.3 Tilesetter

Το Tilesetter περιγράφεται ως «ένα εργαλείο δημιουργίας συνόλου πλακιδίων και επεξεργασίας χαρτών», καθώς ο χρήστης του λογισμικού μπορεί να δημιουργήσει γρήγορα και εύκολα ολόκληρο το σύνολο πλακιδίων που χρειάζεται ή ολοκληρωμένα σετ πλακιδίων από εικόνες βάσης. Τα σετ μπορούν να εξαχθούν απευθείας στα Unity, Godot και GameMaker Studio 2, ενώ παράλληλα πρόκειται και για ένα πρόγραμμα επεξεργασίας εικόνας. [16] Παρακάτω παρουσιάζονται οι λειτουργίες για την έκδοση 2.0.0 του λογισμικού αλλά πολλές από αυτές ισχύουν και για παλαιότερες εκδόσεις.

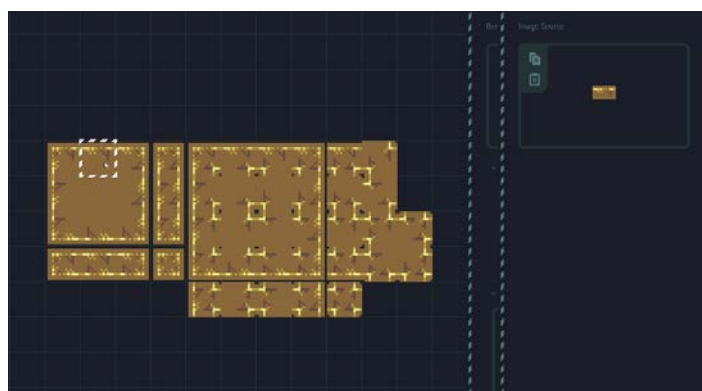
3.3.1 Tiles και Sources

Τα πλακίδια και οι πηγές είναι το θεμέλιο του Tilesetter. Αν και η βασική ιδέα των πλακιδίων είναι αυτονόητη, το Tilesetter διαθέτει επίσης την έννοια των Πηγών, οι οποίες διαδραματίζουν σημαντικό ρόλο στη δημιουργία συνόλων πλακιδίων. Υπάρχει σημαντική διάκριση μεταξύ αυτών των δύο και του τρόπου που συνεργάζονται (Σχήμα 33). [17]



Σχήμα 33: Διάκριση και συνεργασία των Tiles και Sources

Στο Tilesetter, η εμφάνιση ενός πλακιδίου δεν είναι κάτι εγγενές στο πλακίδιο. Αντίθετα, τα δεδομένα εικόνας αποθηκεύονται στις Πηγές, οι οποίες στη συνέχεια χρησιμοποιούνται για να δημιουργήσουν την τελική εμφάνιση ενός πλακιδίου. Αυτή η αποκόλληση επιτρέπει την επαναχρησιμοποίηση της ίδιας εικόνας σε διαφορετικά πλακίδια, καθώς πολλά απ' αυτά μπορούν να αναφέρουν την ίδια πηγή για τα γραφικά τους. Αυτό είναι ιδιαίτερα χρήσιμο όταν δημιουργείτε σετ πλακιδίων.



Σχήμα 34: Λειτουργία Tiles και Sources στη δημιουργία σετ πλακιδίων

Τα πλακίδια είναι τα δομικά στοιχεία για περιβάλλοντα που βασίζονται σε πλακίδια και αποτελούν το επίκεντρο του Tilesetter. Η εμφάνισή τους συντίθεται από μια συλλογή Πηγών. Ένα βασικό πλακίδιο

θα χρησιμοποιεί μία μόνο πηγή για την εικόνα του, ωστόσο, τα πλακίδια που δημιουργούνται μπορεί να συνδυάζουν πολλαπλές πηγές για να επιτύχουν την εμφάνισή τους. Εκτός από τα γραφικά, τα πλακίδια μπορούν επίσης να έχουν καθορισμένη συμπεριφορά που χρησιμοποιείται για αυτόματη τοποθέτηση πλακιδίων.

3.3.2 Επεξεργασία πλακιδίων

Όλα τα πλακίδια σε ένα έργο Tilesetter βρίσκονται στο Set View. Πρόκειται για ένα περιβάλλον που βασίζεται σε πλέγμα, όπου τα πλακίδια μπορούν να τοποθετηθούν ελεύθερα και είναι εύκολα προσβάσιμα για επεξεργασία. Βρίσκεται στην επάνω αριστερή περιοχή της οθόνης και εκεί πραγματοποιείται το μεγαλύτερο μέρος της εργασίας. Το αριστερό πλήκτρο του ποντικιού χρησιμοποιείται για την επιλογή κελιών στο πλέγμα και πολλά κελιά μπορούν να επιλεγούν κάνοντας κλικ και σύροντας. Με το δεξί κλικ εμφανίζεται το μενού περιβάλλοντος, επιτρέποντας την εκτέλεση πολλών λειτουργιών στα επιλεγμένα κελιά. [18]

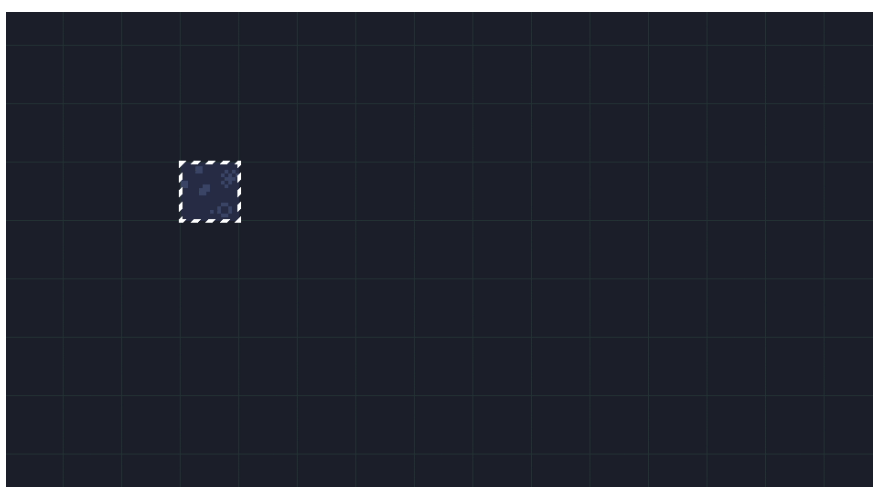
Οι βασικές λειτουργίες είναι:

- **Επιλογή.** Μπορεί να πραγματοποιηθεί ή κάνοντας κλικ στο αριστερό κουμπί του ποντικιού ή κάνοντας κλικ και σύροντας σε ένα μη επιλεγμένο κελί για να επιλεγούν πολλά κελιά ή κρατώντας πατημένο το Shift για προσθήκη στην τρέχουσα επιλογή.
- **Μετακίνηση.** Μπορεί να πραγματοποιηθεί σύροντας το αριστερό κουμπί του ποντικιού σε ένα επιλεγμένο κελί.
- **Διαγραφή.** Διαγραφή κλειδιού.
- **Μετατόπιση.** Μπορεί να πραγματοποιηθεί σύροντας το μεσαίο κουμπί του ποντικιού ή σύροντας το αριστερό κουμπί του ποντικιού κρατώντας πατημένο το Alt.
- **Ζουμ.** Τροχός ποντικιού ή Ctrl++/Ctrl+- (Command++/Command+- σε macOS).
- **Αντιγραφή.** Υπάρχουν δύο διαφορετικές λειτουργίες αντιγραφής για πλακίδια στην προβολή συνόλου.
 - Έξυπνη αντιγραφή. Χρησιμοποιώντας Ctrl+C (Command+C σε macOS) ή κάνοντας δεξί κλικ και επιλέγοντας «Αντιγραφή» θα αντιγραφούν τα πλακίδια με τέτοιο τρόπο ώστε τα πλακίδια που επικολλήθηκαν αργότερα να αναφέρουν τις ίδιες Πηγές με τα αρχικά. Αυτή η λειτουργία δεν αντιγράφει δεδομένα εικόνας και επομένως δεν επιτρέπει την επικόλληση εκτός του Tilesetter.
 - Raw Copy. Χρησιμοποιώντας Ctrl+Shift+C (Command+Shift+C σε macOS) ή κάνοντας δεξί κλικ και επιλέγοντας «Copy Raw» θα αντιγραφούν τα ακατέργαστα δεδομένα εικόνας των πλακιδίων για επικόλληση εντός ή εκτός Tilesetter.
 - **Διαχείριση έργου.** Στο επάνω αριστερό μέρος του Set View υπάρχουν κουμπιά για τη φόρτωση, την αποθήκευση ή την προετοιμασία ενός έργου.

3.3.3 Δημιουργία πλακιδίων

Η παραδοσιακή διαδικασία δημιουργίας ενός σετ πλακιδίων συνήθως μπορεί να είναι μια χρονοβόρα εργασία όταν γίνεται με το χέρι, αλλά το Tilesetter απλοποιεί πολύ τη δουλειά με τις δυνατότητες παραγωγής tileset. Καθορίζοντας τουλάχιστον μία εικόνα βάσης και μία εικόνα άκρης, το Tilesetter είναι σε θέση να δημιουργήσει γρήγορα ένα πλήρες σύνολο πλακιδίων από αυτά τα στοιχεία. Τα πλακίδια δημιουργούνται από ένα ή περισσότερα κεντρικά πλακίδια. [19]

Κάνοντας δεξί κλικ σε ένα πλακίδιο βάσης και επιλέγοντας «Δημιουργία περιγραμμάτων», θα δημιουργηθούν όλα τα κομμάτια περιγράμματος για το επιλεγμένο πλακίδιο. Οι λευκές διακεκομμένες γραμμές (Σχήμα 35) είναι σύμβολα κράτησης θέσης για υφές άκρων που θα καθορίσει ο χρήστης. Αυτή είναι η προεπιλεγμένη κατάσταση ενός συνόλου πλακιδίων που δημιουργήθηκε πρόσφατα και από εδώ θα διαμορφωθεί το πώς θα φαίνονται τα περιγράμματα του, εάν θα έχει προσαρμοσμένες γωνίες και άλλες λεπτομέρειες.



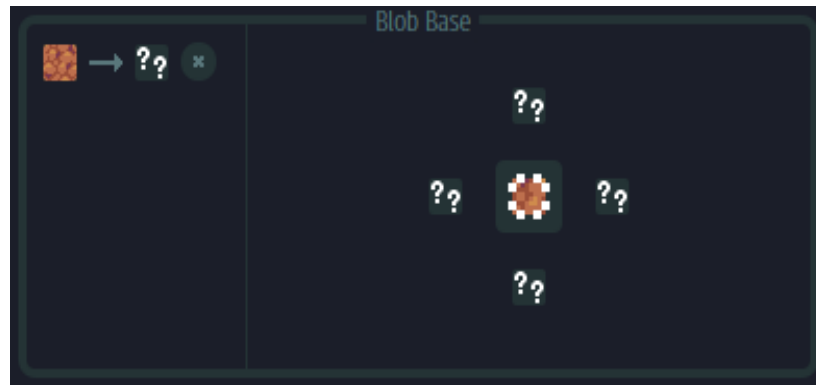
Σχήμα 35: Δημιουργία πλακιδίων

3.3.4 Συμπεριφορά πλακιδίων

Τα πλακίδια έχουν συχνά μια άρρητη σχέση μεταξύ τους στο πλαίσιο της επεξεργασίας και της δημιουργίας χαρτών. Για παράδειγμα, πολλά πλακίδια μπορεί να αποτελούν ένα μεμονωμένο αντικείμενο κατά τη σχεδίαση χαρτών, ενώ μεμονωμένα πλακίδια σε αυτήν την ομάδα μπορεί να μην έχουν νόημα όταν σχεδιάζονται μόνα τους. [20]

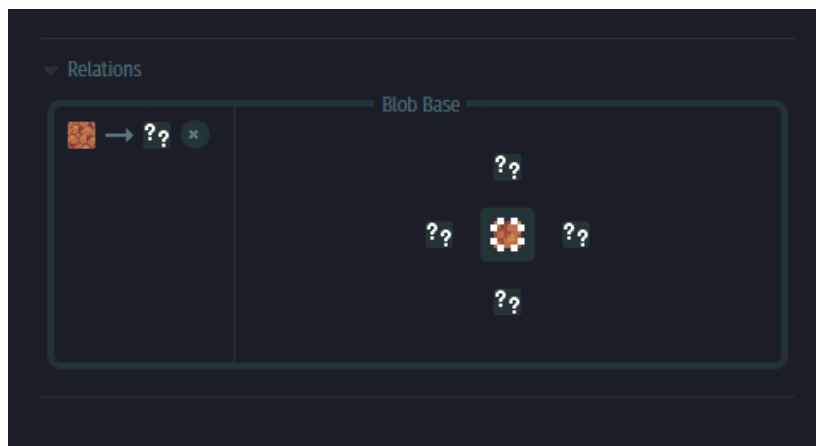
Στο Tilesetter, τα Relations χρησιμοποιούνται για να περιγράψουν πώς σχετίζονται τα πλακίδια με άλλα. Αντιστοιχίζονται σε πλακίδια μέσω λειτουργιών που εκτελούνται σε αυτά, ενώ ορισμένοι τύποι Relations συνδέουν πλακίδια τόσο σε περιβάλλον παραγωγής όσο και σε περιβάλλον επεξεργασίας χάρτη, ενώ άλλοι χρησιμοποιούνται μόνο για το τελευταίο. Τα Relations ενός πλακιδίου μπορούν να προβληθούν στην προβολή Ιδιοτήτων πλακιδίων στην καρτέλα «Relations».

Επιλέγοντας ένα κεντρικό πλακίδιο στο σετ, εμφανίζονται οι λεπτομέρειες των Relations στην Προβολή Ιδιοτήτων πλακιδίων. Στο Σχήμα 36, η αριστερή πλευρά δείχνει όλα τα γειτονικά πλακίδια στα οποία το σετ έχει περιθώρια για σύνδεση και η δεξιά πλευρά δείχνει ποια πλακίδια περιγράμματος χρησιμοποιούνται για κάθε σχηματισμό γειτόνων.



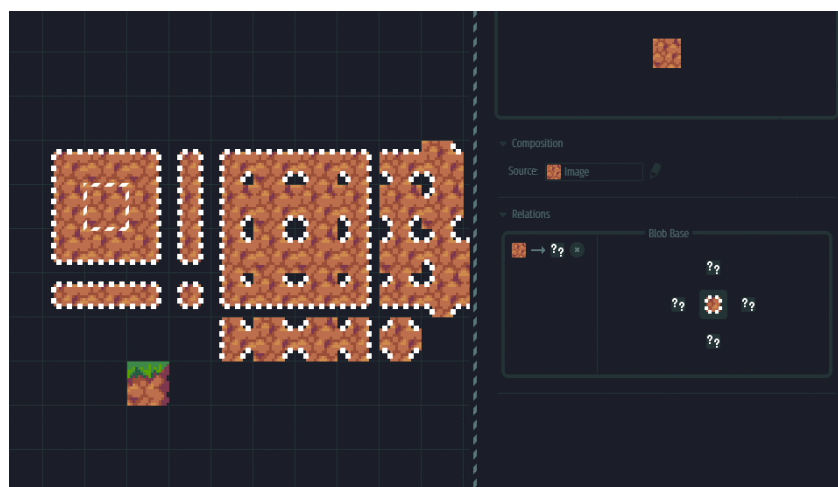
Σχήμα 36: Λειτουργία Relations

Κάνοντας κλικ στις μικρότερες υποδοχές γύρω από την κεντρική οθόνη στα δεξιά, είναι δυνατή η περιήγηση σε πιθανούς γείτονες, κάνοντας η κεντρική υποδοχή να εμφανίζει το πλακίδιο που χρησιμεύει ως περίγραμμα για αυτό το σενάριο. (Σχήμα 37)



Σχήμα 37: Λειτουργία Relations 2

Κάνοντας κλικ στην κεντρική υποδοχή, μπορεί να αλλάξει το συγκεκριμένο πλακίδιο περιγράμματος με οποιοδήποτε άλλο. Αυτό επιτρέπει τη χρήση προσχεδιασμένων πλακιδίων για αυτόματη τοποθέτηση πλακιδίων στο πρόγραμμα επεξεργασίας χαρτών. (Σχήμα 38)



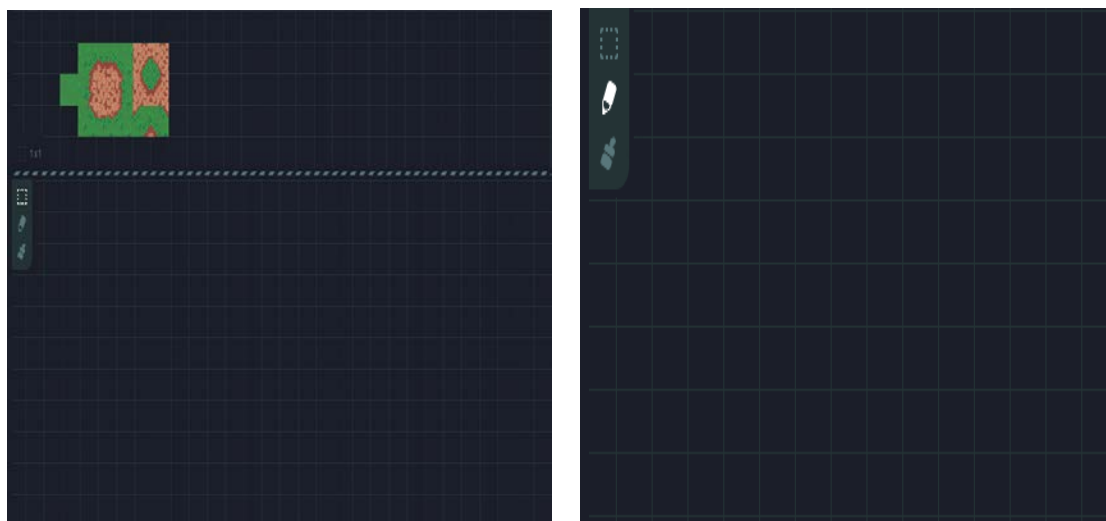
Σχήμα 38: Λειτουργία Relations 3

3.3.5 Επεξεργασία χάρτη

Η κύρια λειτουργία του Tiled είναι να δημιουργεί γρήγορα σύνολα πλακιδίων, ωστόσο, μπορεί να είναι χρήσιμη η προβολή τους σε πραγματικό περιβάλλον. Εξαιτίας αυτού, το Tiled περιλαμβάνει επίσης ένα πρόγραμμα επεξεργασίας χαρτών όπου τα πλακίδια που έχουν δημιουργηθεί μπορούν να δοκιμαστούν εν κινήσει. [21]

Τα υπάρχοντα πλακίδια στο έργο μπορούν να σχεδιαστούν ελεύθερα στην προβολή Sandbox με το εργαλείο μολυβιού. Αυτή η λειτουργία επεξεργασίας μπορεί να επιλεγεί μέσω του μεσαίου εικονιδίου επάνω αριστερά ή πατώντας το εικονίδιο με το μολύβι. Το μολύβι επιτρέπει το σχεδιασμό των πλακιδίων που υπάρχουν στο έργο επιλέγοντάς τα πρώτα στο Set View.

Μόλις επιλεγεί ένα επιθυμητό πλακίδιο, ο χρήστης μπορεί να χρησιμοποιήσει το αριστερό κουμπί του ποντικιού για να κάνει κλικ και να σύρει γύρω από το Sandbox View για να σχεδιάσει το επιλεγμένο πλακίδιο (Σχήμα 39). Αν χρησιμοποιήσει το δεξί κουμπί του ποντικιού, θα διαγραφούν τα πλακίδια που υπάρχουν στον χάρτη. Επιλέγοντας πολλά πλακίδια παραπάνω, το εργαλείο θα σφραγίσει ολόκληρη την επιλογή, διασφαλίζοντας ότι οι διαδοχικές σφραγίδες δεν επικαλύπτονται.

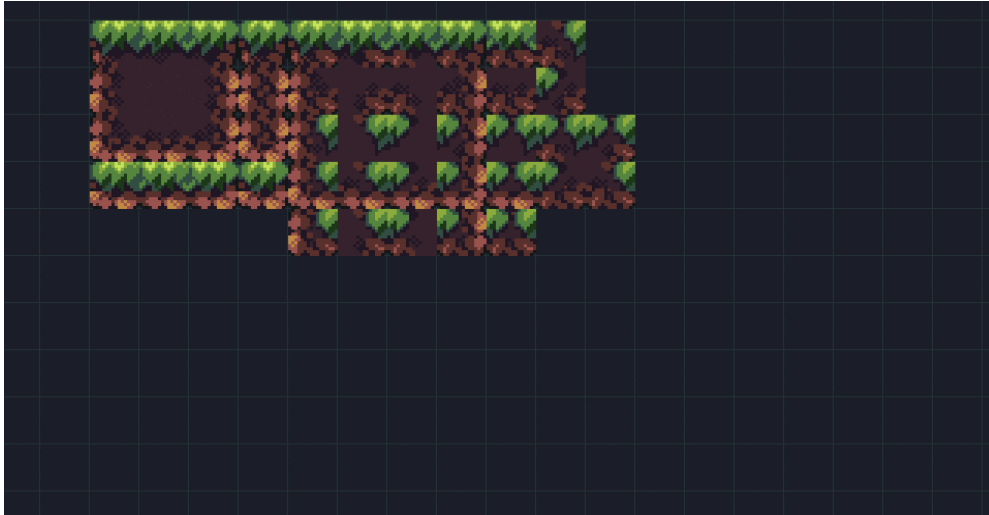


Σχήμα 39: Επεξεργασία χάρτη

3.3.6 Εξαγωγή project

Το Tiled είναι δυνατή η εξαγωγή των πλακιδίων και χαρτών που έχουν δημιουργηθεί σ' αυτό σε μια ποικιλία διαφορετικών μορφών, με μερικά από αυτά να μπορούν εύκολα να εισαχθούν σε μηχανές παιχνιδιών. Σε περίπτωση που η μορφή που αναζητείται δεν υποστηρίζεται, μπορεί να προστεθεί ορίζοντας προσαρμοσμένες μορφές. [22]

Η εξαγωγή πραγματοποιείται είτε σε μεμονωμένα πλακίδια για επεξεργασία χαρτών σε εξωτερικό λογισμικό είτε σε προκατασκευασμένους χάρτες που έχουν σχεδιαστεί στο πρόγραμμα επεξεργασίας χαρτών. Για να εξαχθούν μεμονωμένα πλακίδια, επιλέγεται μια περιοχή στο Set View, δεξί κλικ και οποιαδήποτε επιλογή στο «Εξαγωγή». (Σχήμα 40) Η εξαγωγή πλακιδίων με αυτόν τον τρόπο θα λάβει υπόψη τα Relations, διατηρώντας τη συμπεριφορά αυτόματης τοποθέτησης πλακιδίων ανάλογα με την επιλεγμένη μορφή.



Σχήμα 40: Εξαγωγή Tiles

Οι χάρτες μπορούν να εξαχθούν κάνοντας το ίδιο στην προβολή Sandbox. Σε αυτήν την περίπτωση, μόνο τα πλακίδια που εμφανίζονται στην επιλογή θα συμπεριληφθούν στην εξαγωγή και καμία από τις συμπεριφορές των πλακιδίων δεν θα ληφθεί υπόψη. (Σχήμα 41)

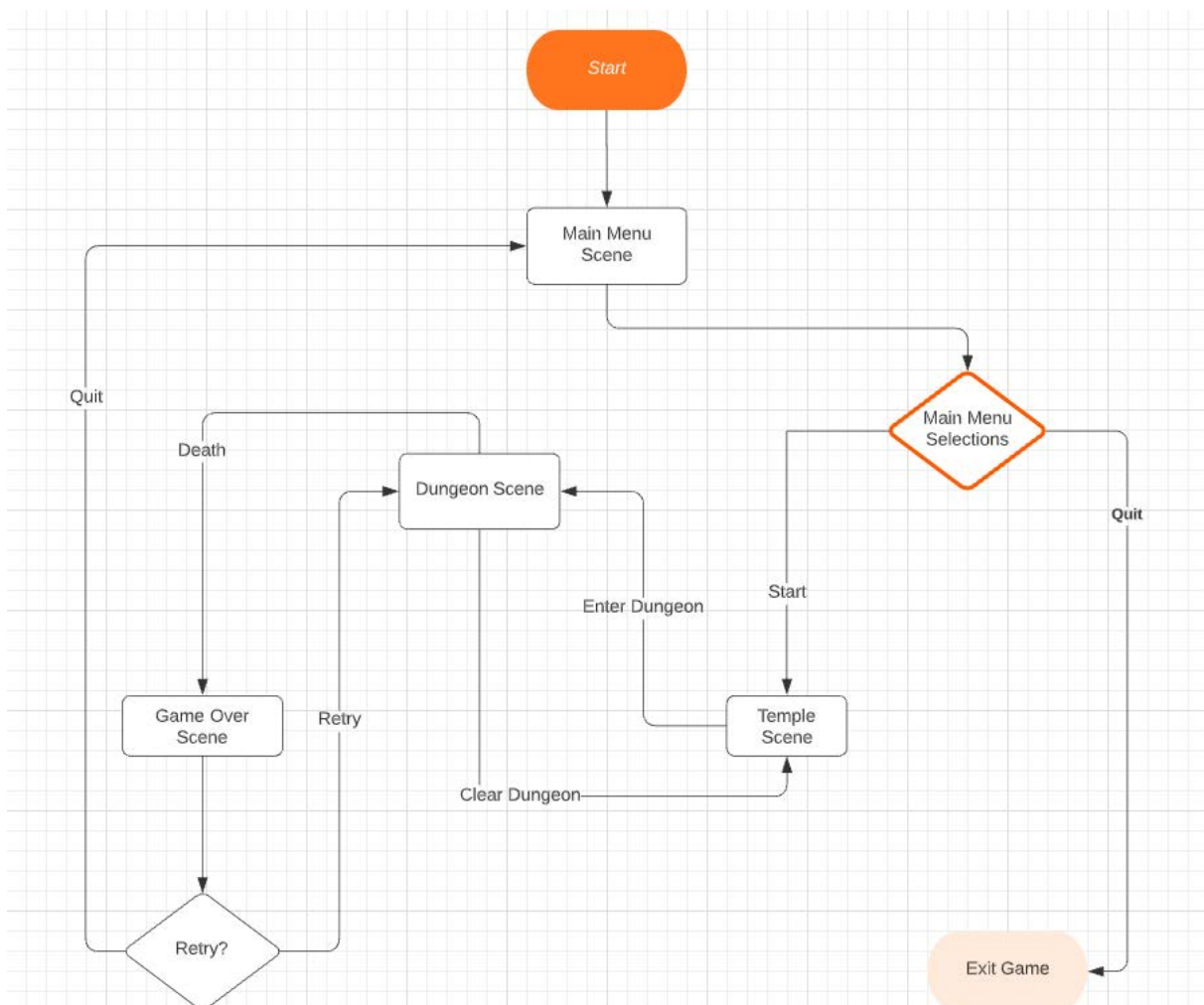


Σχήμα 41: Εξαγωγή Maps

Κεφάλαιο 4ο: Διαδικασία Ανάπτυξης

4.1 Επεξήγηση Παιχνιδιού

Αφού αναλύθηκαν τα θεωρητικά κομμάτια των προγραμμάτων που χρησιμοποιήθηκαν, σε αυτό το κεφάλαιο θα αναλύσουμε το πρακτικό μέρος, την δημιουργία του ίδιου του παιχνιδιού (Σχήμα 43). Το παιχνίδι το οποίο δημιουργήθηκε είναι το Top-Down 2D RPG παιχνίδι στο οποίο σκοπός του παίκτη είναι να πάει στην σπηλιά με τα τέρατα, και να τα εξοντώσει. Αυτό γίνεται σε 3 διαφορετικές σπηλιές όπου με την ολοκλήρωση τους ο παίκτης κερδίζει (Σχήμα 42). Εκτός από τα στοιχεία του ίδιου του Unity, υπάρχουν πολλά κομμάτια κώδικα τα οποία δουλεύουν ταυτόχρονα για να λειτουργήσει όλο αυτό, τα οποία θα αναλυθούν σε αυτό το κεφάλαιο.



Σχήμα 42 Use Case Scenario μιας εκτέλεσης του παιχνιδιού



Σχήμα 43 Αρχική οθόνη παιχνιδιού

4.2 Ανάλυση λειτουργιών

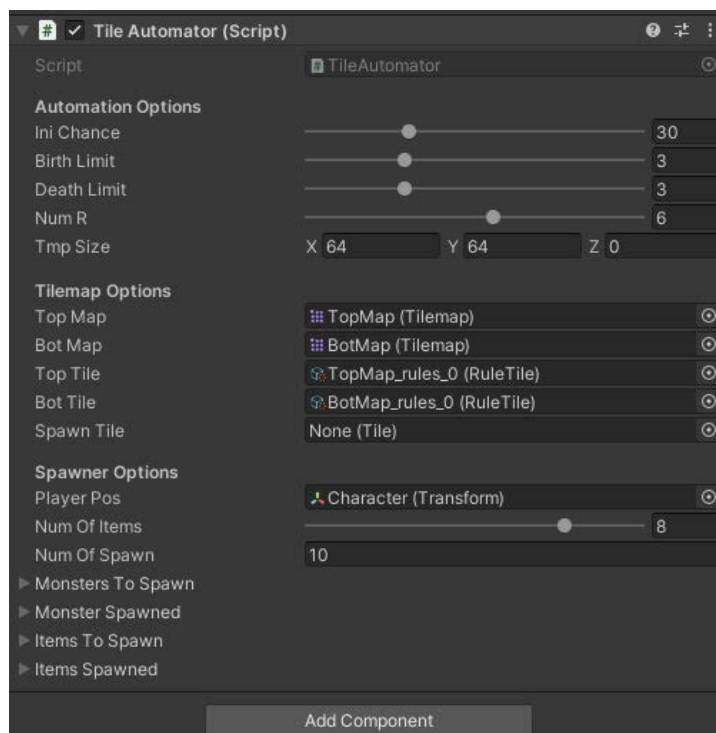
4.2.1 Κυτταρικό Αυτόματο

Μεγάλο κομμάτι του παιχνιδιού βασίζεται στο Κυτταρικό Αυτόματο. Τα κυτταρικά αυτόματα είναι δομές που μοντελοποιούν τη συμπεριφορά κατανεμημένων συστημάτων με συγκεκριμένη χωρική (γεωμετρική) διάταξη. Ονομάζονται "κυτταρικά" επειδή είναι εμπνευσμένα από κυτταρικές διατάξεις όπου κάθε κύτταρο αλληλοεπιδρά με τα γειτονικά του στο χώρο και είναι "αυτόματα" επειδή η συμπεριφορά κάθε κυττάρου είναι πλήρως καθορισμένη από το αντίστοιχο μοντέλο, άρα "αυτόματη" [23]. Το Script αυτό δημιουργεί έναν χάρτη τυχαία βάση των κανόνων του Game of Life του Conway. Το Game of Life είναι ένα κυτταρικό αυτόματο το οποίο φτιάχτηκε από τον Βρετανό μαθηματικό John Horton Conway το 1970 με σκοπό την προσομοίωση οργανικών δομών. Λειτουργεί πάνω σε ένα πλέγμα στο οποίο ορίζει ένα στοιχείο του ως ζωντανό γεμίζοντας το ή νεκρό αφήνοντας το κενό. Η κατάσταση ενός κελιού ορίζεται από την κατάσταση των γειτονικών του κελιών και ένα σετ κανόνων το οποίο είναι:

1. Οποιοδήποτε ζωντανό κελί με δύο ή τρεις ζωντανούς γείτονες, παραμένει ζωντανό
2. Οποιοδήποτε νεκρό κελί με τρεις ζωντανούς γείτονες αλλάζει σε ζωντανό
3. Όλα τα υπόλοιπα ζωντανά κελιά πεθαίνουν στην επόμενη επανάληψη και ομοίως όλα τα υπόλοιπα νεκρά κελιά παραμένουν νεκρά

Χρησιμοποιώντας αυτή τη λογική και παραμετροποιώντας την, δημιουργήθηκε ένα script που παράγει τυχαίους χάρτες βάση των παραμέτρων που εισάγουμε (Σχήμα 44) , που θα είναι η βάση της σπηλιάς του παιχνιδιού. Πιο συγκεκριμένα έχουμε:

Automation Options: Οι ρυθμίσεις όσον αφορά το κυτταρικό αυτόματο. Αλλάζοντας αυτές ο χάρτης που παράγεται αλλάζει δραστικά (Σχήμα 45):



Σχήμα 44: Παράμετροι του Tile Automator

- **Ini Chance:** Η αρχική πιθανότητα ενός κελιού να είναι νεκρό ή ζωντανό.
- **Birth Limit:** Το όριο των ζωντανών κελιών που πρέπει να έχει ένα νεκρό κελί για να μπορέσει να αλλάξει κατάσταση στην επόμενη επανάληψη της προσομοίωσης.
- **Death Limit:** Το όριο των νεκρών κελιών που πρέπει να έχει ένα ζωντανό κελί για να μπορέσει να αλλάξει κατάσταση στην επόμενη επανάληψη της προσομοίωσης.
- **Num R:** Ο αριθμός των επαναλήψεων που γίνεται με την εντολή που γίνεται μέσα από το Script.
- **Tmp Size:** Το μέγεθος σε pixel του παραχθέντος χάρτη.

Tilemap Options: Οι ρυθμίσεις που αφορούν τα γραφικά που θα χρησιμοποιηθούν για την παραγωγή του χάρτη:

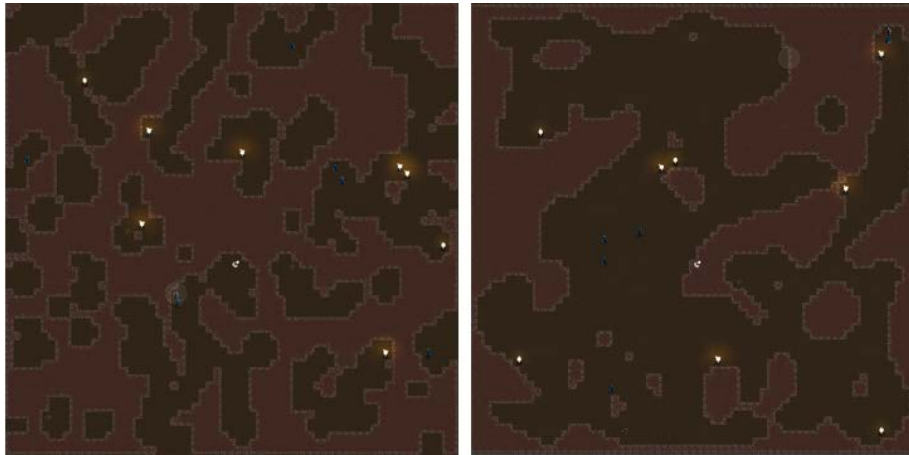
- **Top/Bot Map:** Τα tilemaps που υπάρχουν στην σκηνή ως Grid που θα δημιουργηθεί το ο χάρτης του παιχνιδιού.
- **Top/Bot Tile:** Τα rule Tiles με τα οποία θα σχεδιαστούν τα παραπάνω Tilemaps. Αλλάζοντας τα μπορούν να δημιουργηθούν διάφορα είδη πιστών όπως για παράδειγμα πέτρα και λάβα, ή γρασίδι και θάλασσα, δίνοντας έτσι στο παιχνίδι ποικιλία χωρίς περισσότερη δυσκολία.
- **Spawn Tile:** Χρησιμοποιείται για λόγους συντήρησης κυρίως, που θα εξηγηθούν παρακάτω στην ανάλυση του κώδικα.

Spawner Options: Ρυθμίσεις που αφορούν την προσθήκη αντικειμένων στον χάρτη που έχει παραχθεί.

- **Player Pos:** Μεταβλητή σχετική με την θέση του χαρακτήρα στον χάρτη. Χρησιμοποιείται για την μεταφορά του παίκτη σε τυχαίο σημείο του χάρτη όταν παραχθεί.
- **Spawn/Item Num:** Ο αριθμός των αντιπάλων και των αντικειμένων που θα εμφανιστούν στον χάρτη.
- **Monster/Items To Spawn:** Μια λίστα με gameobjects μέσα από την οποία θα επιλεγθούν τυχαία τα αντικείμενα και τέρατα των οποίων ο αριθμός έχει οριστεί από το Spawn/Item Num.

- **Monster/Items Spawned:** Μια λίστα με τα δημιουργημένα αντικείμενα στον χάρτη, χρησιμοποιείται για την διαγραφή τους σε κάθε επανάληψη της προσομοίωσης.

Χρησιμοποιώντας τις παραπάνω ρυθμίσεις μπορεί να βγει μια πληθώρα αποτελεσμάτων με μικρές αλλαγές.



Σχήμα 45: Παράδειγμα χάρτη με Ini Chance 30, Birth 3 Death 3 και 7 επαναλήψεις(Αριστερά) και Ini chance 50, Birth 4, Death 3 και 3 επαναλήψεις(Δεξιά)

Στο κομμάτι του κώδικα ξεκινάμε από την δήλωση όλων των παραπάνω σαν μεταβλητές μαζί και με κάποιες ιδιωτικές μεταβλητές οι οποίες δεν χρήζουν αλλαγή από το περιβάλλον του Unity.

Το width και height ορίζει το πλάτος και το ύψος του χάρτη τα οποία εξάγει από το Vector3 tmpSize που βάζουμε στο editor. Τα xRand και yRand είναι τυχαίες μεταβλητές που θα χρησιμοποιηθούν αργότερα για την τυχαία δημιουργία αντικειμένων πάνω στον χάρτη. Το terrainMap είναι ένας πίνακας που έχει μέγεθος όσες και τις θέσεις του χάρτη και 2 καταστάσεις, 0 για νεκρό κελί και 1 για ζωντανό κελί.

Η βασική μέθοδος του TileAutomator είναι η doSim(int num) που τρέχει την προσομοίωση όσες φορές ορίζει το num. Αρχίζει ελέγχοντας άμα έχει αρχικοποιηθεί το terrainMap και σε περίπτωση που δεν έχει αρχικοποιηθεί δημιουργεί νέο πίνακα int 2 διαστάσεων με ορίσματα το width και το height και καλεί την μέθοδο initPos()

Στο initPos() υπάρχει μόνο μια εμφωλευμένη επανάληψη που περνάει από όλα τα στοιχεία του πίνακα terrainMap[x,y] και δίνει μια τιμή 0 ή 1 βάση του iniChance που έχουμε ορίσει στον editor

- Αφού οριστεί ο πίνακας το επόμενο βήμα είναι η εκτέλεση της προσομοίωσης βάση του numR που έχουμε δηλώσει στον editor, οπότε δημιουργούμε μια for επανάληψη που καλεί την μέθοδο genTilePos στο terrainMap με όρισμα το ίδιο το terrainMap, ανανεώνοντας σε κάθε βήμα τις καταστάσεις των κελιών.
- Στην genTilePos δημιουργείται η καινούργια κατάσταση του χάρτη έχοντας ως βάση την προηγούμενη επανάληψη. Ορίζουμε δύο μεταβλητές, την neighb τύπου int που θα είναι η κατάσταση των γειτονικών κελιών και την myB τύπου BoundsInt με την οποία θα ορίσουμε ένα ιδεατό ορθογώνιο γύρω από το κελί μας για να έχουμε πρόσβαση στα γειτονικά κελιά.
- Με μια εμφωλευμένη for επανάληψη περνάμε από όλα τα στοιχεία του χάρτη(x,y). Αρχικά γίνεται έλεγχος για το αν τα γειτονικά στοιχεία βρίσκονται μέσα στα όρια του χάρτη μας. Σε περίπτωση που δεν βρίσκονται τότε τα θεωρούμε ζωντανά, αλλιώς προσθέτουμε την

αντίστοιχη τιμή του γειτονικού στοιχείου στο `neighb`. Έστερα αφού συγκρίνουμε με το `death limit` αν είναι ζωντανό το κελί ή το `birth limit` αν είναι νεκρό, ενημερώνουμε αναλόγως τον νέο χάρτη.

Αφού επιστρέψει τον καινούργιο χάρτη η μέθοδος, γίνεται γέμισμα με το αντίστοιχο `tileset` ανάλογα αν είναι ζωντανό ή νεκρό το κελί.

Έστερα γίνεται η κλήση των τριών μεθόδων `ItemSpawner`, `MonsterSpawner` και `PlayerSpawner` που παράγουν τυχαίες θέσεις στον χάρτη για τα αντικείμενα, τους αντιπάλους και τον παίκτη μας. Για τα αντικείμενα και τα τέρατα η λειτουργία είναι ίδια, παίρνουμε ένα τυχαίο σημείο στον χάρτη και ελέγχουμε εάν είναι νεκρό (δηλαδή αν είναι πάτωμα και όχι τοίχος στην περίπτωση μας). Στην περίπτωση που είναι νεκρό διαλέγουμε τυχαία ένα από τα `GameObjects` που έχουμε ορίσει στον `editor` (`MonstersToSpawn`, `ItemsToSpawn`) και τα δημιουργούμε ως αντικείμενα στον χάρτη. Προσθέτουμε το αντικείμενο που παρήγαμε στον αντίστοιχο πίνακα (`MonsterSpawned`, `ItemSpawned`) και μειώνουμε κατά ένα την μεταβλητή που ορίζει πόσα αντικείμενα/τέρατα θα υπάρχουν στον χάρτη μας (`numOfSpawn`, `numOfItems`). Σε περίπτωση που δεν είναι νεκρό ξαναπαίρνουμε τυχαίο σημείο στον χάρτη. Αυτό επαναλαμβάνεται μέχρι ο αριθμός των αντικειμένων να φτάσει σε αυτό που έχουμε ορίσει.

Όσον αφορά τον παίκτη η διαδικασία είναι πιο απλή, παίρνουμε ένα τυχαίο σημείο στον χάρτη μέχρι να βρούμε κάποιο που είναι νεκρό και μεταφέρουμε τον χαρακτήρα μας σε εκείνο το σημείο.

Οι τελευταίες μέθοδοι που υπάρχουν είναι το `Despawner` και το `clearMap` τα οποία καταστρέφουν ότα αντικείμενα έχουν δημιουργηθεί στην προσομοίωση που υπάρχει και καθαρίζει τα `tiles` ή/και το `terrain map`, αντιστοίχως.

4.2.2 Κίνηση χαρακτήρα

Η κίνηση του χαρακτήρα γίνεται με την χρήση του συστήματος του Unity. Απαραίτητες προϋποθέσεις είναι το `sprite` του χαρακτήρα να έχει πάνω του ένα στοιχείο `Collider` που είναι αυτό που θα αλληλοεπιδρά με τα άλλα υπαρκτά στοιχεία στον χάρτη και ένα στοιχείο `Rigidbody` το οποίο είναι αυτό που θα εφαρμόσουμε την φυσική επάνω του. Οποιοδήποτε αντικείμενο έχει πάνω του `Rigidbody` στοιχείο σταματά να είναι μια απλή εικόνα και εφαρμόζονται πάνω του κανόνες φυσικής όπως για παράδειγμα βαρύτητα ή τριβή με αυτό που ορίζουμε ως έδαφος. Στην δική μας περίπτωση ορίζουμε και μια μεταβλητή τύπου `enum` με όλες τις καταστάσεις που μπορεί να είναι ο χαρακτήρας μας (`idle`, `κινούμενος`, `σε μάχη` κλπ.)

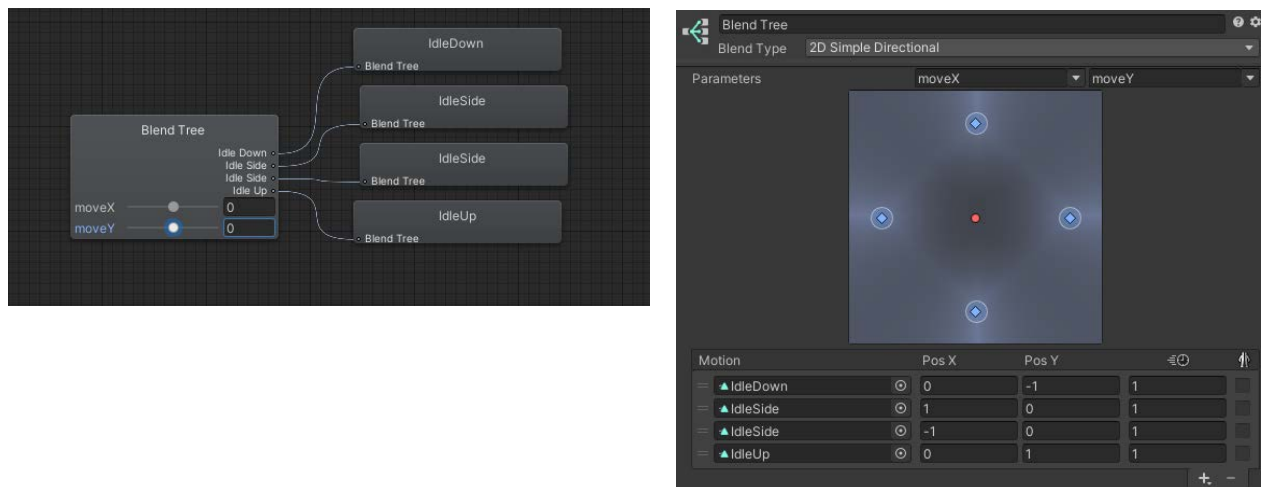
Χρησιμοποιώντας τα παραπάνω, για την κίνηση του χαρακτήρα στο παιχνίδι μας, δημιουργήθηκε ένα `script` το οποίο εφαρμόζεται πάνω στο στοιχείο του χαρακτήρα μας και λειτουργεί ως εξής:

- Παίρνουμε την τιμή των αξόνων από τα `Input` (Αριστερό/Δεξί βελάκι για `Horizontal Axis` και Πάνω/Κάτω βελάκι για `Vertical Axis`) σε κάθε `frame` και τα χρησιμοποιούμε για την δημιουργία μιας `Vector3` μεταβλητής
- Αν η τιμή της μεταβλητής στον άξονα `x` και `y` είναι 0, αλλάζουμε την κατάσταση του χαρακτήρα σε `Idle` μέσω του `enum` που έχουμε ορίσει και έστερα μηδενίζουμε το `velocity` του `rigid body`. Αυτό έχει ως αποτέλεσμα όταν δεν υπάρχει `input` κίνησης ο χαρακτήρας μας να σταματάει να κινείται.

- Αν η τιμή της μεταβλητής είναι διάφορη του 0 τότε καλούμε το script UpdateAnimationAndMove το οποίο αλλάζει την κατάσταση του χαρακτήρα μας καθώς και το animation.

4.2.3 Animation χαρακτήρα

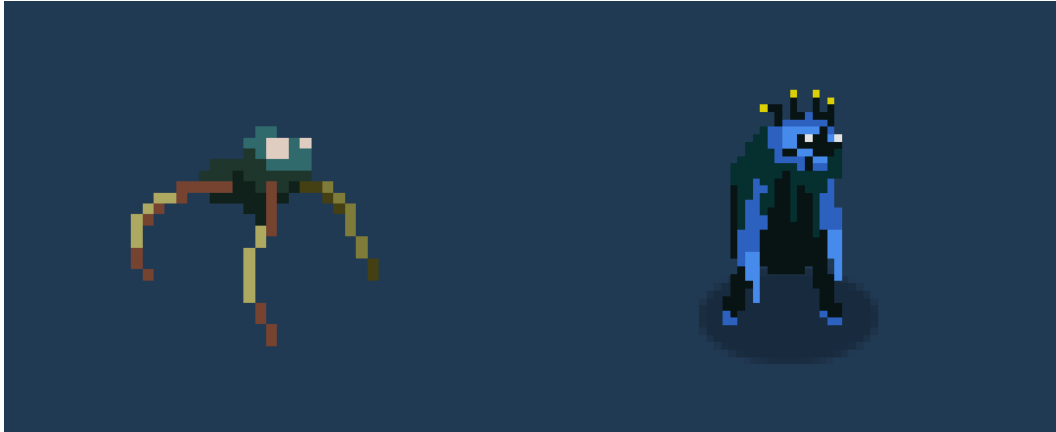
Το animation στην περίπτωση μας γίνεται με την χρήση blend tree. Έχουν δημιουργηθεί 3 διαφορετικά blend trees, ένα για την κίνηση, ένα για το idle, και ένα για την επίθεση του χαρακτήρα. Αυτό στο οποίο εξυπηρετεί το Blend Tree είναι η αυτόματη αναγνώριση της κατεύθυνσης βάσει των μεταβλητών που έχουμε (Σχήμα 46). Δίνοντας τις τιμές που έχουμε λάβει από τους άξονες, στις μεταβλητές που ορίζονται στον Animator του χαρακτήρα διαλέγει το ίδιο το Unity ποιο animation θα παίξει.



Σχήμα 46: Idle Blend Tree Rules

4.2.4 Αντίπαλοι

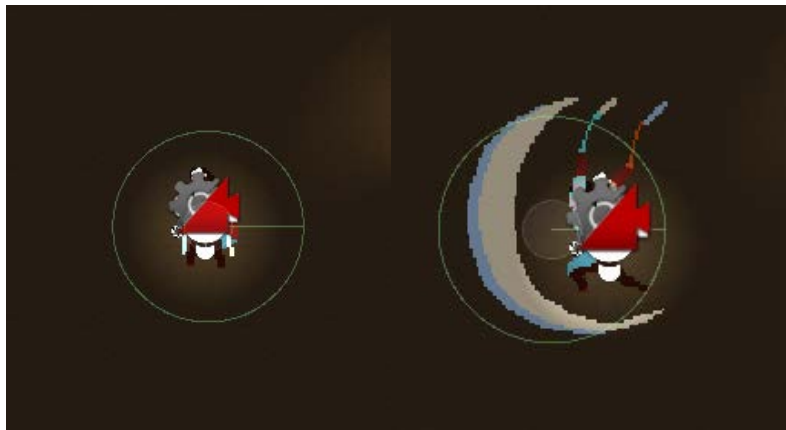
Οι αντίπαλοι στο παιχνίδι είναι 2 (Σχήμα 47) και είναι ένας διαφορετικός για κάθε σπηλιά εκτός από την τρίτη που υπάρχει συνδυασμός από τα δυο. Η ανίχνευση του χαρακτήρα μας είναι απλή και γίνεται απλά μέσω μιας ακτίνας γύρω από τον αντίπαλο στην οποία άμα βρεθεί ο χαρακτήρας μας αρχίζει να κινείται προς αυτόν. υπάρχει ένα μεταβλητό μέγεθος το οποίο ονομάζεται attack distance και χρησιμοποιείται για να δηλώσει σε ποια απόσταση σταματάει να κυνηγάει ο αντίπαλος και αρχίζει να επιτίθεται. Οι πληροφορίες αυτές είναι αποθηκευμένες πάνω σε ένα script το οποίο μπαίνει σε κάθε αντίπαλο και έχει πληροφορίες όπως η ζωή του αντιπάλου, η ταχύτητα κίνησης του, το πόση ζημιά ζωής κάνει ανά επίθεση στον χαρακτήρα μας καθώς και τις αποστάσεις στις οποίες ανιχνεύει και χτυπάει τον χαρακτήρα μας. Οι αντίπαλοι όπως και ο χαρακτήρας έχουν έναν Collider 2D για την ανίχνευση σύγκρουσης με τον χαρακτήρα και τους τοίχους. Αντίστοιχα έχουν πάνω τους και Blend Trees για κίνηση επίθεση και idle. Εάν η ζωή τους φτάσει στο 0 τότε παίζει ένα animation θανάτου και εξαφανίζονται μετά από λίγο.



Σχήμα 47: Αντίπαλοι

4.2.5 Σύστημα μάχης

Το σύστημα μάχης του παιχνιδιού χρησιμοποιεί colliders που ανοίγουν και κλείνουν ανάλογα την κατάσταση του χαρακτήρα (Σχήμα 48). Ο χαρακτήρας μπορεί να κάνει μια ή δυο επιθέσεις (combo). Αν δεν πατήσει σε επαρκή χρόνο το κουμπί της επίθεσης ο χαρακτήρας ξανακάνει την πρώτη κίνηση χωρίς να προχωρήσει στην δεύτερη. Με την επίθεση αρχίζει το animation της επίθεσης του χαρακτήρα και από πίσω ενεργοποιείται ο collider ο οποίος στρέφεται προς την μεριά που κοιτάει ο χαρακτήρας, εάν ανιχνεύσει ένα άλλο collider το οποίο έχει επάνω του το tag “Enemy” προχωράει στο να μειώσει την ζωή του αντιπάλου, αλλιώς επιστρέφει στην αρχική του θέση και απενεργοποιείται.



Σχήμα 48: Αριστερά, κλειστό collider κεντραρισμένο στον χαρακτήρα, Δεξιά αναμμένο collider με φορά προς την φορά επίθεσης

4.2.6 Γραφικά

Τα γραφικά εκτός από τον χαρακτήρα και τους αντιπάλους έχουν δημιουργηθεί από εμένα με την χρήση κυρίως του Aseprite. Για την δημιουργία των αυτόματων tiles που χρησιμοποιήθηκαν στις σπηλιές του παιχνιδιού χρησιμοποιήθηκε και το Tiled. Αναλύεται παρακάτω η διαδικασία δημιουργίας τους :

- Αφού δημιουργήσουμε το βασικό μας tileset στο Aseprite, δηλαδή το πως θέλουμε κάθε μεριά να είναι (κεντρικό, άκρες και γωνίες [Σχήμα 49]), το επιλέγουμε στο πρόγραμμα και το κάνουμε αντιγραφή



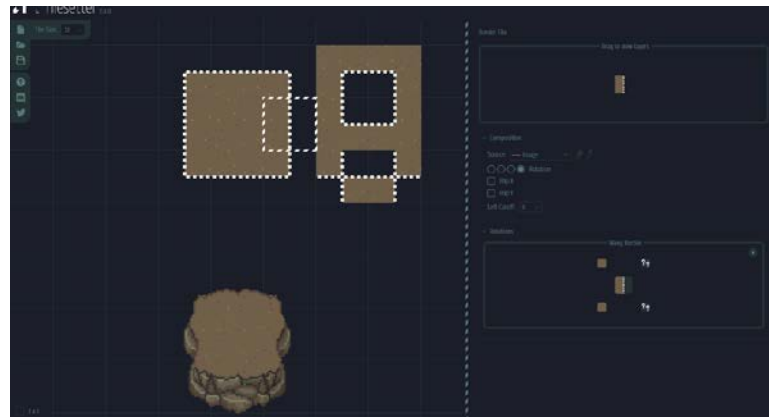
Σχήμα 49 Aseprite tiles

- Ανοίγοντας το Tiledseter κάνουμε την επιλογή New Project και ορίζουμε το μέγεθος του κάθε Tile σε pixel στην επιλογή Tile Size (στην δικιά μας περίπτωση 32px)
- Κάνουμε επικόλληση δυο φορές την επιλογή μας, μια φορά σε ένα σημείο και μια ακόμα πιο κάτω ή πάνω από το προηγούμενο.
- Διαλέγοντας το κεντρικό κομμάτι πατάμε δεξί κλικ και Build Borders(Wang)(Σχήμα 50).



Σχήμα 50 Tiledseter setup

- Από το αποτέλεσμα που μας έβγαλε παίρνουμε την κάθε πλευρά που έχει λευκές γραμμές (σχήμα 51), και αφού την επιλέξουμε, δεξιά στο composition κάνουμε την επιλογή source και διαλέγουμε την αντίστοιχη πλευρά από το σχήμα μας (Σχήμα 52).

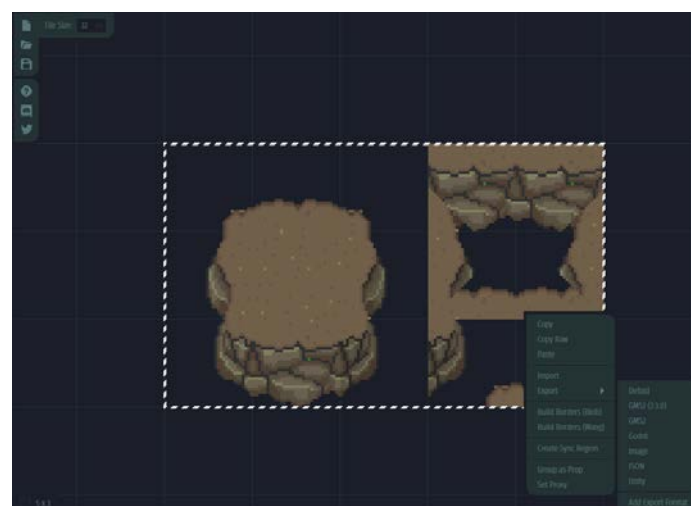


Σχήμα 51 Διαδικασία Tilesetter

- Αφού ολοκληρώσουμε την διαδικασία για κάθε πλευρά επιλέγουμε όλο το αποτέλεσμα και κάνοντας δεξί κλικ διαλέγουμε την επιλογή Export->Unity(Σχήμα 53)

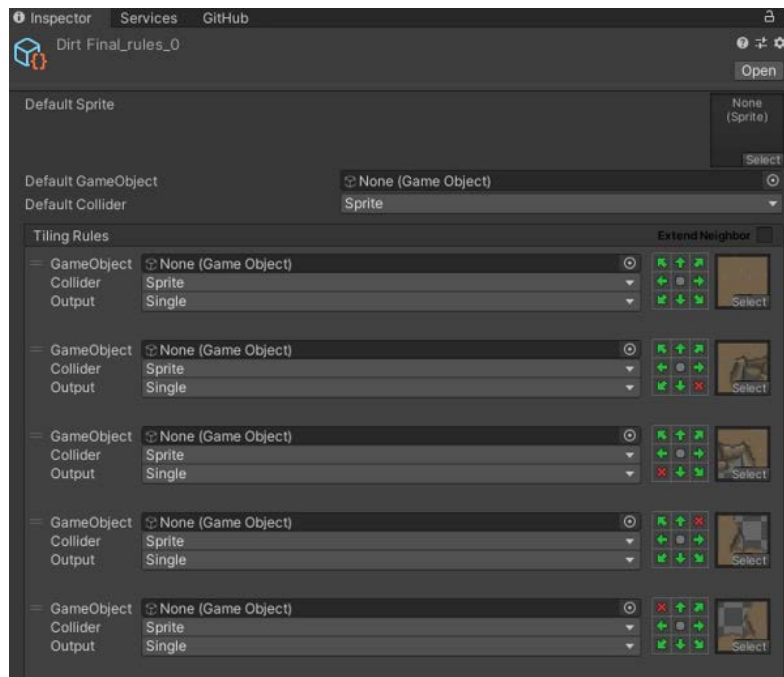


Σχήμα 52 Διαδικασία Tilesetter 2



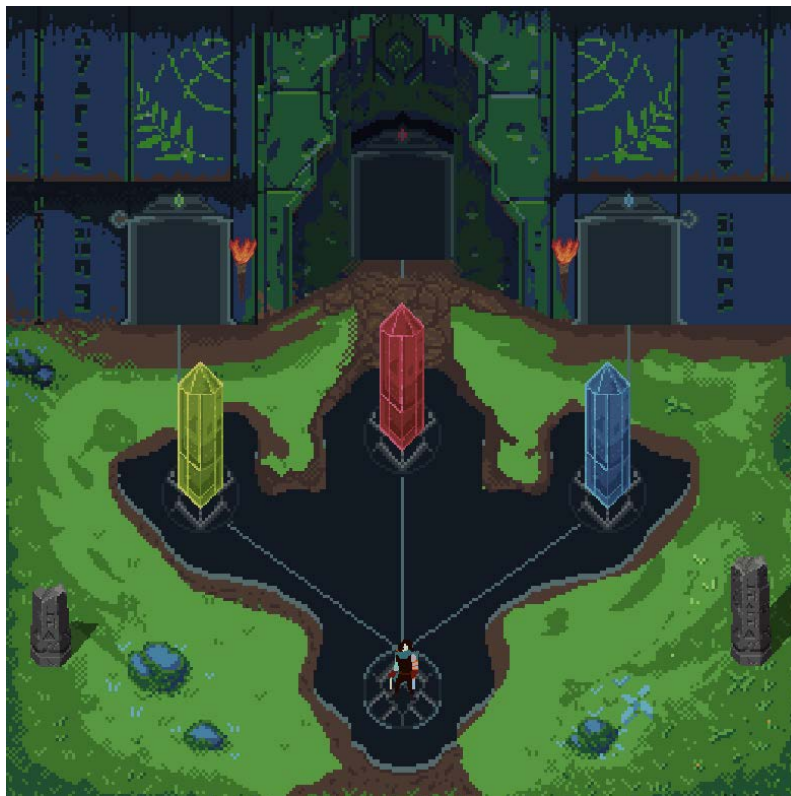
Σχήμα 53 Εξαγωγή σε Unity

- Ανοίγουμε το αρχείο που δημιουργήθηκε και γίνεται import των αρχιων στο Unity, καταλήγοντας έτσι με ένα αρχείο που έχει μέσα τους απαραίτητους κανόνες για την αυτόματη δημιουργία.(Σχήμα 54)



Σχήμα 54 Τελικοί αυτόματοι κανόνες Tiles

Το Unity επίσης προσφέρει μια πληθώρα από εφέ όπως φώτα, post processing, και συστήματα σωματιδίων (particle effects). Η χρήση τους γίνεται για να ενισχύσει την αισθητική του παιχνιδιού ακόμα και αν τα ίδια τα γραφικά δεν το πετυχαίνουν σε μεγάλο βαθμό. Στο παιχνίδι έγινε η χρήση και των τριών τεχνικών για να αποδοθούν τα γραφικά με όσο το δυνατόν καλύτερο τρόπο (Σχήμα 55,56)



Σχήμα 55 Πίστα χωρίς ενισχύσεις γραφικών



Σχήμα 56 Πίστα με ενισχύσεις γραφικών

Σημαντικό κομμάτι των ενισχύσεων είναι επίσης η χρήση των Normal Maps. Τα Normal Maps(σχήμα 57) είναι μια δευτερεύουσα εικόνα η οποία χρησιμοποιείται σαν μάσκα πάνω από την κανονική και έχει ως σκοπό να δώσει την αίσθηση του τρισδιάστατου όταν έρχεται σε επαφή με το φως. Για να δημιουργηθεί, δίνουμε χρώμα στην αρχική εικόνα που να αντιστοιχεί στην κατεύθυνση που θα χτυπάει το φως βάση μιας normal σφαίρας (Σχήμα 58). Αυτό μετά μεταφράζεται από το Unity και δημιουργεί το επιθυμητο εφέ(Σχήμα 59).

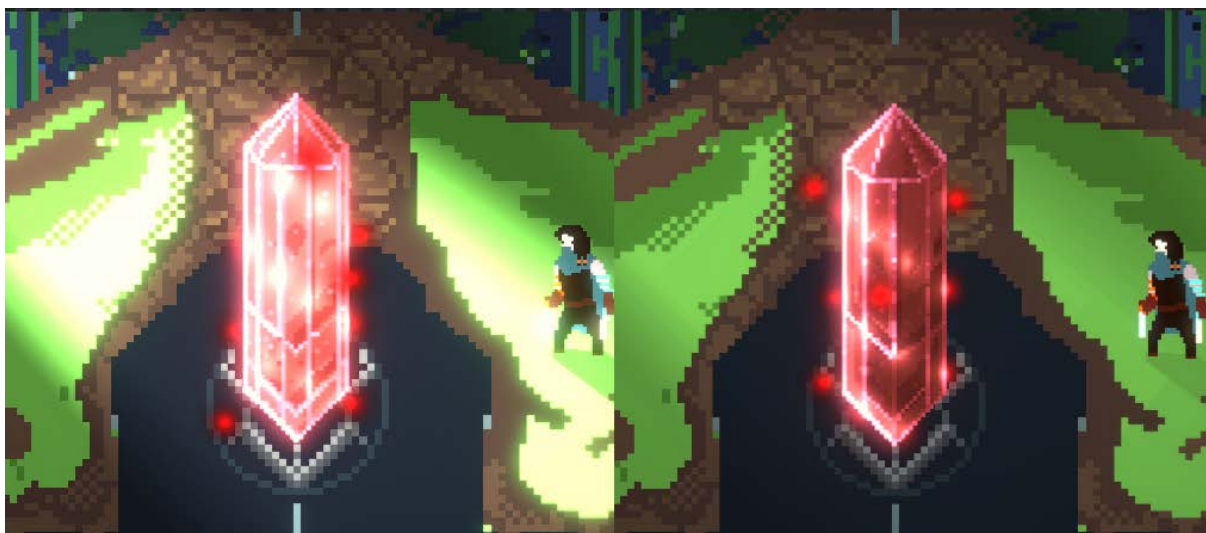


Σχήμα 57 Μια εικόνα μαζί με το αντίστοιχο normal map της



Σχήμα 58 Normal Sphere

Για την χρήση τους στο Unity θα πρέπει να γίνει η αντίστοιχη επιλογή(Use Normal Maps) στο κάθε φως οποίο υπάρχει στο παιχνίδι.



Σχήμα 59 Χωρίς χρήση Normal Maps (Αριστερά), Με χρήση normal maps (Δεξιά)

6 Συμπεράσματα και προτάσεις για βελτίωση

Η δημιουργία ενός βιντεοπαιχνιδιού είναι μια περίπλοκη διαδικασία, αλλά με την χρήση των κατάλληλων εργαλείων μπορεί να μειωθεί ο χρόνος δημιουργίας καθώς και η πολυπλοκότητα. Επίσης παρατηρούμε ότι με την χρήση εξωτερικών προγραμμάτων αλλά και ιδεών (Χρήση λογικής κυτταρικού αυτομάτου για την δημιουργία τυχαίων πιστών) η διαδικασία ανάπτυξης γίνεται πιο εύκολη, αλλά ταυτόχρονα προσθέτουμε πολυπλοκότητα στο παιχνίδι χωρίς να επιβραδύνει την διαδικασία δημιουργίας του.

Στην κατεύθυνση βελτίωσης του παιχνιδιού, θα μπορούσε να μεγαλώσει πολύ το παιχνίδι εκμεταλλευόμενοι άλλες τεχνικές για δημιουργία τυχαίων χαρτών (όπως για παράδειγμα παραγωγή μέσω Perlin Noise) κάνοντας έτσι πολύ πιο περίπλοκες δομές με την χρήση των τεχνικών που παρουσιάσαμε στην εργασία (Δημιουργία γραφικών στο Aseprite, και χρήση του Tiled για την εξαγωγή των επεξεργασμένων γραφικών στο Unity). Το κομμάτι κώδικα που αφορά την δημιουργία των χαρτών, μπορεί να χρησιμοποιηθεί αυτόνομο από άλλους προγραμματιστές αλλάζοντας μόνο τις παραμέτρους στον editor. Μπορεί επίσης βελτιωθεί η λειτουργικότητα με ορισμένες αλλαγές στον κώδικα για να εξυπηρετήσει διάφορους σκοπούς όπως παιχνίδι πολλαπλών χρηστών ή αύξηση χρήσης τεχνητής νοημοσύνης για τους αντιπάλους.

Μια ακόμα βελτίωση είναι η προσθήκη περαιτέρω συστημάτων όπως σύστημα αποθηκευτικού χώρου για αντικείμενα, σύστημα διαλόγων, καθώς και σύστημα πρόοδου χαρακτήρα.

7 Βιβλιογραφία

- [1] Lowood, «Pong: The First Commercially Successful Video Game: History of Information,» [Ηλεκτρονικό]. Available: <https://historyofinformation.com/detail.php?id=891>.
- [2] Codex Gamicus, «1970s in video gaming: Video game crash of 1977,» [Ηλεκτρονικό]. Available: https://gamicus.fandom.com/wiki/1970s_in_video_gaming#Video_game_crash_of_1977.
- [3] Vince, «Ultimate List of different types of videogames,» [Ηλεκτρονικό]. Available: <https://www.idtech.com/blog/different-types-of-video-game-genres>.
- [4] Anurag, «Pros and Cons of using Unity 3D,» [Ηλεκτρονικό]. Available: <https://www.newgenapps.com/blogs/pros-cons-of-using-unity3d-2/>.
- [5] Unity, «Unity Manual: Animation System Overview,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/AnimationOverview.html>.
- [6] Unity, «Unity - Manual: Sprite Editor,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/SpriteEditor.html>.
- [7] Unity, «Animation Rigging | 0.2.7-preview,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Packages/com.unity.animation.rigging@0.2/manual/index.html>.
- [8] Unity, «Unity-Manual: Animation Parameters,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/AnimationParameters.html>.
- [9] Unity, «Unity - Manual: Blend Trees,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/class-BlendTree.html>.
- [10] Unity, «Unity - Manual: Tilemap,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/class-Tilemap.html>.
- [11] Unity, «Unity - Manual: Grid,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/class-Grid.html>.
- [12] Unity, «Unity - Manual: Painting on Tilemaps,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/Tilemap-Painting.html>.
- [13] Unity, «Scriptable Brushes | 2D Tilemap Extras | 1.6.0-preview.1,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Packages/com.unity.2d.tilemap.extras@1.6/manual/Brushes.html>.
- [14] Unity, «Using Rule Tiles- Unity Learn,» [Ηλεκτρονικό]. Available: <https://learn.unity.com/tutorial/using-rule-tiles>.
- [15] Unity, «Animated Tile | 2D Tilemap Extras | 1.6.0-preview.1,» [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Packages/com.unity.2d.tilemap.extras@1.6/manual/AnimatedTile.html>.
- [16] «AlternativeTo,» 29 05 2021. [Ηλεκτρονικό]. Available:

<https://alternativeto.net/software/tilesetter/?p=3>.

- [17] «Tiles and Sources --- Tilesetter documentation,» 2020. [Ηλεκτρονικό]. Available: https://www.tilesetter.org/docs/tiles_and_sources. [Πρόσβαση 06 01 2022].
- [18] Tilesetter.org, «Working with Tiles — Tilesetter documentation,» 2020. [Ηλεκτρονικό]. Available: https://www.tilesetter.org/docs/working_with_tiles. [Πρόσβαση 06 01 2022].
- [19] Tilesetter.org, «Generating Tilesets — Tilesetter documentation,» 2020. [Ηλεκτρονικό]. Available: https://www.tilesetter.org/docs/generating_tilesets. [Πρόσβαση 06 01 2022].
- [20] Tilesetter.org, «Tileset Behavior — Tilesetter documentation,» 2020. [Ηλεκτρονικό]. Available: https://www.tilesetter.org/docs/tileset_behavior. [Πρόσβαση 06 01 2022].
- [21] Tilesetter.org, «Map Editing — Tilesetter documentation,» 2020. [Ηλεκτρονικό]. Available: https://www.tilesetter.org/docs/map_editing. [Πρόσβαση 06 01 2022].
- [22] Tilesetter.org, «Exporting — Tilesetter documentation,» 2020. [Ηλεκτρονικό]. Available: <https://www.tilesetter.org/docs/exporting>. [Πρόσβαση 07 01 2022].
- [23] E. Tzafestas, «Κυτταρικά αυτόματα και το παιχνίδι της ζωής,» [Ηλεκτρονικό]. Available: <http://users.softlab.ntua.gr/~brensham/Demos/CellularAutomata/>.