



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΥΛΙΚΩΝ

Συναισθηματική Αλληλεπίδραση σε Εικονικά Περιβάλλοντα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασιλική Ε. Θεοδωρέλη

Επιβλέπων : Στέφανος Δ. Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΥΛΙΚΩΝ

Συναισθηματική Αλληλεπίδραση σε Εικονικά Περιβάλλοντα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασιλική Ε. Θεοδωρέλη

Επιβλέπων : Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12^η Νοεμβρίου 2007.

.....
Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

.....
Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Παναγιώτης Τσανάκας
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2007

.....
Βασιλική Ε. Θεοδωρέλη

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Βασιλική Ε. Θεοδωρέλη, 2007

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στα πλαίσια της διπλωματικής αυτής εργασίας εξετάστηκαν τα χαρακτηριστικά που πρέπει να έχουν οι εικονικοί χαρακτήρες που χειρίζεται ο υπολογιστής και με τους οποίους αλληλεπιδρούν οι άνθρωποι στα παιχνίδια αλλά και σε άλλες διαδραστικές εφαρμογές, ώστε να παρουσιάζουν όσο το δυνατόν πιο ανθρώπινη συμπεριφορά, όχι τόσο στα πλαίσια της νοημοσύνης τους όσο στα πλαίσια της πειστικότητας των αντιδράσεών τους.

Η έκφραση συναισθημάτων από τους εικονικούς χαρακτήρες, η επιρροή της προσωπικότητας και της διάθεσής τους στην συμπεριφορά τους και η ικανότητά τους να αντλούν πληροφορίες από τις συνέπειες των πράξεών τους και κατά συνέπεια να τις αξιολογούν και να μην επαναλαμβάνουν τα ίδια λάθη, είναι χαρακτηριστικά απαραίτητα για την πειστικότητά τους.

Η μοντελοποίηση των συναισθημάτων είναι λοιπόν βασική και πρέπει να γίνει με τέτοιο τρόπο που να επιτρέπει την αλληλεπίδρασή τους καθώς και την έκφρασή τους με τέτοιο τρόπο που να μπορεί να γίνει αντιληπτός από τους ανθρώπους. Για τον σκοπό αυτό μελετάμε το μοντέλο OCC και αναφερόμαστε στις προσθήκες και τις αλλαγές που πρέπει να γίνουν για να εξυπηρετεί αποτελεσματικότερα του στόχους της σχεδίασης.

Παράλληλα με την θεωρητική ανάλυση του ζητήματος υλοποιήθηκε και ένα εικονικό περιβάλλον στο οποίο μπορούμε να παρατηρήσουμε στην πράξη μερικά από τα επιθυμητά χαρακτηριστικά των εικονικών χαρακτήρων. Πρόκειται για ένα περιβάλλον που αναπτύχθηκε με την μηχανή παραγωγής γραφικών Torque Game Engine, και στο οποίο ο παίκτης βρίσκεται μαζί με δύο εικονικούς χαρακτήρες οι οποίοι έχουν διαφορετική προδιάθεση απέναντι του και έτσι διαφοροποιούνται ως προς την συμπεριφορά τους. Επίσης η διάθεση των παικτών επηρεάζει τον τρόπο με τον οποίο κινούνται στον χώρο και ακόμη ο τρόπος με τον οποίο παίζει ο χρήστης επηρεάζει την συμπεριφορά των εικονικών χαρακτήρων.

Τέλος, μιας και η εφαρμογή έχει υλοποιηθεί με την αρχιτεκτονική πελάτη-εξυπηρετητή, έχουμε επιτρέψει στον διαχειριστή του εξυπηρετητή να μπορεί να παρακολουθεί τους πελάτες που είναι συνδεδεμένοι σε αυτόν και επίσης να μπορεί να επηρεάσει τον τρόπο με τον οποίο κινείται ο παίκτης στον κόσμο, προκαλώντας έτσι απροσδόκητη συμπεριφορά στην πλευρά του πελάτη, με στόχο να εξεταστούν τα συναισθήματα που του δημιουργούνται.

Λέξεις κλειδιά

Εικονικοί χαρακτήρες, εικονικός κοσμος, αλληλεπίδραση, συναισθηματική κατάσταση, προσωπικότητα, διάθεση, μοντέλο OCC, Torque Game Engine, προγραμματισμός για παιχνίδια, προσαρμοστική συμπεριφορά, αρχιτεκτονική πελάτη-εξυπηρετητή.

Abstract

In this diploma thesis we look into the characteristics that a virtual character that interacts with humans should have in order to be realistic and to behave and react in a manner similar to that of real people.

The ability to express emotions, the influence of their personality and mood in their behavior, and the capacity evaluate an action from its consequences are critical characteristics. The Non Player Characters are thus expected not only to show intelligence but also to behave in a way compatible with the environment they act in as well as their previous actions, in order to be convincing.

The modeling of emotions is fundamental and needs to be done in a way that permits first, the interactions of the various emotional states and second the expression of the emotions in a way understandable by the human player. So, we study the OCC model of emotions and we refer to the changes and additions that need to be done so that the model addresses in an even more accurate way to the design goals.

Alongside with the theoretical analysis we implemented a virtual environment, using the Torque Game Engine, in order to observe in reality some of the characteristics mentioned above. The player in the “Moods” environment co-exists with two Non Player Characters that adopt their behavior to the way he plays. We can observe their different actions given that they have different goals and they are biased in a different way towards the player. In addition the mood of the players is designed to influence the way the move around the world.

Finally, given that the project is implemented in the client-server architecture we have enabled the server’s manager to observe the clients connected to the server and to change in an unpredicted manner, the movement of the player’s avatar, in order to induce emotions to the player.

Key words

Virtual Characters, virtual world, interaction, emotional state, personality, mood, OCC model, Torque Game Engine, game programming, adaptive behavior, client-server architecture.

Πίνακας Περιεχομένων

1	<u>ΕΙΣΑΓΩΓΗ</u>	9
2	<u>ΣΧΕΔΙΑΣΜΟΣ ΡΕΑΛΙΣΤΙΚΩΝ NON PLAYER CHARACTERS</u>	11
2.1	ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΝΑΙΣΘΗΜΑΤΩΝ ΓΙΑ ΔΙΑΔΡΑΣΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ	12
2.1.1	ΤΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΝΟΣ ΔΙΑΔΡΑΣΤΙΚΟΥ ΜΟΝΤΕΛΟΥ ΣΥΝΑΙΣΘΗΜΑΤΩΝ	12
2.1.2	ΤΟ ΜΟΝΤΕΛΟ OCC	13
2.1.3	ΕΛΛΕΙΨΕΙΣ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ OCC ΜΟΝΤΕΛΟΥ ΚΑΙ ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ	18
2.2	ΣΤΟΧΟΙ ΚΑΤΑ ΤΟΝ ΣΧΕΔΙΑΣΜΟ ΕΝΟΣ ΔΙΑΔΡΑΣΤΙΚΟΥ NON PLAYER CHARACTER	21
2.2.1	ΕΙΚΟΝΙΚΗ ΠΡΟΣΩΠΙΚΟΤΗΤΑ	21
2.2.2	ΔΥΝΑΜΙΚΗ ΠΡΟΣΑΡΜΟΓΗ ΣΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΧΡΗΣΤΗ	24
2.2.3	ΕΠΙΡΡΟΗ ΤΗΣ ΔΙΑΘΕΣΗΣ	25
2.3	ΠΑΡΑΔΕΙΓΜΑ ΣΥΣΤΗΜΑΤΟΣ ΜΕ NPC'S ΠΟΥ ΣΥΝΟΜΙΛΟΥΝ	25
2.4	ΑΠΛΟΠΟΙΗΜΕΝΟ ΜΟΝΤΕΛΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	28
3	<u>Η ΜΗΧΑΝΗ TORQUE GAME ENGINE</u>	29
3.1	Η ΙΣΤΟΡΙΑ ΤΗΣ TORQUE GAME ENGINE	29
3.2	ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	29
3.2.1	ΥΠΟΣΤΗΡΙΞΗ ΠΟΛΛΩΝ ΠΑΙΚΤΩΝ – ΔΥΝΑΤΟΤΗΤΑ ΔΙΚΤΥΩΣΗΣ	29
3.2.2	ΠΑΡΑΓΩΓΗ ΓΡΑΦΙΚΩΝ	31
3.2.3	ΚΟΝΣΟΛΑ ΚΑΙ TORQUESCRIPT	34
3.2.4	ΠΡΟΣΟΜΟΙΩΣΗ ΓΕΓΟΝΟΤΩΝ	36
3.2.5	DATABLOCKS	37
3.3	ΣΥΜΠΛΗΡΩΜΑΤΙΚΑ ΕΡΓΑΛΕΙΑ	38
3.3.1	SHOW TOOL PRO	38
3.3.2	ONLINE ΥΠΟΣΤΗΡΙΞΗ – FORUM.	40
4	<u>ΤΟ ΠΕΡΙΒΑΛΛΟΝ MOODS</u>	41
4.1	ΤΟ ΣΕΝΑΡΙΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	41
4.2	ΘΕΩΡΗΤΙΚΑ ΣΤΟΙΧΕΙΑ ΠΟΥ ΕΝΣΩΜΑΤΩΘΗΚΑΝ ΣΤΗΝ ΕΦΑΡΜΟΓΗ	42
4.3	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ	45
4.3.1	ΠΕΡΙΒΑΛΛΟΝ CLIENT	45
4.3.2	ΠΕΡΙΒΑΛΛΟΝ SERVER	48
4.4	ΠΕΡΙΓΡΑΦΗ GAMEPLAY	49
5	<u>ΕΠΕΚΤΑΣΕΙΣ ΚΑΙ ΝΕΟΙ ΣΤΟΧΟΙ</u>	65
5.1	ΒΕΛΤΙΩΣΗ ΑΝΤΙΛΗΨΗΣ ΤΩΝ NON PLAYER CHARACTERS	65
5.2	ΧΡΗΣΗ ΤΟΥ ΑΠΟΜΑΚΡΥΝΣΜΕΝΟΥ ΕΛΕΓΧΟΥ ΓΙΑ ΕΝΑ TURING TEST	66
6	<u>ΓΛΩΣΣΑΡΙ</u>	67
7	<u>ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ</u>	68

Πίνακας Σχημάτων

- Σχήμα 1.** Το μοντέλο του μοντέλου OCC
- Σχήμα 2.** Ο ρόλος της προσωπικότητας και της συναισθηματικής κατάστασης στα πλαίσια ενός ευφυούς πράκτορα.
- Σχήμα 3.** Η συμπεριφορά της Jill σαν (α) εξωστρεφής και (β) εσωστρεφής προσωπικότητα.
- Σχήμα 4.** Σύστημα παραγωγής αντιδράσεων των χαρακτήρων
- Σχήμα 5.** Διαφορετικές αρχιτεκτονικές εφαρμογών με το μοντέλο πελάτη-εξυπηρετητή.
- Σχήμα 6.** Διαφορετικοί τρόποι διασύνδεσης στην αρχιτεκτονική πελάτη-εξυπηρετητή
- Σχήμα 7.** Όψη του Mission Editor
- Σχήμα 8.** Όψη του Gui Editor
- Σχήμα 9.** Όψη του Show Tool Pro
- Σχήμα 10.** Η είσοδος του παίκτη στον περιβάλλον Moods
- Σχήμα 11.** Διάγραμμα καταστάσεων Φίλου-NPC
- Σχήμα 12.** Διάγραμμα καταστάσεων Εχθρού-NPC
- Σχήμα 13.** Το βασικό μενού της εφαρμογής
- Σχήμα 14.** Menu Αναζήτησης Εξυπηρετητών
- Σχήμα 15.** Προειδοποίηση εξόδου από την mission area
- Σχήμα 16.** Τα Huds και τα βασικά στοιχεία του κόσμου.
- Σχήμα 17.** Παράθυρο εξόδου από την εφαρμογή
- Σχήμα 18.** Γραφικό περιβάλλον εξυπηρετητή
- Σχήμα 19.** Διαθέσιμα animations για τον χαρακτήρα του Moods
- Σχήμα 20.** Τα τρία αντικείμενα της εφαρμογής
- Σχήμα 21.** Η αντιστοίχιση του συντελεστή διάθεσης στις τρεις κατηγορίες διάθεσης
- Σχήμα 22.** Η λειτουργία του Φίλου- NPC (a)
- Σχήμα 23.** Η λειτουργία του Φίλου- NPC (b)
- Σχήμα 24.** Η λειτουργία του Εχθρού NPC

1 Εισαγωγή

Η επικράτηση των Υπολογιστών σε όλες σχεδόν τις όψεις της καθημερινότητας του σύγχρονου ανθρώπου είναι ένα γεγονός που δύσκολα θα μπορούσε κανείς να αρνηθεί. Η διασκέδαση και ειδικότερα τα παιχνίδια είναι ένας τομέας στον οποίο η παρουσία των υπολογιστών είναι καταλυτικής σημασίας και μάλιστα, θα μπορούσαμε να πούμε ότι η πρόοδος και οι καινοτομίες της τεχνολογίας έχει καθορίσει σε μεγάλο βαθμό την εξέλιξή τους.

Η βιομηχανία της ανάπτυξης παιχνιδιών κατέχει μία θέση βαρύνουσας σημασίας καθώς τροφοδοτεί συνεχώς την παγκόσμια αγορά με νέα δημοφιλή προϊόντα, συντηρεί και δημιουργεί νέες θέσεις εργασίας και αποτελεί ένα σημαντικό ερευνητικό τομέα στον οποίο τα όποια ευρήματα βρίσκουν γρήγορα το αντίκρισμα τους στην πράξη. Δύσκολα μπορεί να φανταστεί κανείς μία καλύτερη ανταμοιβή για έναν ερευνητή απ' το να βλέπει το αποτέλεσμα της δουλειάς του να χρησιμοποιείται άμεσα στην παραγωγική διαδικασία και να επηρεάζει το καταναλωτικό κοινό κυρίως όσον αφορά τις απαιτήσεις του. Και φυσικά δεν είναι καθόλου ψέμα να υποστηρίξει κανείς ότι η βιομηχανία παιχνιδιών είναι ιδιαίτερα ανταγωνιστική, με τις διάφορες μεγάλες εταιρίες και τα αντίστοιχα προϊόντα τους να δίνουν μάχη για την κατάκτηση του νεανικού και όχι μόνο κοινού.

Τι κάνει όμως τα βιντεοπαιχνίδια τόσο δημοφιλή; Το κύριο χαρακτηριστικό τους είναι ότι προσφέρουν στον παίκτη έναν μοναδικό εικονικό κόσμο στον οποίο μπορεί να εκφραστεί, να εκτονωθεί, να αυτοεπιβεβαιωθεί ακόμη και να απογοητευτεί, έχοντας όμως πάντα τη δυνατότητα να ξαναπροσπαθήσει και να τα πάει καλύτερα.

Ο κόσμος αυτός συνοδεύεται συνήθως από στοιχεία που συχνά συναντάμε στις κινηματογραφικές ταινίες όπως εντυπωσιακά γραφικά και καλά δομημένη πλοκή και η κατασκευή του είναι αυτή που απαιτεί από την τεχνολογία να εξελίσσεται διαρκώς για να μπορέσει να υλοποιήσει και τις πιο δημιουργικές ιδέες των σχεδιαστών.

Η ανάπτυξη την τεχνολογίας των παιχνιδιών όμως συμπαρασύρει και τις απαιτήσεις του κοινού, το οποίο απαιτεί όλο και καλύτερη ποιότητα εικόνας και ήχου, όλο και πιο αληθοφανή σενάρια, όλο και πιο πειστική ανταπόκριση στις κινήσεις του παίκτη μέσα στο παιχνίδι. Αυτό που είναι σημαντικό να σημειωθεί είναι ότι σε ένα βιντεοπαιχνίδι δεν είναι μόνο ο παίκτης-άνθρωπος που καθορίζει την πορεία του. Ο κρυφός παίκτης-υπολογιστής είναι εξίσου σημαντικός δεδομένου ότι με την μειωμένη αντίληψη που διαθέτει είναι πλέον υποχρεωμένος να αναπαράγει έναν πιστευτό κόσμο, με ρεαλιστικά φαινόμενα και αντιδράσεις που να πλαισιώνει τον παίκτη-άνθρωπο.

Εδώ είναι και το σημείο στο οποίο συναντάμε στον παρόν τις μεγαλύτερες προκλήσεις όσον αφορά την τεχνολογία των παιχνιδιών. Στα περισσότερα παιχνίδια συναντάμε παίκτες οι οποίοι δεν ελέγχονται από κάποιον άνθρωπο αλλά παρ' όλα αυτά αποτελούν αναπόσπαστα κομμάτια του ηλεκτρονικού κόσμου του παιχνιδιού. Οι λεγόμενοι Non Player Characters (NPCs) είναι αυτοί που ανάλογα τον ρόλο που τους έχει δοθεί, πρέπει να βοηθήσουν (εάν το χρειάζεται) ή να εμποδίσουν (χωρίς να αποτρέψουν ολοκληρωτικά) τον παίκτη να καταφέρει τον όποιο σκοπό έχει θέσει μέσα στο παιχνίδι.

Δεδομένου ότι ο υπολογιστής δεν διαθέτει ακόμη επαρκή ευφυΐα στο να αναλύει τα διάφορα συμβάντα στον εικονικό κόσμο, και να προσαρμόζει ρεαλιστικά και άμεσα την απόκρισή του με έναν τρόπο που θα ήταν αντίστοιχος αυτού του εκάστοτε παίκτη από μόνος του, είναι ιδιαίτερης σημασίας να δημιουργηθεί ένα όσο

το δυνατόν καλύτερο μοντέλο για τους χαρακτήρες αυτούς ώστε να διατηρείται η επιθυμητή ισορροπία μέσα στον παιχνίδι. Είναι σαφές άλλωστε ότι κανείς δεν θα έβρισκε ενδιαφέρον σε ένα υπερβολικά εύκολο παιχνίδι, ή αντίστοιχα σε ένα πολύ δύσκολο αλλά ούτε και σε κάποιο προβλέψιμο.

Το ζητούμενο λοιπόν είναι να δημιουργηθούν οι κατάλληλες συνθήκες ώστε το παιχνίδι να παρουσιάζει έναν δυναμικό χαρακτήρα που θα επιτρέπει στον παίκτη να βρίσκει ενδιαφέρον σε αυτό, κρατώντας φυσικά χαμηλά την υπολογιστική ισχύ και τις απαιτήσεις μνήμης.

Παρακάτω θα εξετάσουμε πιο συγκεκριμένα ποια είναι τα επιθυμητά χαρακτηριστικά ενός NPC και ποιες νέες ανάγκες προκύπτουν ώστε να παραμένουν πειστικοί και αποτελεσματικοί στον εκάστοτε ρόλο τους.

2 Σχεδιασμός ρεαλιστικών Non Player Characters

Σε πολλά παιχνίδια στρατηγικής ή ρόλων, οι χαρακτήρες που δεν ελέγχονται από τον χρήστη κατέχουν μία πολύ σημαντική θέση στην εξέλιξη του παιχνιδιού. Είναι αυτοί που κάνουν τον εικονικό κόσμο πιο πιστευτό στον παίκτη καθώς αντιπροσωπεύουν την ανθρώπινη παρουσία η οποία είναι άκρως απαραίτητη για να μπορέσει ο παίκτης να βιώσει όλα τα συναισθήματα που μπορεί να του προσφέρει η εμπειρία του παιχνιδιού.

Η χρησιμότητα τους ωστόσο κρύβει μια παγίδα δεδομένου ότι ο παίκτης έχει, συχνά ασυνείδητα, την απαίτηση από αυτούς τους χαρακτήρες να συμπεριφέρονται με τρόπο ανθρώπινο. Σύμφωνα με μελέτες που έχουν γίνει [16], οι άνθρωποι παίκτες έχουν την τάση να αποδέχονται ακόμα και παίκτες αρχάριους ή με λιγότερες ικανότητες απ' ό,τι οι ίδιοι εάν γνωρίζουν ότι πρόκειται για παίκτες που ελέγχονται από κάποιον άνθρωπο. Αυτό έχει να κάνει με το γεγονός ότι η ικανοποίηση του νικητή πολλές φορές πηγάζει από το γεγονός ότι ο ηττημένος αισθάνεται την υπεροχή του, και του προκαλούνται αντίθετα συναισθήματα. Σίγουρα ένας αντίπαλος που δεν μπορεί να νιώσει την «ντροπή μιας ήττας» είναι ενδεχομένως ένας αντίπαλος χωρίς ενδιαφέρον. Ο στόχος μας λοιπόν είναι να δώσουμε στους NPCs έναν όσο το δυνατόν πιο ανθρώπινο χαρακτήρα.

Αυτό σημαίνει ότι ο υπολογιστής καλείται να προσομοιώσει στην συμπεριφορά των NPCs στοιχεία όπως η αντίληψη του γύρω χώρου και των αντικειμένων και των προσώπων που κινούνται σε αυτόν, η δράση αλλά και η αντίδραση και τέλος η μάθηση τόσο από τα γεγονότα που λαμβάνουν χώρα όσο και από τον αντίκτυπο που έχουν οι διάφορες αποφάσεις τους. Πρέπει επίσης να μπορούν να χρησιμοποιούν την γλώσσα και να συμμετέχουν σε συνομιλίες, να εξάγουν συμπεράσματα με βάση τη λογική και να έχουν μία γενικότερη γνώση του κόσμου γύρω τους καθώς και της προσωπικής τους ιστορίας. Σε έναν ιδανικό κόσμο οι NPCs πρέπει να είναι ικανοί να μαθαίνουν από τα λάθη τους και να διαφοροποιούν τις αντιδράσεις τους ανάλογα με τις συνθήκες στις οποίες βρίσκονται και ανάλογα με τον χαρακτήρα του καθενός. Σίγουρα όπως και διαφορετικοί άνθρωποι δεν αντιδρούν με τον ίδιο τρόπο στα ίδια ερεθίσματα, το ίδιο περιμένει να δει κανείς και από διαφορετικούς NPCs.

Είναι σαφές λοιπόν ότι η εγκυρότητα ενός NPC στον εκάστοτε εικονικό κόσμο δεν έχει τόσο να κάνει τόσο με την νοημοσύνη του όσο με την ικανότητά του να αντιδράει με ανθρώπινο τρόπο στα διάφορα γεγονότα. Δεδομένου όμως ότι οι εικονικοί χαρακτήρες στερούνται σώματος, είναι προφανώς δύσκολο να μπορέσουν να ενσωματώσουν όλες αυτές τις βιολογικές αντιδράσεις που συναντά κανείς σε έναν ζωντανό οργανισμό όπως η αύξηση της αδρεναλίνης σε στιγμές έντασης, η ταχυπαλμία σε περιπτώσεις φόβου κτλ. Γι' αυτό λοιπόν πρέπει να χρησιμοποιηθούν μοντέλα που κόσμου, των συναισθημάτων και όλων γενικά των πραγμάτων που καθορίζουν την συμπεριφορά ενός χαρακτήρα, ώστε τελικά οι αντιδράσεις του να πλησιάζουν όσο το δυνατόν τις ανθρώπινες.

Η τάση που τείνει να επικρατήσει είναι η μοντελοποίηση των ανθρώπινων συναισθημάτων και η προσπάθεια καθορισμού της συμπεριφοράς των NPCs μέσω της συναισθηματικής τους κατάστασης, καθώς οι περισσότεροι ερευνητές τεχνητής νοημοσύνης αποδέχονται ότι ο δρόμος για τους πραγματικά ευφυείς υπολογιστές θα περάσει μέσα από το χώρο των ανθρωπίνων συναισθημάτων.

Είναι χαρακτηριστικός ο τρόπος που περιγράφεται η ανάγκη για την ανάπτυξη συναισθηματικών υπολογιστών από την ομάδα Affective Computing του MIT:

“Η σημασία αυτού προκύπτει από την δουλειά του Damasio καθώς και άλλων που έχουν μελετήσει ασθενείς που στερούνται βασικών συναισθημάτων και κατά συνέπεια υποφέρουν από αδυναμία να πάρουν αποφάσεις με λογικό τρόπο. Η φύση της αδυναμίας τους αυτής είναι περίεργα παρόμοια με τον τρόπο που παίρνουν αποφάσεις οι Boolean – decision making μηχανές, και τα απλά ευφυή συστήματα τεχνητής νοημοσύνης. Πρόσφατα ευρήματα δείχνουν πλέον ότι στους ανθρώπους τα συναισθήματα είναι απαραίτητα για την λήψη αποφάσεων με λογικό και ευέλικτο τρόπο. Η υπόθεσή μας είναι ότι τα συναισθήματα θα είναι επίσης απαραίτητα για της μηχανές ώστε να παίρνουν αποφάσεις με τον ίδιο τρόπο, και ώστε να παρουσιάζουν έναν πραγματικά δημιουργικό τρόπο σκέψης και γενικότερα ένα πλήθος από γνωσιακές ικανότητες που μοιάζουν με αυτές του ανθρώπου.” [14]

Επίσης σημαντική θέση έχει και η προσπάθεια να συνδυαστεί η συναισθηματική κατάσταση των χαρακτήρων με μία γενική προδιάθεση όσων αφορά τις πράξεις τους, η οποία θα μπορούσε να παίξει το ρόλο του ανθρώπινου χαρακτήρα όπως αυτός πλάθεται μέσα από τα βιώματα και τις εμπειρίες του καθενός.

Φυσικά είναι μάλλον δύσκολο να μιλήσουμε για την εξέλιξη του χαρακτήρα ενός NPC στο μικρό χρονικό ορίζοντα ενός παιχνιδιού, οπότε μάλλον μιλάμε για προκαθορισμένες από πριν προδιαθέσεις, ωστόσο παραμένει ενδιαφέρον να παρακολουθεί κανείς διαφορετικούς χαρακτήρες να έρχονται αντιμέτωποι με τα ίδια ερεθίσματα και παρ’ όλα αυτά να αναπτύσσουν μια δυναμική συμπεριφορά η οποία θα διαφοροποιείται ανάλογα με τον «χαρακτήρα» του καθενός, όπως αυτός θα έχει καθοριστεί κατά την φάση του σχεδιασμού τους.

2.1 Μοντελοποίηση συναισθημάτων για διαδραστικές εφαρμογές

Η προσπάθεια να συμπεριληφθεί η συναισθηματική κατάσταση στα χαρακτηριστικά των NPCs σίγουρα ξεκινάει με την μοντελοποίηση των συναισθημάτων και των μεταξύ τους αλληλεπιδράσεων, καθώς και της επιρροής που ασκούν στην συλλογιστική διαδικασία και κατά συνέπεια στις πράξεις των ανθρώπων. Φυσικά έχουν προταθεί κατά καιρούς διάφορα μοντέλα αλλά στον συγκεκριμένο τομέα μας ενδιαφέρουν μόνο αυτά που παρέχουν τις κατάλληλες δομές ώστε να μπορούν να αντιμετωπίσουν αποτελεσματικά την αλληλεπίδραση μεταξύ των συναισθημάτων.

2.1.1 Το χαρακτηριστικά ενός διαδραστικού μοντέλου συναισθημάτων

Όπως αναφέραμε, τα συναισθήματα είναι ένα απαραίτητο στοιχείο για να είναι πιστευτός ένας χαρακτήρας κατά την επαφή του με τους ανθρώπους, και φυσικά οι χαρακτήρες χρειάζονται ένα μοντέλο ώστε να μπορέσουν να συνθέσουν συναισθήματα και να εκφράσουν.

Ένα ολοκληρωμένο μοντέλο πρέπει να επιτρέπει στον χαρακτήρα να επεξεργάζεται τα συναισθήματά του με τον ίδιο τρόπο που κάνουν οι άνθρωποι. Ένα γεγονός που για παράδειγμα θα αναστάτωνε έναν άνθρωπο θα πρέπει να έχει την ίδια επίδραση και στον εικονικό χαρακτήρα. Το μοντέλο πρέπει να μπορεί να εκτιμήσει

όλες τις καταστάσεις στις οποίες μπορεί να βρεθεί ένας χαρακτήρας, και ακόμη να περιλαμβάνει τις κατάλληλες δομές μεταβλητών που να επηρεάζουν την ένταση του κάθε συναισθήματος. Ένα τέτοιο μοντέλο επιτρέπει στον παίκτη να δείξει το σωστό συναίσθημα, στη σωστή στιγμή και κυρίως στην σωστή ένταση, κάτι το οποίο είναι απαραίτητο για την πειστικότητά του.

Από την παρουσίαση ενός γεγονότος και την κατηγοριοποίησή του, μέχρι την τελική συμπεριφορά του παίκτη, υπάρχει μία λογική διαδικασία που ακολουθείται και η οποία μπορεί να χωριστεί σε 5 διακεκριμένες φάσεις.

1. Κατηγοριοποίηση
Στην φάση κατηγοριοποίησης ο χαρακτήρας εκτιμά ένα γεγονός, μία ενέργεια ή ένα αντικείμενο και καταλήγει με πληροφορία σχετικά με το ποιες κατηγορίες συναισθημάτων θα επηρεαστούν.
2. Προσδιορισμός έντασης
Κατά την φάση αυτή ο χαρακτήρας υπολογίζει πόσο πολύ θα επηρεαστούν οι κατηγορίες στις οποίες κατέληξε κατά την προηγούμενη φάση της κατηγοριοποίησης.
3. Αλληλεπίδραση
Οι δύο προηγούμενες φάσεις καθορίζουν την συναισθηματική αξία ενός γεγονότος, μίας πράξης ή ενός αντικειμένου. Τώρα κατά την φάση της αλληλεπίδρασης η τιμή αυτή που προέκυψε πρέπει να συνδυαστεί με την τρέχουσα συναισθηματική κατάσταση του χαρακτήρα.
4. Αντιστοίχιση
Οι διάφορες (συνήθως πολυάριθμες) κατηγορίες συναισθημάτων που ορίζει το μοντέλο, πρέπει να αντιστοιχηθούν σε έναν μικρότερο αριθμό από εκφράσεις συναισθημάτων ώστε να καθοριστεί με σχετικά απλούστερο βαθμό η τελική συναισθηματική κατάσταση του χαρακτήρα.
5. Έκφραση
Η συναισθηματική κατάσταση που προκύπτει ως αποτέλεσμα των προηγούμενων φάσεων πρέπει τελικά να εκφραστεί π.χ. μέσω των διαθέσιμων εκφράσεων του προσώπου ή γενικά να επηρεάσει την συμπεριφορά του χαρακτήρα.

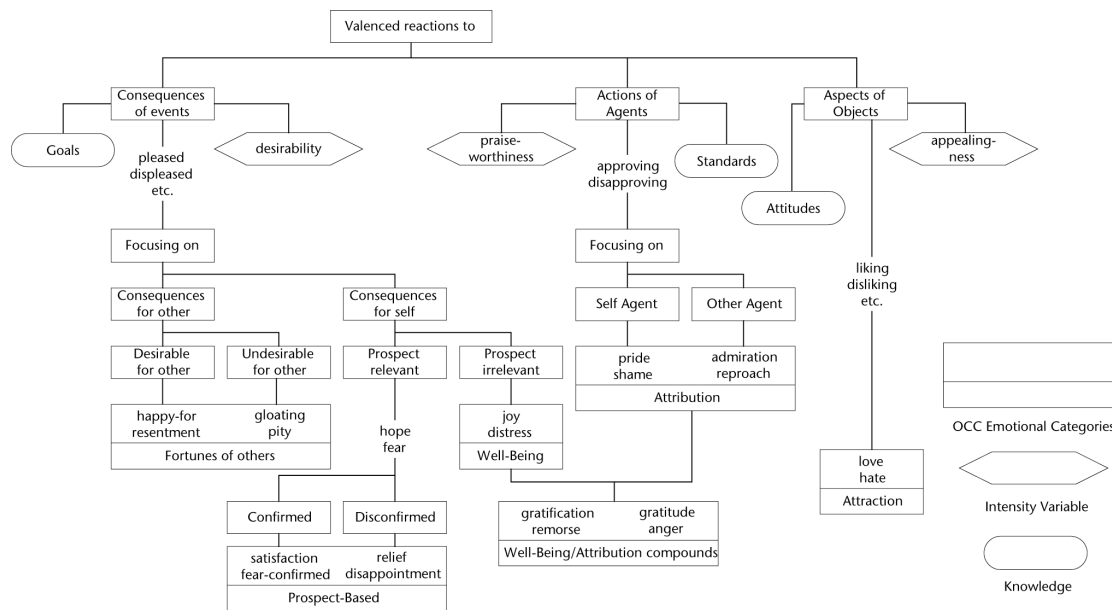
2.1.2 Το μοντέλο OCC

Η θεωρία των Ortony, Clore και Collins (OCC) βασίζεται στην υπόθεση ότι τα συναισθήματα αναπτύσσονται ως συνέπεια συγκεκριμένων συλλογισμών και ερμηνειών και επομένως μπορούμε να πούμε ότι επικεντρώνεται αποκλειστικά στις νοητικές αιτίες των συναισθημάτων. Σύμφωνα με τους συγγραφείς τρία πράγματα καθορίζουν αυτές τις αιτίες: τα γεγονότα (events), οι φορείς (agents), και τα αντικείμενα (objects) [2]

«Τα γεγονότα εξετάζονται από το πρίσμα των συνεπειών τους, οι φορείς από το πρίσμα των ενεργειών τους και τέλος τα αντικείμενα από το πρίσμα των ιδιοτήτων τους.» [11]

Τα συναισθήματα λοιπόν δεν είναι παρά οι αντιδράσεις σε αυτές τις όψεις του κόσμου. Για παράδειγμα ένας χαρακτήρας μπορεί να ικανοποιηθεί ή όχι από τις συνέπειες ενός γεγονότος (*pleased/displeased*), να επικροτήσει ή να απορρίψει τις δραστηριότητες κάποιου άλλου χαρακτήρα (*approve/disapprove*) και τέλος να του αρέσουν ή όχι τα χαρακτηριστικά ενός αντικειμένου (*like/dislike*).

Στο μοντέλο OCC υπάρχει μία διάκριση σχετικά με τον αποδέκτη των συνεπειών μίας δράσης. Εάν οι συνέπειες αφορούν κάποιον τρίτο μπορούν να διαχωριστούν σε επιθυμητές ή ανεπιθύμητες ενώ οι συνέπειες που αφορούν τον ίδιο τον χαρακτήρα μπορούν να διαχωριστούν σε σχετικές ή άσχετες προσδοκίες. Οι σχετικές προσδοκίες τέλος μπορούν να διαχωριστούν σε επιβεβαιωμένες ή μη ανάλογα με τον αν τελικά συμβαίνουν στον κόσμο ή όχι. Ο διαχωρισμός αυτός μπορεί να φανεί και στο παρακάτω Σχήμα 1 που δεν είναι παρά το αρχικό σχεδιάγραμμα του μοντέλου OCC.



Σχήμα 1 – Το μοντέλο του μοντέλου OCC [2]

Επίσης προσφέρει μία δομή μεταβλητών όπως η πιθανότητα ενός γεγονότος ή το κατά πόσο ένα αντικείμενο είναι γνώσιμο ή όχι στον παίκτη, που επιτρέπει να καθοριστεί η ένταση των διαφόρων τύπων συναισθημάτων. Το μοντέλο OCC είναι αρκετά λεπτομερειακό λοιπόν ώστε να καλύπτει σχεδόν όλες τις καταστάσεις στις οποίες μπορεί να βρεθεί ένας χαρακτήρας από πλευράς συναισθηματικών αλληλεπιδράσεων.

Μπορούμε να αναλύσουμε περαιτέρω την λειτουργία του μοντέλου εξετάζοντας την συμπεριφορά του σε καθεμία από τις φάσεις επεξεργασίας των γεγονότων του κόσμου, όπως αυτές παρουσιάστηκαν προηγουμένως.

Κατηγοριοποίηση

Κατά τη φάση της κατηγοριοποίησης ο χαρακτήρας επεξεργάζεται τα γεγονότα και αποκτά από την επεξεργασία αυτή πληροφορίες σχετικά με τις συναισθηματικές κατηγορίες που θα πρέπει να επηρεαστούν. Είναι προφανές ότι πέρα από το γεγονός (ή την πράξη ή το αντικείμενο) με το οποίο έρχεται αντιμέτωπος ο χαρακτήρας δεν του χρειάζεται κάποια εξωτερική πληροφόρηση, αφού είναι σε θέση να γνωρίζει τις προτιμήσεις του ίδιου του εαυτού.

Για παράδειγμα ας υποθέσουμε ότι ο έχουμε φτιάξει έτσι τον εικονικό μας χαρακτήρα ώστε να είναι πεπεισμένος ότι του αρέσουν οι μανάνες. Εάν ένας παίκτης του δώσει μερικές τότε οι χαρακτήρας θα εκτιμήσει αρχικά τις συνέπειες για τον παίκτη, και θα καταλήξει σε λύπη, καθώς ο παίκτης τώρα έχει λιγότερες μανάνες απ' ότι πριν και στην συνέχεια θα εκτιμήσει τις συνέπειες του γεγονότος για τον ίδιο του τον εαυτό, και θα καταλήξει σε ικανοποίηση, καθώς τώρα έχει επιτέλους

περισσότερες μπανάνες. Επίσης θα εκτιμήσει την πράξη του παίκτη που του έδωσε τις μπανάνες και θα καταλήξει σε θαυμασμό και τέλος θα εκτιμήσει και το ίδιο το αντικείμενο και θα καταλήξει σε αγάπη για αυτό.

Για να μπορέσει ένας χαρακτήρας να κάνει σωστά την διαδικασία της κατηγοριοποίησης όπως φαίνεται και από το παραπάνω παράδειγμα πρέπει κατά κύριο λόγο να έχει γνώση.

Αρχικά πρέπει να γνωρίζει την σχέση του με τον παίκτη, την οποία συνήθως προϋποθέτουμε ως καλή. Ακόμη πρέπει να γνωρίζει τι σημαίνει η κάθε πράξη για τον παίκτη (User Model) ώστε να μπορεί για παράδειγμα να ενεργοποιηθεί το συναίσθημα «χαρούμενος για κάποιον» όταν ο άλλος παίκτης προβεί σε μία πράξη που αναμένεται να τον κάνει να αισθανθεί καλύτερα. Κατά δεύτερον, πρέπει να έχει κάποιον στόχο (Goals) τον οποίο θέλει να επιτύχει και στον οποίο τα διάφορα αντικείμενα ή γεγονότα τον κάνουν να έρχεται πιο κοντά ή πιο μακριά. Για παράδειγμα στον στόχο «Επιβίωση» οι μπανάνες συνεισφέρουν θετικά. Τρίτον, πρέπει να ξέρει τι μπορεί να περιμένει από τον παίκτη, για να μπορεί αντίστοιχα να εκτιμήσει την αξία μίας πράξης. Όπως καταλαβαίνουμε όσο πιο απίθανη είναι μία «καλή» πράξη, τόσο πιο έντονο θετικό συναίσθημα περιμένουμε να προκαλέσει. Έτσι, στο παράδειγμά μας, ο χαρακτήρας πρέπει να ξέρει ότι ο παίκτης δεν τριγυρνάει μοιράζοντας μπανάνες ώστε να μπορέσει να νιώσει θαυμασμό όταν αυτός του δώσει μερικές. Τέλος, όπως προαναφέραμε ο χαρακτήρας προφανώς πρέπει να ξέρει ότι του αρέσουν οι μπανάνες (Attitudes)

Οι απαιτήσεις σε γνώση που έχει το μοντέλο OCC για τον χαρακτήρα πρέπει να προσδιοριστούν και να αποθηκευτούν κατά τον σχεδιασμό του από τον προγραμματιστή. Είναι σαφές ότι ένας εικονικός χαρακτήρας δεν μπορεί να έχει απόψεις από μόνος του, και άρα μπορούμε να τον βάλουμε σε όποιο καλούπι εξυπηρετεί τον σκοπό μας. Αυτή είναι και μια πρώτη έννοια προσωπικότητας του χαρακτήρα.

Ένας τρόπος να περιγραφεί αυτή η γνώση είναι καλύπτοντας όλες τις δυνατές περιπτώσεις καταστάσεων και συνθηκών με τις οποίες ενδεχομένως να βρεθεί αντιμέτωπος ο εικονικός χαρακτήρας, μαζί με πληροφορίες για το πως κάθε κατάσταση αντιμετωπίζεται, και τι συνέπειες έχει. Αυτή η προσέγγιση ταιριάζει μόνο σε χαρακτήρες που δρουν σε έναν περιορισμένο κόσμο, και στον οποίο τα ενδεχόμενα είναι λίγα και ελεγχόμενα.

Όταν όμως αυξάνεται ο αριθμός των γεγονότων, των πράξεων και των αντικειμένων είναι σαφής η ανάγκη για τον ορισμό αφαιρέσεων. Ένα τέτοιο παράδειγμα είναι η αντιστοίχιση της μπανάνας στην γενικότερα κατηγορία των φαγώσιμων, μαζί με τα μήλα το ψωμί κτλ. Εξάλλου το OCC είναι φτιαγμένο για να μοντελοποιεί σε ιδιαίτερα λεπτομερειακό βαθμό τα ανθρώπινα συναισθήματα, χωρίς όμως να είναι απαραίτητος αυτός ο βαθμός λεπτομέρειας για την σχεδίαση ενός πιστευτού χαρακτήρα.

Προσδιορισμός Έντασης

Η ένταση κάθε συναισθηματικής κατηγορίας υπολογίζεται ξεχωριστά για τα γεγονότα, τις πράξεις και τα αντικείμενα που συναντά ο παίκτης. Όσον αφορά τα γεγονότα η ένταση αντιστοιχίζεται με το κατά πόσο ο χαρακτήρας επιθυμούσε το γεγονός αυτό να συμβεί ή όχι, για τις πράξεις η ένταση σχετίζεται με το κατά πόσο η πράξη ήταν αξιόπαινη ή όχι, και τέλος για τα αντικείμενα η ένταση αφορά το κατά πόσο το αντικείμενο είναι αρεστό στον παίκτη.

Μία μεταβλητή που είναι απαραίτητη για να μπορεί ο χαρακτήρας να υπολογίσει το πόσο επιθυμητή είναι μία πράξη, είναι η ιεραρχία των στόχων του

χαρακτήρα. Για παράδειγμα, αν ένας στόχος έχει παραπάνω από μία ενέργειες που πρέπει να γίνουν ώστε να επιτευχθεί (π.χ. το καθάρισμα ενός φρούτου είναι επιμέρους στόχος όταν θέλουμε να κάνουμε μία φρουτοσαλάτα), τότε η επίτευξη του τελικού στόχου πρέπει να είναι πιο επιθυμητή από την επίτευξη ενός από τους επιμέρους στόχους. Βέβαια γεγονότα μπορεί να συμβαίνουν χωρίς απαραίτητα να έχουν να κάνουν με τους στόχους του χαρακτήρα, και σε αυτή την περίπτωση πρέπει να είναι ικανός να εκτιμήσει και αυτό το γεγονός. Δηλαδή, στο προηγούμενο παράδειγμα, αν ο παίκτης κοπεί φτιάχνοντας την φρουτοσαλάτα του, πρέπει να μπορεί να εκτιμήσει το γεγονός αυτό, άσχετα με τον αν δεν είναι ένα απαραίτητο βήμα που θα τον φέρει πιο κοντά στον στόχο του.

Αλληλεπίδραση

Το μοντέλο OCC δεν περιλαμβάνει το πολύ σημαντικό κομμάτι της αλληλεπίδρασης των διαφόρων συναισθηματικών κατηγοριών, και αυτό όπως θα δούμε και παρακάτω είναι ένα από τα μειονεκτήματά του.

Αντιστοίχιση

Εάν ο χαρακτήρας δεν μπορεί να εκφράσει και τις 22 συναισθηματικές κατηγορίες του OCC μοντέλου, τότε θα πρέπει οι κατηγορίες αυτές να αντιστοιχιστούν στις εκφράσεις που διαθέτει. Για παράδειγμα αν διαθέτει μόνο εκφράσεις προσώπου τότε μπορούμε να επικεντρωθούμε μόνο στα 6 βασικά συναισθήματα της χαράς, της λύπης, του θυμού, της αηδίας, του φόβου και της έκπληξης, τα οποία σύμφωνα με τον Ekman είναι τα μόνα συναισθήματα που μπορούν να είναι αναγνωρίσιμα από οποιονδήποτε άνθρωπο, ανεξαρτήτου φυλής και πολιτιστικής προέλευσης. [7]

Είναι ενδιαφέρον ότι και οι 11 θετικές συναισθηματικές κατηγορίες του OCC αντιστοιχίζονται στην μοναδική θετική έκφραση του προσώπου που είναι το χαμόγελο, ενώ αντιθέτως για τις αρνητικές κατηγορίες υπάρχουν οι αντίστοιχες αρνητικές εκφράσεις (θυμός, λύπη, αηδία και φόβος). Η έκφραση του προσώπου που αφορά την έκπληξη δεν μπορεί να αντιστοιχιστεί σε καμία κατηγορία του μοντέλου OCC, καθώς σύμφωνα με αυτό η έκπληξη δεν συμπεριλαμβάνεται στα συναισθήματα που μοντελοποιεί.

Το γεγονός ότι ο χαρακτήρας ενδεχομένως να είναι περιορισμένος στις 6 αυτές βασικές εκφράσεις του προσώπου δεν μειώνει την δυναμική του μοντέλου OCC καθώς ο παίκτης που έρχεται σε επαφή με τον χαρακτήρα θα μπορεί να αναγνωρίσει τα διάφορα συναισθήματα κρίνοντας και από το περιβάλλον στο οποίο τα εκδηλώνει ο χαρακτήρας. Για παράδειγμα, το χαμόγελο του χαρακτήρα που μόλις του δώσαμε μία μπανάνα, χαρακτηρίζεται εύκολα ως χαρά και εκτίμηση για το πρόσωπό μας, ενώ το χαμόγελο ενός χαρακτήρα που παρατηρούμε ότι μόλις κατάφερε να ανεβεί σε κάποιο μεγάλο εμπόδιο, αναγνωρίζεται εξίσου εύκολα ως περηφάνια και αυτοεκτίμηση. Είναι σημαντικό λοιπόν να διατηρηθούν οι συναισθηματικές κατηγορίες του μοντέλου, καθώς με βάση τα συμφραζόμενα ο παίκτης μπορεί να αντιληφθεί διάφορα συναισθήματα να εκφράζονται μέσω των ενδεχομένως περιορισμένων εκφράσεων του εικονικού χαρακτήρα.

Έκφραση

Η συναισθηματική κατάσταση του χαρακτήρα καθορίζεται από τις τιμές σε καθεμία από τις συναισθηματικές του κατηγορίες και αυτή η συναισθηματική κατάσταση πρέπει να εκφραστεί με κάθε δυνατό μέσο. Αυτό συμπεριλαμβάνει τις εκφράσεις του προσώπου, τον τόνο της φωνής, τον τρόπο με τον οποίο κινείται κτλ.

Οι εκφράσεις του προσώπου έχουν μελετηθεί αρκετά αλλά παραμένουν αρκετά ερωτήματα όπως για παράδειγμα το αν πρέπει ο χαρακτήρας να εκδηλώνει μία μόνο έκφραση στο πρόσωπό του ή ένα συνδυασμό αυτές που μπορεί. Για παράδειγμα η δεύτερη επιλογή σίγουρα παράγει ένα πιο αληθοφανές αποτέλεσμα αλλά απαιτεί έναν ιδιαίτερα λεπτομερειακό χαρακτήρα και ενδεχομένως να κάνει την όλη διαδικασία πολύ απαιτητική για τον υπολογιστή ενός απλού χρήστη. Έτσι λοιπόν, αν και σε επιστημονικές εφαρμογές μπορούμε να έχουμε τον συνδυασμό πολλαπλών εκφράσεων στο πρόσωπο του εικονικού χαρακτήρα, στην ανάπτυξη εφαρμογών που έχουν να κάνουν με παιχνίδια και γενικότερα με εμπορικές εφαρμογές κάτι τέτοιο θα ήταν δύσκολο να επιτευχθεί με αποδοτικό τρόπο.

Επίσης, για τους χαρακτήρες που συνομιλούν με τον παίκτη, είναι ιδιαίτερα σημαντικό να συμβαδίζει ο τόνος της φωνής με την συναισθηματική κατάσταση που εκείνη τη στιγμή προβάλλει ο χαρακτήρας. Εάν για παράδειγμα βλέπουμε έναν χαμογελαστό χαρακτήρα περιμένουμε να τον ακούσουμε να μιλάει με διαφορετικό τόνο απ' ότι κάποιος άλλος που εκδηλώνει θυμό.

Ένα ακόμα πολύ σημαντικό στοιχείο είναι να συνειδητοποιήσουμε ότι ένας εικονικός χαρακτήρας μπορεί να εκφράσει την συναισθηματική του κατάσταση όχι μόνο μέσω του προσώπου του αλλά και μέσω ολόκληρου του σώματός του, καθώς επίσης και μέσω της συμπεριφοράς του. Είναι σαφές ότι οι πράξεις μας μια δεδομένη στιγμή δεν καθορίζονται μόνο από τον χαρακτήρα μας αλλά και από την ψυχολογική μας κατάσταση.

Τέλος, επειδή η αντιστοίχιση της συναισθηματικής κατάστασης σε έκφραση, είναι και το τελικό αποτέλεσμα της λειτουργίας του μοντέλου, πρέπει να γίνει με πολύ προσεκτικό τρόπο καθώς ακόμη και αν η όλη διαδικασία έχει γίνει με άρτιο τρόπο, αν η έκφραση του παίκτη δεν αντιστοιχεί στο συναίσθημα που θέλουμε αυτός να εκφράζει, τότε το αποτέλεσμα δεν θα είναι πειστικό.

Γι' αυτό είναι αναγκαίο να υπάρχουν δυνατές θεωρητικές βάσεις πίσω από την διαδικασία της έκφρασης ώστε τα προσωπικά βιώματα του σχεδιαστή να μην επηρεάσουν το τελικό αποτέλεσμα, και φυσικά να υπάρχει η κατάλληλη διαδικασία ελέγχου από ανεξαρτήτους χρήστες ώστε να ελεγχθεί και πρακτικά η εντύπωση που δίνουν οι εκφράσεις του χαρακτήρα σε κάποιον που δεν είναι γνώστης της όλης διαδικασίας του υπάρχει από πίσω.

Ένα παράδειγμα που αφορά ακριβώς αυτή την ανάγκη μπορούμε να δούμε με το σύστημα Nemesys που αναπτύχθηκε από τους Eckschlager, Lanke και Bernhaupt με σκοπό να ελεγχθεί αν οι παίκτες μπορούν πράγματι να αντιληφθούν τα συναισθήματα που εκφράζουν οι εικονικοί χαρακτήρες. [6]

Βέβαια το σύστημα αυτό δεν ήταν βασισμένο στο μοντέλο OCC αλλά παρ' όλα αυτά έχει σημασία η λειτουργικότητά του και όχι τόσο η υλοποίησή.

Στο εγχείρημα αυτό ο χρήστης βρίσκεται σε έναν ειδικά διαμορφωμένο χώρο ώστε να μπορούν να καταγράφονται οι αντιδράσεις του και γενικότερα και η δική του συναισθηματική κατάσταση. Αυτό γίνεται γιατί είναι ιδιαίτερα χρήσιμο στον συγκεκριμένο ερευνητικό τομέα να ανιχνεύονται παράλληλα και τα συναισθήματα που έχει ο χρήστης ώστε να μπορεί να επαληθευθεί αν πράγματι η εισαγωγή συναισθημάτων (emotion induction) στους NPCs κάνει τον παίκτη να βιώνει περισσότερο την εμπειρία του παιχνιδιού, όπως έχουμε εξηγήσει άλλωστε ότι είναι και ο στόχος μας.

Στο Nemesys λοιπόν, ο χρήστης πρέπει να επικοινωνήσει με διάφορους εικονικούς χαρακτήρες και πρέπει να ερμηνεύσει σωστά τα συναισθήματα που αυτοί εκφράζουν ώστε να μπορέσει να τελειώσει το παιχνίδι. Η επικοινωνία των

συναισθημάτων δεν γίνεται μόνο μέσα από τους καθορισμένους διαλόγους αλλά και από κινήσεις του σώματος, όπως άλλωστε συμβαίνει και στην πραγματικότητα.

Τέλος μία επέκταση που σχεδιάζεται για το σύστημα είναι να υπάρχει ανατροφοδότηση των βιομετρικών στοιχείων του παίκτη με σκοπό να μπορούν οι NPCs να χρησιμοποιήσουν και αυτή την πληροφορία για να καθορίσουν τις αντιδράσεις τους και τα συναισθήματά τους απέναντι στα διάφορα γεγονότα του εικονικού κόσμου.

2.1.3 Ελλείψεις και μειονεκτήματα του OCC μοντέλου και πιθανές επεκτάσεις

Όταν κάποιος έρχεται αντιμέτωπος με την πολυπλοκότητα του μοντέλου OCC του είναι εύκολο να δεχθεί ότι είναι ότι ακριβώς του χρειάζεται ώστε να προσθέσει συναισθήματα στους εικονικούς του χαρακτήρες. Ωστόσο παρά την πολύ καλή δομή του, το μοντέλο OCC δεν είναι χωρίς ελλείψεις και αυτές γίνονται εμφανείς κατά την διαδικασία ανάπτυξης του εικονικού χαρακτήρα. [2]. Οι ελλείψεις αυτές συχνά κρίνονται ως λεπτομέρειες αλλά είναι αρκετές (εάν δεν ληφθούν υπ' όψιν) να κάνουν τον χαρακτήρα να μοιάζει με έναν διόλου πιστευτό κλόουν.

1. Συνάρτηση ιστορικού

Κατά την φάση καθορισμού της έντασης μίας συναισθηματικής κατηγορίας, θα ήταν πολύ χρήσιμη η ύπαρξη μίας συνάρτησης ιστορικού, η οποία αντιπροσωπεύει ότι έχει συμβεί στον χαρακτήρα από την άποψη γεγονότων, πράξεων και αντικειμένων. Είναι λογικό ότι τα νέα γεγονότα πρέπει να εκτιμώνται λαμβάνοντας υπ' όψιν τι άλλο έχει συμβεί στον χαρακτήρα. Για παράδειγμα (επιστρέφουμε στις μπανάνες στις οποίες έχει ο χαρακτήρας αδυναμία) αν ένας παίκτης δίνει συνεχώς μπανάνες στον χαρακτήρα είναι αναμενόμενο το γεγονός αυτό να γίνεται όλο και λιγότερο επιθυμητό για τον χαρακτήρα και άρα ο χαρακτήρας να μην ενθουσιάζεται τόσο όσο αρχικά.

Ένα άλλο πλεονέκτημα της συνάρτησης ιστορικού έχει κάνει με το γεγονός ότι το μοντέλο OCC χρησιμοποιεί την πιθανότητα ενός γεγονότος για να υπολογίσει πόσο επιθυμητό είναι. Έτσι μπορεί κάθε γεγονός να έχει μία αρχική πιθανότητα να συμβεί, αλλά αυτή μπορεί να αλλάξει αν σε ένα μικρό χρονικό διάστημα το γεγονός επαναληφθεί με ρυθμό που δεν αντιστοιχεί με αυτή την a-priori εκτίμηση πιθανότητας. Αν λοιπόν υποθέσουμε ότι ο παίκτης για κάποιον δικό του λόγο αρχίζει να δίνει συνεχώς μπανάνες σε έναν χαρακτήρα (υπόθεση που υποθέτουμε ότι αντιβαίνει με την αρχική εκτίμηση της πιθανότητας να αποχωριστεί ο παίκτης τις μπανάνες του) τότε ο χαρακτήρας μπορεί μέσω της συνάρτησης ιστορικού να υπολογίσει την πραγματική συχνότητα με την οποία συμβαίνει το γεγονός αυτό και άρα να προσαρμόσει την συμπεριφορά του αντίστοιχα. Αν τώρα περάσει μία κάποια περίοδος χωρίς να συμβεί τίποτα, η τιμή της πιθανότητας του γεγονότος ξανα-αρχικοποιείται στην προκαθορισμένη της τιμή.

Τέλος η συνάρτηση ιστορίας μπορεί να αποδειχθεί χρήσιμη όσον αφορά την δυνατότητα να παρακολουθηθεί η πορεία που ακολουθεί ο χαρακτήρας καθώς και η πρόοδος που έχει κάνει για την επίτευξη ενός στόχου. Δεδομένου ότι το μοντέλο OCC συνυπολογίζει την προσπάθεια του χαρακτήρα για να καθορίσει πόσο επιθυμητό είναι ένα γεγονός, η συνάρτηση

ιστορίας θα μπορούσε να βοηθήσει ώστε να υπάρχει ένα αρχείο με τις κινήσεις του χαρακτήρα και κατ' επέκταση μία αρχική βάση ώστε να υπολογιστεί η προσπάθεια που έχει κάνει και πόσο κοντά έχει φτάσει στο στόχο του.

2. Σχεδιαστής προσωπικότητας

Το μοντέλο OCC είναι σχεδιασμένο ώστε να μοντελοποιεί τα συναισθήματα με τρόπο γενικά συμβατό με τα ανθρώπινα δεδομένα. Εάν λάβουμε υπ' όψιν ότι η συνέπεια των πράξεων είναι ένα βασικό χαρακτηριστικό που κάνει τους εικονικούς χαρακτήρες πιστευτούς, τότε μπορούμε να εκφράσουμε αυτή της συνέπεια ως προς την συμπεριφορά σαν την προσωπικότητα του εικονικού χαρακτήρα.

Οι χρήστες είναι συνηθισμένοι στην αλληλεπίδραση με ανθρώπους που εκφράζουν την προσωπικότητά τους μέσω της συμπεριφοράς και των αντιδράσεων τους. Έτσι η φαινομενικά ουδέτερη συμπεριφορά ενός εικονικού χαρακτήρα ενδεχομένως να γίνει αντιληπτή σαν αφύσικη και γι' αυτό θα ήταν χρήσιμο να υπάρχει ένας τρόπος να καθορίζεται εκ των προτέρων μία προσωπικότητα για τον εικονικό χαρακτήρα και να μπορεί να συμπεριφέρεται με βάση αυτήν.

3. Αλληλεπίδραση συναισθηματικών κατηγοριών

Όλοι μας ξέρουμε από την καθημερινή μας ζωή ότι τα συναισθήματα δεν λειτουργούν με μια Boolean λογική τύπου On-Off. Αντιφατικά συναισθήματα μπορούν να συνυπάρχουν και ένα νέο γεγονός που μας επηρεάζει δεν υπερκαλύπτει την προηγούμενη συναισθηματική μας κατάσταση

Κάτι αντίστοιχο περιμένουμε να ισχύει και για τους εικονικούς μας χαρακτήρες. Ας υποθέσουμε για παράδειγμα ότι ο χαρακτήρας μας καταφέρνει να κοπεί φτιάχνοντας την περιβόητη φρουτοσαλάτα του, τότε μάλλον θα καταλήξει θυμωμένος με τον εαυτό του. Αν τώρα ένας παίκτης αποφάσισε να του χαρίσει μία μπανάνα, θα ήταν μάλλον αφύσικο να δούμε τον χαρακτήρα να χοροπηδάει από την χαρά του, δείχνοντας να έχει διαγράψει από τη μνήμη του το πρόσφατο κόψιμο. Μία πιο λογική αντίδραση θα ήταν απλά να μετριαστεί ο θυμός του και να πάει προς μια πιο θετική συναισθηματική κατάσταση.

Φυσικά είναι αδύνατο να μπορέσει κανείς να προβλέψει με ακρίβεια τη επιρροή θα έχει η κάθε συναισθηματική κατηγορία στην άλλη, καθώς υπάρχουν πολυάριθμοι παράγοντες που μπορεί να καθορίσουν την αντίδραση ενός πραγματικού ανθρώπου σε αντίστοιχες συνθήκες, όπως για παράδειγμα ο χαρακτήρας του.

Μια πιο απλή και ωστόσο ρεαλιστική προσέγγιση θα ήταν να υπάρχει μια αντιστοίχιση μεταξύ των συναισθημάτων σε ζεύγη θετικό-αρνητικό (π.χ. χαρά-λύπη, ενθουσιασμός-απογοήτευση) και να επηρεάζει η αλλαγή ενός από τα δύο μόνο το αντίστοιχό του.

Όσον αφορά τα μειονεκτήματα του μοντέλου, όπως είδαμε και κατά την περιγραφή της φάση της κατηγοριοποίησης, το OCC μοντέλο ενδεχομένως να περιλαμβάνει έναν βαθμό λεπτομέρειας που να ξεπερνάει τις ανάγκες μας στην δημιουργία πιστευτών χαρακτήρων, και τα επιχειρήματα προς αυτή την κατεύθυνση είναι αρκετά.

- Οι χαρακτήρες συχνά δεν μπορούν να αποτυπώσουν όλες τις συναισθηματικές κατηγορίες. Για παράδειγμα ένας εικονικός χαρακτήρας με την ικανότητα να εκδηλώνει τα συναισθήματά του μόνο μέσω εκφράσεων του προσώπου θα πρέπει να έχει μόνο αυτές τις συναισθηματικές κατηγορίες που μπορεί και να εκφράσει. Επομένως θα πρέπει κάθε φορά στο μοντέλο να χρησιμοποιούνται μόνο οι κατηγορίες που μπορεί να χρησιμοποιήσει ο χαρακτήρας.
- Μερικές κατηγορίες του OCC μοιάζουν να είναι πολύ στενά συνδεδεμένες με άλλες, όπως για παράδειγμα η ευγνωμοσύνη (gratitude) και η ικανοποίηση (gratification), αν και αναλύοντάς τις βλέπουμε ότι πηγάζουν από διαφορετικά αίτια (η ικανοποίηση προκύπτει από μία πράξη που έκανε ο ίδιος ο χαρακτήρας, ενώ η ευγνωμοσύνη από μια πράξη κάποιου άλλου). Γενικά δεν είναι ξεκάθαρο το αν μία τόσο λεπτομερής διαφοροποίηση των συναισθημάτων έχει κάποιο πρακτικό πλεονέκτημα στο ζήτημα της ρεαλιστικής συμπεριφοράς των εικονικών χαρακτήρων.
- Στην περίπτωση που ο χαρακτήρας στερείται μοντέλου για την συμπεριφορά του παίκτη (User Model) τότε όλα τα συναισθήματα που έχουν να κάνουν που εμπίπτουν στην κατηγορία «fortune of others» μπορούν να παραλειφθούν καθώς ο χαρακτήρας δεν θα είναι σε θέση να αξιολογήσει τις επιπτώσεις κάποιου γεγονότος για τον άλλο παίκτη.

Φαίνεται λοιπόν ότι το OCC μοντέλο δεν είναι τέλειο. Υπάρχει ανάγκη τόσο για προσθήκη όσο και για απλοποίηση ορισμένων χαρακτηριστικών. Χαρακτηριστικό είναι δε το γεγονός ότι και ο ίδιος ο Ortony παραδέχτηκε σχετικά ότι το μοντέλο OCC είναι ενδεχομένως υπερβολικά περίπλοκο για την ανάπτυξη πιστευτών εικονικών χαρακτήρων. Έτσι πρότεινε την χρήση 5 μόνο θετικών και 5 αρνητικών κατηγοριών από τις 22 που περιλαμβάνει το αρχικό μοντέλο. Οι κατηγορίες αυτές είναι, όσον αφορά τα θετικά συναισθήματα: χαρά, ελπίδα, ανακούφιση, ευγνωμοσύνη και αγάπη, και άγχος, φόβος, απογοήτευση, θυμός και μίσος, όσον αφορά τα αρνητικά. Είναι ενδιαφέρον να παρατηρήσουμε ότι επέκλεισε όλες τις κατηγορίες που απαιτούν την ύπαρξη ενός User Model για τα χαρακτηριστικά του παίκτη.

Περαιτέρω μελέτες, μειώνουν ακόμη περισσότερο τις διαθέσιμες κατηγορίες για την περιγραφή της συναισθηματικής κατάστασης του παίκτη, ώστε να επιτευχθεί μία ένα-προς-ένα αντιστοίχιση των συναισθημάτων με τις διαθέσιμες εκφράσεις του προσώπου. Αυτό είναι λογικό αν υποθέσουμε ότι συνήθως τα συναισθήματά μας εκφράζονται περισσότερο μέσω του προσώπου μας και λιγότερο μέσω του υπολοίπου σώματος.

Το συμπέρασμα λοιπόν που μπορούμε να βγάλουμε είναι ότι το μοντέλο OCC επαρκεί για την μοντελοποίηση των συναισθημάτων του εικονικού χαρακτήρα αλλά δεν δίνει μία επαρκή λύση όσον αφορά τι πρέπει να κάνει ο χαρακτήρας με την συναισθηματική κατάσταση που μόλις υπολόγισε ότι έχει. Χρειαζόμαστε λοιπόν επιπλέον βοηθητικές δομές για να μπορέσει ο χαρακτήρας να εκφράσει τα συναισθήματά του με τρόπο ώστε να μπορούν να γίνουν αντιληπτά από τους άλλους παίκτες. Επίσης η απλοποίηση του μοντέλου έτσι ώστε να συμβαδίζει με τις εκφραστικές ικανότητες του παίκτη δείχνει να βοηθάει αρκετά όσον αφορά το μέγεθος και τη δυσκολία της διαδικασίας ανάπτυξης.

2.2 Στόχοι κατά τον σχεδιασμό ενός διαδραστικού Non Player Character

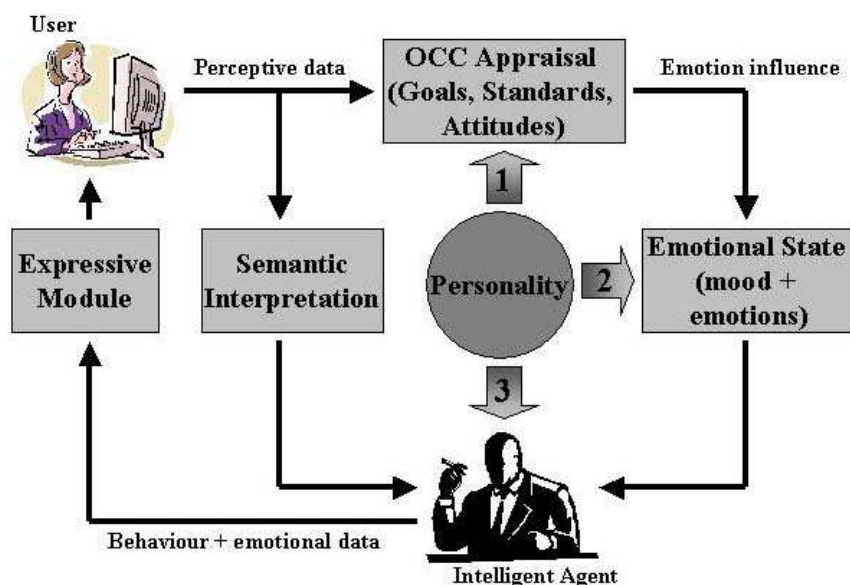
Οι Εικονικοί Χαρακτήρες που σχεδιάζουμε με στόχο να είναι όσο πιο πιστευτοί στους ανθρώπους με τους οποίους έρχονται σε επαφή εκτός από τον συναισθηματικό τους κόσμο πρέπει να έχουν και άλλα χαρακτηριστικά ώστε να υπάρχει η απαραίτητη συνοχή στην συμπεριφορά τους.

2.2.1 Εικονική προσωπικότητα

Με την εξέλιξη των τρισδιάστατων γραφικών μπορούμε πλέον να δημιουργήσουμε όλο και πιο πιστευτούς χαρακτήρες, τουλάχιστον όσον αφορά την εικόνα τους. Αυτό όμως που συχνά λείπει είναι ο καθορισμός της κινητήριας δύναμης των χαρακτήρων η οποία δεν είναι παρά η ατομικότητά τους. Ο στόχος μας είναι να καθορίσουμε τι σχέση έχει η προσωπικότητα του κάθε χαρακτήρα με την αντίληψη, την επικοινωνία και την έκφρασή του. [6]

Όπως αναφέραμε και στην προηγούμενη ενότητα συνήθως η διαδικασία της αξιολόγησης που κάνει ο χαρακτήρας προκειμένου να καθορίσει την συμπεριφορά του, οδηγείται από ένα σύστημα που βασίζεται στο OCC μοντέλο σύμφωνα με το οποίο τα διάφορα γεγονότα, πράξεις και αντικείμενα του κόσμου αξιολογούνται με βάση τους στόχους (Goals), την νοοτροπία (Attitude) και τα πρότυπα (Standards) του χαρακτήρα. Τα τρία αυτά χαρακτηριστικά καθορίζουν και την προσωπικότητα του χαρακτήρα. Πρόσφατες έρευνες όμως δείχνουν ότι η προσωπικότητα μπορεί να μοντελοποιηθεί και με ένα πιο αφηρημένο τρόπο ο οποίος δεν εξαρτάται από τον τομέα για τον οποίο γίνεται ο σχεδιασμός του χαρακτήρα.

Στο παρακάτω Σχήμα 2 [6]. βλέπουμε τον ρόλο ακριβώς που παίζει η προσωπικότητα σαν συνδετικό στοιχείο μεταξύ της αντίληψης, της έκφρασης και της επικοινωνίας. Τα δεδομένα που προέρχονται από τις αισθήσεις αξιολογούνται σε συναισθηματική βάση από το κατάλληλο μοντέλο (π.χ. OCC) και το αποτέλεσμα που προκύπτει αποτελεί την επιρροή των συναισθημάτων η οποία μαζί με την προσωπικότητα του χαρακτήρα θα καθορίσουν την νέα συναισθηματική κατάσταση και διάθεση του χαρακτήρα. Ένας ευφυής χαρακτήρας τώρα λαμβάνει υπ' όψιν όλα αυτά τα δεδομένα και παράγει την αντίστοιχη συμπεριφορά.



Σχήμα 2 – Ο ρόλος της προσωπικότητας και της συναισθηματικής κατάστασης στα πλαίσια ενός ευφυούς πράκτορα. [6]

Ένας χαρακτήρας (εικονικός ή μη) λοιπόν, είναι μία οντότητα που διαρκώς αλλάζει καθώς μεταβάλλεται η συναισθηματική του κατάσταση. Αντίθετα όμως με τα συναισθήματα η προσωπικότητα του χαρακτήρα είναι κάτι το στατικό και έχει την ίδια επιρροή κάθε χρονική στιγμή. Το μοντέλο που περιγράφει τους χαρακτήρες μέσω των συναισθημάτων και την προσωπικότητάς τους ονομάζεται μοντέλο PE (Personality, Emotional State).

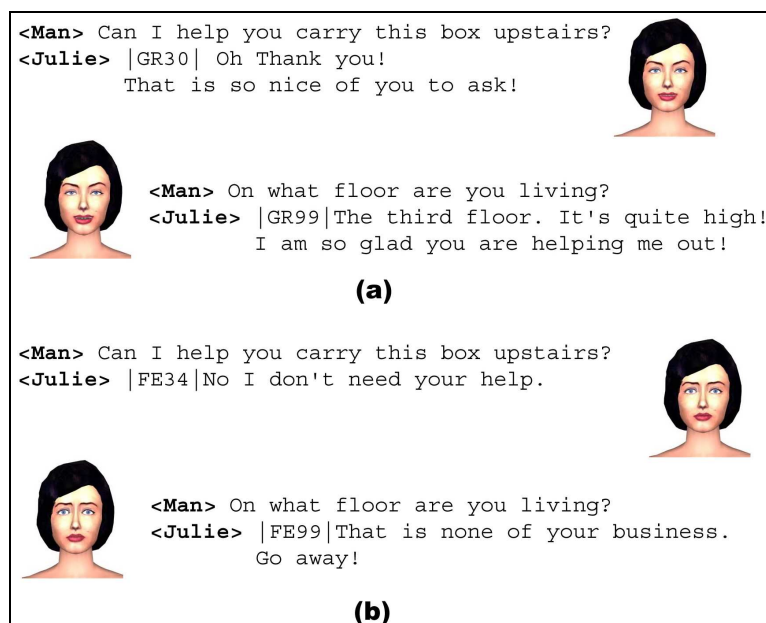
Όπως έχουμε δει και στην περιγραφή του OCC η συναισθηματική κατάσταση του χαρακτήρα καθορίζεται από την ένταση που χαρακτηρίζει καθεμία από τις συναισθηματικές κατηγορίες (στο OCC αυτές είναι 22, ενώ αντίστοιχα ο Ekman ορίζει 6 βασικές και παγκόσμια αναγνωρίσιμες κατηγορίες για την περιγραφή των συναισθημάτων των εκφράσεων του προσώπου). Κατ' επέκταση μπορούμε να ορίσουμε ένα παρόμοιο μοντέλο για την προσωπικότητα του χαρακτήρα. Υπάρχουν διάφορα δημοφιλή μοντέλα που ορίζουν τις διάφορες διαστάσεις της ανθρώπινης προσωπικότητας. Για παράδειγμα υπάρχει το μοντέλο OCEAN σύμφωνα με το οποίο η προσωπικότητα ενός χαρακτήρα μπορεί να εκφραστεί σαν σταθμισμένος γραμμικός συνδυασμός των παρακάτω βασικών συνιστωσών: Openness, Conscientiousness, Extraversion, Agreeableness, και Neuroticism.

Το μοντέλο λοιπόν της προσωπικότητας είναι αυτό που σε συνδυασμό με το μοντέλο των συναισθημάτων θα οδηγήσει στον τελικό καθορισμό της συμπεριφοράς του χαρακτήρα. Αν αναλογιστούμε όμως ότι τα 3 βασικά χαρακτηριστικά στα οποία βασίζεται το OCC έχουν σε μεγάλο βαθμό να κάνουν με τον τομέα στον οποίο δραστηριοποιείται ο χαρακτήρας, προκύπτει ένα ερώτημα σχετικά με το πως θα μπορέσει το OCC να συνεργαστεί με ένα αφαιρετικό και ανεξάρτητο μοντέλο προσωπικότητας. Αυτό θα μπορούσε να επιτευχθεί εάν θεωρήσουμε ότι τα βασικά χαρακτηριστικά που χρειάζεται το OCC είναι πλήρως ορισμένα για το περιβάλλον στο οποίο κινείται ο χαρακτήρας και έτσι το μοντέλο προσωπικότητας μπορεί να λειτουργήσει ως ένα κριτήριο επιλογής το οποίο καταδεικνύει ποιοι από τους στόχους και τα πρότυπα είναι συμβατά με την προσωπικότητα που θέλει να υλοποιήσει ο εκάστοτε χαρακτήρας. Για παράδειγμα εάν χρησιμοποιήσουμε το μοντέλο OCEAN ο παρακάτω πίνακας μπορεί να μας δείξει ποια είναι η σχέση ανάμεσα στις διαστάσεις του μοντέλου και στα βασικά χαρακτηριστικά του OCC.

	Goals	Standards	Attitudes
Openness		Αλλαγή στα πρότυπα σε νέες καταστάσεις	Αντιμετώπιση απέναντι σε νέα στοιχεία
Conscientiousness	Εγκατάλειψη και υιοθέτηση στόχων, αποφασιστικότητα		
Extraversion			Επιθυμία για επικοινωνία
Agreeableness	Εγκατάλειψη και υιοθέτηση στόχων για το καλό άλλων	Συμβιβασμός στα πρότυπα για το καλό άλλων	Προσαρμοστικότητα σε άλλους ανθρώπους
Neuroticism			

Τέλος, ένα παράδειγμα που δείχνει την λειτουργία της προσωπικότητας στην συμπεριφορά του χαρακτήρα είναι το εξής. Ας υποθέσουμε το παρακάτω σενάριο: η Jill είναι μια κοπέλα που μόλις γύρισε από τα ψώνια της και βρίσκεται έξω από το σπίτι της φορτωμένη με σακούλες. Ένας περαστικός σταματάει και προσφέρεται να την βοηθήσει. Εάν τώρα η Jill είναι εξωστρεφής περιμένουμε να χαρεί και να δεχθεί την ευγενική προσφορά του αγνώστου. Στην αντίθετη περίπτωση που η Jill είναι ντροπαλή και ελαφρώς νευρωτική μάλλον θα τρομάξει από την ξαφνική εισβολή στον προσωπικό της χώρο και θα αρνηθεί την βοήθεια. Φυσικά στην πραγματικότητα κανείς άνθρωπος δεν έχει ακραία χαρακτηριστικά γι' αυτό και η προσωπικότητα του κάθε χαρακτήρα εκφράζεται σαν ένας συνδυασμός των χαρακτηριστικών του μοντέλου, σε διάφορες όμως εντάσεις.

Στο παρακάτω Σχήμα 3 [6] βλέπουμε διαλόγους που προέκυψαν από την υλοποίηση ενός τέτοιου συστήματος και εφαρμογή του παραπάνω σεναρίου για διαφορετικές προσωπικότητες του εικονικού χαρακτήρα.



Σχήμα 3 – Η συμπεριφορά της Jill σαν (α) εξωστρεφής και (β) εσωστρεφής προσωπικότητα. [6]

2.2.2 Δυναμική προσαρμογή στα χαρακτηριστικά του χρήστη

Οι τεχνολογίες που αφορούν την μάθηση στα ψηφιακά παιχνίδια γίνονται όλο και πιο σημαντικές. Ένα κύριο χαρακτηριστικό πολλών μοντέλων μάθησης είναι ότι λειτουργούν off-line, και αυτό σημαίνει ότι ένας NPC για παράδειγμα μπορεί να εκπαιδευτεί σύμφωνα με κάποια πρότυπα συμπεριφοράς και μετά να εισαχθεί στον εικονικό κόσμο χωρίς όμως να έχει πια την ικανότητα να μάθει. Αυτός είναι ένας περιορισμός που θέλουμε σιγά σιγά να υπερνικήσουμε ώστε να μπορέσουμε στο μέλλον να σχεδιάσουμε χαρακτήρες που αναλύουν αυτά που συμβαίνουν, και κυρίως τις συνέπειες των πράξεών τους, και δείχνουν την ικανότητά τους να μαθαίνουν από το παρελθόν και να αξιοποιούν στο έπακρο τις πληροφορίες που τους προσφέρει.

Έχουν γίνει πολύ σημαντικά βήματα σε αυτή την κατεύθυνση όπως τα παιχνίδια Black & White και The Sims όπου βλέπουμε στο μεν πρώτο ένα νευρωνικό δίκτυο που διαρκώς επανα-εκπαιδεύεται με σκοπό να καταγράφει επιτυχώς τις επιθυμίες του παίκτη, στο μεν δεύτερο οι κοινωνικές επαφές οδηγούν την συμπεριφορά των παικτών οι οποίοι αναπτύσσουν επαφές μεταξύ τους στα πρότυπα μιας κανονικής κοινωνίας και οι οποίοι μπορεί να αλλάξουν τις προτιμήσεις τους βασιζόμενοι στην προηγούμενη εμπειρία τους. Μάλιστα, αρκετή έρευνα αφορά ακριβώς αυτόν τον τομέα, την ανάπτυξη δηλαδή ευφυών κοινωνικών αισθητήρων για να εξοπλίσουν NPCs με ανάγκη για ρεαλιστικές κοινωνικές επαφές.

Αν και η δυναμική μάθηση είναι ένα πολύ επιθυμητό χαρακτηριστικό για διάφορους λόγους, όπως για παράδειγμα για να μπορούν οι εικονικοί χαρακτήρες να προσαρμόζονται στον διαφορετικό τρόπο παιχνιδιού του κάθε παίκτη, είναι ακόμη αρκετά δύσκολο να υλοποιηθεί. Ένα βασικό ζήτημα είναι ότι ένα τέτοιο μοντέλο, καθαρά λόγω του δυναμικού του χαρακτήρα, μπορεί να οδηγήσει σε πολύ απρόβλεπτα αποτελέσματα, τα οποία να μεν μπορεί μερικές φορές να εμπλουτίζουν το παιχνίδι αλλά συνήθως οδηγούν σε εσφαλμένες συμπεριφορές οι οποίες μειώνουν το επίπεδο του gameplay και σε ακόμη χειρότερη περίπτωση μπορούν να οδηγήσουν και σε δυναμικά bugs τα οποία είναι προφανές ότι δεν μπορούν να ελεγχθούν πόσο μάλλον με τις κλασικές διαδικασίες αποσφαλμάτωσης. Ο δυναμικός χαρακτήρας λοιπόν είναι ένα δίκιο μαχαίρι και γι' αυτό απαιτείται πολύ λεπτομερικός σχεδιασμός κατά την υλοποίηση ενός τέτοιου συστήματος.

Παρά το γεγονός ότι η διαδικασία ανάπτυξης σθεναρών και αποτελεσματικών αλγορίθμων δυναμικής μάθησης και των αντίστοιχων αρχιτεκτονικών για τα παιχνίδια παρουσιάζει πολλά εμπόδια, οι ανταμοιβές ένα κάτι τέτοιο επιτευχθεί αξίζουν τον κόπο. Το μεγαλύτερο κέρδος θα είναι σίγουρα η δυναμική προσαρμογή στην συμπεριφορά του παίκτη και στις ικανότητές του. Πιο συγκεκριμένα ένας ιδιαίτερος στόχος είναι η ανάπτυξη τεχνολογιών που να δίνουν την δυνατότητα να τροποποιείται (παροδικά ή για μεγαλύτερο χρονικό διάστημα) ο κόσμος ή και η συμπεριφορά των αντιπάλων ώστε να μπορεί ο παίκτης να ξεπερνάει κάποια δυσκολία που ενδεχομένων αντιμετωπίζει σε κάποιο σημείο.

Επίσης είναι ένα επιθυμητό χαρακτηριστικό να μπορεί η δυναμική μάθηση να βοηθάει τους εικονικούς χαρακτήρες να αναπτύξουν μία φυσιολογική ανθρώπινη συμπεριφορά. Κατά την διάρκεια του παιχνιδιού ένας άνθρωπος έχει συναίσθηση των πράξεών του και των συνεπειών που είχαν αυτές. Έτσι καταφέρνει να αποφεύγει τις δυσάρεστες επαναλήψεις, εκφράζοντας την ευφυΐα του να μαθαίνει κανείς από τα λάθη του. Όταν λοιπόν ο στόχος μας είναι η πειστικότητα του NPC πρέπει να φροντίσουμε να ακολουθούν μία παρόμοια συμπεριφορά.

2.2.3 Επιρροή της διάθεσης

Πέρα από την επιρροή των συναισθημάτων και της προσωπικότητας στην συμπεριφορά ενός εικονικού χαρακτήρα μπορούμε να μιλήσουμε για ένα ΡΜΕ μοντέλο (Personality, Mood, Emotional State) στο οποίο και η διάθεση του χαρακτήρα παίζει τον δικό της ρόλο. Ξέρουμε από την καθημερινή ζωή ότι η προσωπικότητα δεν είναι ο αποκλειστικός παράγοντας που καθορίζει την συμπεριφορά ενός ανθρώπου, καθώς ο ίδιος άνθρωπος μπορεί να αντιδρά στο ίδιο ερέθισμα με διαφορετικό τρόπο σε διαφορετικές χρονικές στιγμές.

Η έννοια της διάθεσης μπορεί να εκφράσει αυτό το χαρακτηριστικό της ανθρώπινης συμπεριφοράς και συνήθως μοντελοποιείται με έναν απλό μονοδιάστατο τρόπο στην λογική της καλής ή κακής διάθεσης, σαν μία μεταβλητή στο διάστημα [-1, 1]. Μπορούμε επίσης να έχουμε και αρχείο σχετικά με τις τελευταίες διαθέσεις από τις οποίες έχει περάσει ο παίκτης, και έτσι η τελική του συμπεριφορά να προκύπτει σαν συνάρτηση των τριών αυτών βασικών παραγόντων: συναισθηματική κατάσταση, προσωπικότητα και διάθεση.

2.3 Παράδειγμα Συστήματος με NPCs που συνομιλούν

Στα πλαίσια του σχεδιασμού NPCs που μπορούν να εκφράσουν μία προσωπικότητα και μία συγκεκριμένη ψυχολογική κατάσταση έχουν γίνει διάφορες προσπάθειες να δημιουργηθεί ένα πραγματικό σύστημα που να μπορεί να δοκιμαστεί σε πραγματικές συνθήκες.

Όπως έχουμε προαναφέρει στα παιχνίδια είναι ιδιαίτερα σημαντικό να νομίζει ο παίκτης ότι ανταγωνίζεται άλλους ανθρώπους. Αυτό έχει να κάνει με το γεγονός ότι δύσκολα μπορεί κανείς να νιώσει ικανοποίηση νικώντας κάποιον που πολύ απλά δεν τον νοιάζει αν θα ηττηθεί. Χαρακτηριστικά αυτό φαίνεται από τις αντιδράσεις των ίδιων των παικτών που προτιμούν να παίζουν με λιγότερο ικανούς παίκτες ή ακόμη και να χάνουν από πιο ικανούς από αυτούς απλά και μόνο επειδή γνωρίζουν ότι πρόκειται για άλλους ανθρώπους. Είναι λοιπόν λογικό να βγάλουμε το συμπέρασμα ότι ο κύριος λόγος που οι παίκτες-άνθρωποι είναι πιο αρεστοί δεν βρίσκεται στην εξυπνάδα που δείχνουν ή στις ικανότητές τους στο παιχνίδι, αλλά στη γνώση ότι πρόκειται για πραγματικά όντα που νιώθουν όλη τη γκάμα των συναισθημάτων.

Γι' αυτό το λόγο ενώ για πολύ καιρό το βάρος των ερευνών ήταν στο να σχεδιαστούν χαρακτήρες όλο και πιο έξυπνοι και ικανοί μέσα στο παιχνίδι, πλέον είναι κατανοητό ότι η προσπάθειες πρέπει να επικεντρωθούν στον σχεδιασμό χαρακτήρων που όχι μόνο διαθέτουν ένα μοντέλο συναισθημάτων αλλά μπορούν και να τα εκφράσουν με τρόπο που μπορεί να γίνει αντιληπτός από τους άλλους παίκτες.

Ένας κατάλληλος χώρος για να αναπτυχθεί ένα τέτοιο σύστημα είναι τα διάφορα multiplayer, on-line παιχνίδια δράσης όπως το Counter-Strike. Στα παιχνίδια αυτά η συναισθηματική εμπλοκή των παικτών αλλά και των εικονικών χαρακτήρων φαίνεται μέσω της μεταξύ τους επικοινωνίας. Το πλεονέκτημα που έχουν αυτά τα παιχνίδια είναι ότι η επικοινωνία γίνεται αποκλειστικά μέσω ανταλλαγής μηνυμάτων και μάλιστα με την μορφή chat, τα οποία δε, έχουν μερικά πολύ χρήσιμα χαρακτηριστικά ώστε να μπορούν να γίνουν κατανοητά από έναν NPC. Τα χαρακτηριστικά αυτά μας επιτρέπουν να επεξεργαστούμε το πρόβλημα χωρίς να είναι απαραίτητο να αντιμετωπιστεί σε όλο του το μέγεθος το ζήτημα της επεξεργασίας την φυσικής γλώσσας.

Η επιλογή του συγκεκριμένου χώρου λοιπόν έχει το πλεονέκτημα ότι οι παίκτες έχουν ήδη την προδιάθεση να δεχτούν ότι παίζουν απέναντι σε άλλους ανθρώπους. Έτσι, εάν ο NPC είναι ικανός απλά να μην διαψεύσει αυτή την προδιάθεση τότε οι παίκτες δεν θα έχουν κανένα λόγο να πιστέψουν κάτι αντίθετο. Επίσης το γεγονός ότι υπάρχει χώρος για ατέλειες σημαίνει ότι δεν απαιτούνται τρομερά πολύπλοκες τεχνικές για την ανάπτυξη του συστήματος.

Εάν κοιτάξουμε από κοντά την δομή μίας συζήτησης σε ένα τέτοιο περιβάλλον θα δούμε ότι έχει τα παρακάτω χαρακτηριστικά:

- Κυριαρχούν οι συχνές αλλαγές θέματος και είναι σχετικά απλό να χαθεί κανείς όσον αφορά το ποιος μιλάει σε ποιον.
- Είναι συχνό φαινόμενο ένας παίκτης να συμμετέχει παράλληλα σε περισσότερες από μία συζητήσεις ταυτόχρονα και κατ' επέκταση είναι πολύ εύκολο να χαθεί ένα ολόκληρο τμήμα της μίας συζήτησης καθώς ο παίκτης είναι απασχολημένος να απαντάει σε κάποια άλλη.
- Το λεξιλόγιο που χρησιμοποιείται είναι περιορισμένο, η γραμματική και η σύνταξη είναι προβληματικές και η ορθογραφία συνήθως υποφέρει.
- Υπάρχουν ορισμένοι βασικοί τύποι προσωπικοτήτων όσον αφορά τους παίκτες, όπως αυτός που υπερηφανεύεται για τις ικανότητές του ή αυτός που γκρινιάζει για τις συνεχείς ήττες του κτλ.

Εάν οι συζητήσεις που λαμβάνουν χώρα κατά τη διάρκεια του παιχνιδιού έχουν τα παραπάνω χαρακτηριστικά, τότε περιμένουμε να είναι επαρκώς αδόμητες ώστε να μπορούμε να πάρουμε τον απαραίτητο χρόνο για να γίνει η επεξεργασία του κειμένου της συζήτησης και να βρεθεί η κατάλληλη απάντηση. Έτσι είναι σχετικά εύκολο να σχεδιαστεί ένα σύστημα με τα παρακάτω χαρακτηριστικά:

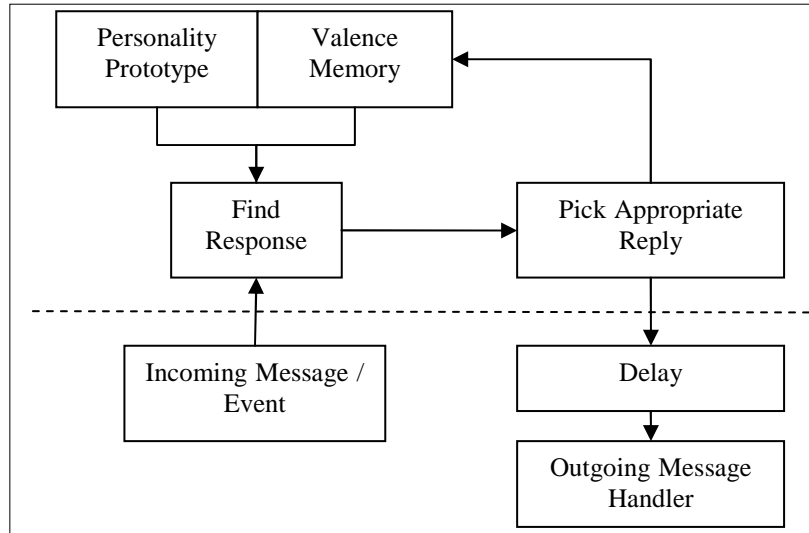
- Αναγνωρίζει τα θετικά και τα αρνητικά γεγονότα κατά τη διάρκεια του παιχνιδιού και αντιδρά με το κατάλληλο μήνυμα από ένα μεγάλο (έτοιμο) ρεπερτόριο.
- Αναγνωρίζει σε γενικές γραμμές ποιο είναι το θέμα για το οποίο συζητάνε οι παίκτες και απαντάει αντίστοιχα με την θέση που έχει ο κάθε παίκτης (φίλος, αντίπαλος κτλ).
- Αγνοεί ότι είναι πολύ περίπλοκο για να μπορεί να το επεξεργαστεί σωστά. Έχει δηλαδή την ευφυΐα να μην κάνει κινήσεις με τις οποίες να κινδυνεύει να τον ξεσκεπάσουν.

Η απλότητα της συζήτησης λοιπόν, βοηθάει τους NPCs να κρύβονται καθώς οι άνθρωποι παίκτες είναι πολύ πιθανόν να μην υποπτευθούν αν ένας άλλος παίκτης αγνοήσει κάποια σχόλια προς αυτόν ή αν πετάξει κάποιο σχόλιο χωρίς πολύ νόημα. Επιπλέον τα στερεότυπα που περιγράφουν τη συμπεριφορά των διαφόρων τύπων παικτών διευκολύνουν την όλη διαδικασία καθώς παρέχουν έναν ξεκάθαρο οδηγό για την διαδικασία ανάπτυξης.

Στο πείραμα που περιγράφεται εδώ, η ερευνητική ομάδα των Zubek και Khoo [18], ανέπτυξε ειδικούς NPCs για το παιχνίδι Counter Strike. Πρόκειται για ένα παιχνίδι με σύντομες αποστολές που προσφέρεται για συνομιλίες μεταξύ των παικτών όταν αυτοί έχουν βγει από το παιχνίδι. Επίσης, χάριν στο γεγονός ότι είναι ένα ιδιαίτερα δημοφιλές παιχνίδι, υπάρχει πληθώρα νέων που μπορούν να μελετηθούν

ώστε να εξαχθούν τα απαραίτητα ψυχολογικά δεδομένα και να κατηγοριοποιηθούν τα είδη των παικτών.

Το σύστημα που χρησιμοποιούν οι χαρακτήρες για να παράγουν λεκτικές αντιδράσεις στα διάφορα γεγονότα φαίνεται στο παρακάτω Σχήμα 4 [18].



Σχήμα 4 – Σύστημα παραγωγής αντιδράσεων των χαρακτήρων

Το σύστημα αυτό βασίζεται στο γνωστό σύστημα Eliza [17]. Τα εισερχόμενα μηνύματα από τον χώρο της συζήτησης περνούν στην Eliza, η οποία είναι εφοδιασμένη με ένα σετ από στερεοτυπικές προσωπικότητες (γκρινιάρης, καυχητής, κλπ) η κάθε μία από τις οποίες διαθέτει ανιχνευτές για γεγονότα και κομμάτια συζήτησης που θα είναι ενδιαφέροντα για αυτήν. Σε κάθε αποστολή ο NPC υλοποιεί και μία διαφορετική προσωπικότητα και διαθέτει μία λίστα συμπεριφορών με διαφορετικές προτεραιότητες. Με τη χρήση κανονικών εκφράσεων κάθε εισερχόμενο μήνυμα αντιστοιχίζεται με κάποια συμπεριφορά, ενώ τα μηνύματα που μοιάζουν σημαντικά αλλά δεν μπορούν να προκαλέσουν μία σαφή συμπεριφορά αντιστοιχίζονται με μερικά πάγια ασαφή και γενικόλογα μηνύματα.

Φυσικά δεν περνιέται κάθε μήνυμα στο σύστημα καθώς είναι λογικό ότι όταν ένας παίκτης βρίσκεται μέσα στον εικονικό χρόνο του παιχνιδιού δεν αναλώνεται σε συζητήσεις αλλά κατά κύριο λόγο αφοσιώνεται στο παιχνίδι. Δεν θα ήταν καθόλου φυσιολογική συμπεριφορά από έναν παίκτη να συμμετέχει σε όλες τις συζητήσεις και παράλληλα να μπορεί να συνεχίζει να παίζει. Γι' αυτόν εξ' άλλου τον λόγο, όπως φαίνεται και από το σχήμα 4, εισάγεται και μία τεχνητή χρονοκαθυστέρηση στις απαντήσεις του NPC αφού δεν θέλουμε να απαντάει πιο γρήγορα απ' ότι θα ήταν ανθρωπίνως δυνατόν.

Τέλος ένα πολύ ενδιαφέρον χαρακτηριστικό του σχεδιασμού του συστήματος είναι ότι οι αποκρίσεις του χαρακτήρα διαμορφώνονται και από τον τρόπο με τον οποίο αντιμετωπίζει τον κάθε παίκτη, ο οποίος δεν είναι κοινός για όλους. Έτσι, δίνει διαφορετικές απαντήσεις στους συμπαίκτες απ' ότι στους αντιπάλους, όπως είναι άλλωστε και αναμενόμενο από έναν χαρακτήρα που θέλει να υιοθετήσει ανθρώπινη συμπεριφορά.

Τα αποτελέσματα του πειράματος ήταν σίγουρα θετικά, μιας και σε γενικές γραμμές οι NPCs κατάφεραν να ξεγελάσουν τους ανθρώπους για ένα αρκετά μεγάλο χρονικό διάστημα, δεδομένου πάντα ότι πρόκειται για ένα αρκετά απλοϊκό σύστημα.

Τα προβλήματα που διαπιστώθηκαν είχαν να κάνουν με την ικανότητα κατανόησης συγκεκριμένων κωδίκων που χρησιμοποιούν οι παίκτες και η εξαγωγή πληροφοριών από αυτούς (για παράδειγμα η χρήση συντομεύσεων αντί των ολόκληρων ονομάτων) και επίσης με το δυσάρεστο γεγονός ότι οι συνομιλίες που παράγει η Eliza είναι σε μεγάλο βαθμό αλλοπρόσαλλες ώστε να μπορούν να γίνονται πιστευτές για μεγάλο χρονικό διάστημα. Οι NPCs είχαν το χαρακτηριστικό ότι άλλαζαν πολύ συχνά θέμα, χωρίς να διατηρούν κάποια ιστορία, κάτι το οποίο μάλλον είναι απαραίτητο για να είναι τελικά πιστευτοί οι χαρακτήρες μας.

Το πείραμα αυτό λοιπόν επαληθεύει όλα όσα έχουμε μέχρι τώρα αναφέρει όσον αφορά την ανάγκη να έχουν οι χαρακτήρες μας συναισθήματα για να είναι πιστευτοί αλλά επίσης να μπορούν να εκφράσουν τα συναισθήματά τους αυτά και κυρίως με τρόπο που να γίνεται αντιληπτός από τους ανθρώπους μιας και αυτοί είναι οι τελικοί αποδέκτες όλων των ευφάνταστων σχεδιασμών που κάνουμε.

2.4 Απλοποιημένο μοντέλο της εφαρμογής

Έχοντας περιγράψει το θεωρητικό υπόβαθρο που καλύπτει τον χώρο του σχεδιασμού πειστικών Non Player Characters μπορούμε πια με λίγα λόγια να μιλήσουμε για τους στόχους που θέσαμε στην δική μας προσέγγιση σε αυτό το πολύπλευρο ζήτημα.

Ο γενικότερος στόχος μας είναι ο σχεδιασμός και η εισαγωγή ενός NPC σε έναν εικονικό κόσμο έτσι ώστε ο χαρακτήρας αυτός να έχει κάποια από τα βασικά χαρακτηριστικά που περιγράψαμε ώστε να είναι ρεαλιστικός και πειστικός.

Καταρχάς ο πρώτος στόχος που θέσαμε είναι να έχει ο παίκτης ένα βασικό μοντέλο προσαρμογής. Αυτό σημαίνει ότι έχουμε έναν NPC ο οποίος παρατηρεί τον κόσμο και βγάζει κάποια συμπεράσματα σχετικά με αυτά που συμβαίνουν γύρω του, και στη συνέχεια τροποποιεί την συμπεριφορά του ώστε να προσαρμόζεται με τα συμπεράσματα αυτά, πάντα στα πλαίσια ενός στόχου που έχει να επιτεύξει.

Δεύτερον, ο εικονικός χαρακτήρας χαρακτηρίζεται από μία συγκεκριμένη διάθεση, η οποία αλλάζει ανάλογα με τα γεγονότα που λαμβάνουν χώρα, και η οποία εκφράζεται μέσω μίας αλλαγής στον τρόπο με τον οποίο κινείται ο παίκτης στον χώρο. Αυτό το χαρακτηριστικό το έχει και ο ίδιος ο παίκτης καθώς ακόμη ένα σημείο το οποίο έχει ενδιαφέρον είναι να απεικονίζεται η συναισθηματική κατάσταση των εικονικών χαρακτήρων αλλά και των ανθρώπων παικτών όχι μόνο με το πρόσωπό τους (κάτι που δεν εφαρμόζεται σε γενικές γραμμές στα σύγχρονα παιχνίδια) αλλά και μέσω των κινήσεων του σώματός τους (κάτι το οποίο μπορεί πιο εύκολα αν υλοποιηθεί σε ένα παιχνίδι).

Τρίτον, θελήσαμε να δείξουμε την διαφορετική συμπεριφορά που αναπτύσσουν οι NPCs ανάλογα με τον χαρακτήρα που υλοποιεί ο καθένας και ανάλογα με τους στόχους που έχει θέσει.

Τέλος, εξετάζουμε και το ζήτημα των διαφορετικών components που μπορούν να συνεργάζονται στα πλαίσια του multiplayer περιβάλλοντος στο οποίο καλούνται συνήθως κάτι τέτοιοι χαρακτήρες να δραστηριοποιούνται.

3 Η μηχανή Torque Game Engine

3.1 Η ιστορία της Torque Game Engine

Για την δημιουργία του εικονικού μας κόσμου χρησιμοποιήθηκε η μηχανή Torque Game Engine (TGE). Η TGE είναι μία πλήρης πλατφόρμα παραγωγής τρισδιάστατων γραφικών για παιχνίδια αν και οι δυνατότητες που παρέχει στον προγραμματιστή είναι τόσο ευρείες που μπορεί να χρησιμοποιηθεί πρακτικά για μεγάλη γκάμα εμπορικών αλλά και εκπαιδευτικών εφαρμογών. Είναι γραμμένη στην αντικειμενοστραφή γλώσσα C++ και υπάρχει η δυνατότητα παραμετροποίησης της αφού ο πηγαίος κώδικας είναι γνωστός. Η υλοποίηση του παιχνιδιού ωστόσο δεν γίνεται με επέμβαση στον πηγαίο κώδικα (ο οποίος αντιπροσωπεύει τις βασικές δομές που απαιτούνται για την ανάπτυξη της εκάστοτε εφαρμογής) αλλά με χρήση της γλώσσας Torque Script.



Η TGE είναι μία εξελιγμένη έκδοση της μηχανής παραγωγής 3D γραφικών που αναπτύχθηκε από την Dynamix για το παιχνίδι Tribes 2 (2001). Έκτοτε η TGE είναι διαθέσιμη από την garagegames σε επαγγελματίες ή μη δημιουργούς παιχνιδιών και ήδη κυκλοφορούν αρκετοί εμπορικοί τίτλοι παιχνιδιών που έχουν υλοποιηθεί στο περιβάλλον της TGE (Marble Blast Gold, TubeTwist, Blockland κ.α.)

3.2 Γενικά Χαρακτηριστικά

3.2.1 Υποστήριξη πολλών παικτών – Δυνατότητα δικτύωσης

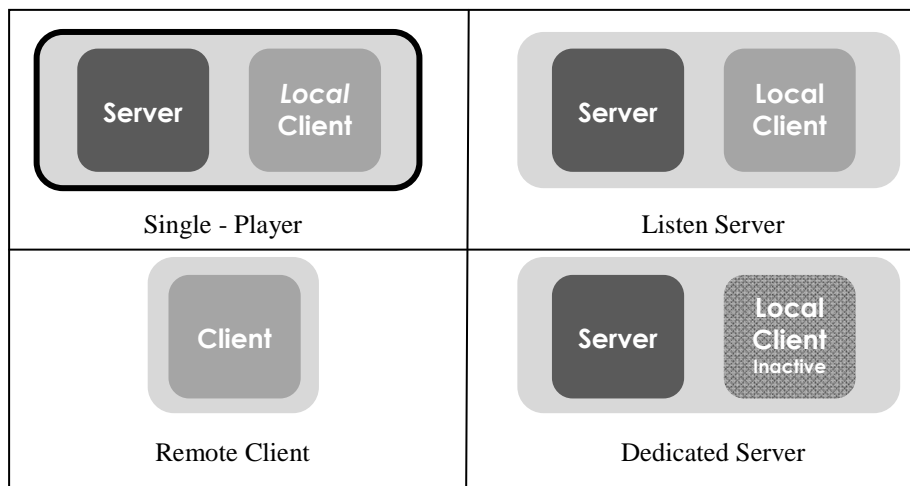
Η TGE είναι υλοποιημένη στο πνεύμα των multiplayer εφαρμογών. Έτσι αν και δεν είναι δεσμευτικό παρέχει ολοκληρωμένη υποστήριξη για αρχιτεκτονικές τύπου πελάτη / εξυπηρετητή (client-server applications). Αυτό σημαίνει ότι η βασική δομή της μηχανής είναι χωρισμένη σε δύο ανεξάρτητα τμήματα όπου το ένα παίζει το ρόλο του εξυπηρετητή – μονάδα ελέγχου, και το άλλο είναι η εξαρτώμενη μονάδα πελάτη (η οποία και δεν μπορεί να υπάρξει χωρίς τον αντίστοιχο εξυπηρετητή).

Κατα την φάση εκτέλεσης ο πελάτης και ο εξυπηρετητής μπορούν είτε να συνυπάρχουν στο ίδιο εκτελέσιμο αρχείο είτε να εκτελούνται χωριστά σε διαφορετικούς υπολογιστές (Σχήμα 5). Ο εξυπηρετητής ανά πάσα στιγμή γνωρίζει πόσοι και ποιοι πελάτες είναι συνδεδεμένοι σε αυτόν (ανάλογα με το είδος της σύνδεσης που έχει επιλεγεί – Σχήμα 6) ενώ οι εκάστοτε πελάτες μπορεί να αντιλαμβάνονται την ύπαρξη του άλλου χωρίς όμως αυτό να είναι απαραίτητο.

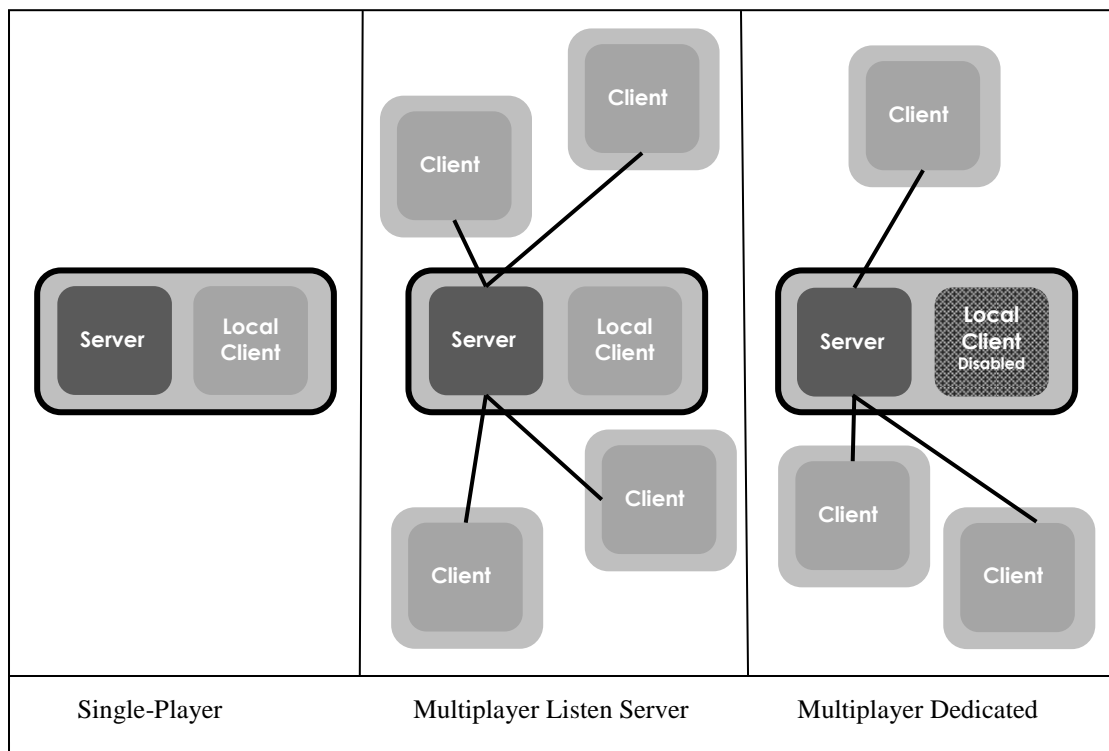
Είναι φανερό λοιπόν ότι ένα βασικό δομικό χαρακτηριστικό της TGE είναι η δυνατότητα δικτύωσης και μάλιστα είναι σαφής η ανάγκη για γρήγορη επικοινωνία με τις ελάχιστες δυνατές απαιτήσεις σε εύρος ζώνης. Η TGE έχει σχεδιαστεί ώστε να καλύπτει αυτές τις ανάγκες στον μεγαλύτερο δυνατό βαθμό λαμβάνοντας υπόψιν ότι μπορεί να υποστηρίξει μέχρι και 128 πελάτες ανά εξυπηρετητή.

Μερικές στρατηγικές για την μείωση του απαιτούμενου εύρους ζώνης που απαιτούνται από την μηχανή είναι:

- Αποστέλλονται τα ελάχιστα δυνατά bits για μία συγκεκριμένη πληροφορία
- Αποστέλλεται αποκλειστικά το κομμάτι της κατάστασης ενός αντικείμενου που έχει αλλάξει και όχι ολόκληρο το αντικείμενο.
- Στέλνει τις σημαντικές ενημερώσεις στον πελάτη με μεγαλύτερο ρυθμό απ' ότι αυτές που ενδεχομένως να μην είναι τόσο σημαντικές.
- Χρησιμοποιεί έναν κατάλογο από συχνά μεταδιδόμενα μηνύματα ώστε να μπορεί να μεταδίδεται μόνο ο αντίστοιχος κωδικός αριθμός του μηνύματος και όχι ολόκληρη η συμβολοακολουθία ξανά και ξανά.



Σχήμα 5 – Διαφορετικές αρχιτεκτονικές εφαρμογών με το μοντέλο πελάτη-εξυπηρετητή.



Σχήμα 6 – Διαφορετικοί τρόποι διασύνδεσης στην αρχιτεκτονική πελάτη-εξυπηρετητή

3.2.2 Παραγωγή γραφικών

Η TGE είναι κατά βάση μία μηχανή παραγωγής γραφικών. Αυτό σημαίνει ότι είναι υπεύθυνη για την δημιουργία του εικονικού κόσμου, τον φωτισμό του, την εισαγωγή των διαφόρων αντικειμένων και φυσικά την δημιουργία των παικτών καθώς και την κίνησή τους. Για να τα κάνει όλα αυτά η TGE χρησιμοποιεί το OpenGL API μαζί και με μία βιβλιοθήκη που επεκτείνει τις δυνατότητές του.

Όλα τα αντικείμενα που εξαφανίζονται σε έναν εικονικό κόσμο της TGE, είτε πρόκειται για ένα κτήριο είτε πρόκειται για ένα απλό κουτί, είτε πρόκειται για το avatar του ίδιου του παίκτη, πρέπει να φορτωθούν από κάποιο κατάλληλο αρχείο. Η TGE υποστηρίζει δύο ειδών αρχεία δεδομένου ότι υπάρχουν διαφορετικές ανάγκες όσον αφορά την ποιότητα των γραφικών γιατί δεν πρέπει να ξεχνάμε ότι η TGE καλείται να δημιουργεί σε πραγματικό χρόνο έναν τεράστιο αριθμό πολυγώνων και πρέπει να μπορεί να το κάνει γρήγορα και αποτελεσματικά.

- Αρχεία τύπου .dts

Τα αρχικά dts προκύπτουν από τα τις λέξεις Dynamix Threespace Shape και αφορούν μοντέλα που δημιουργούνται από polygon editors. Ένα αντικείμενο που περιγράφεται σε ένα αρχείο .dts μπορεί να έχει skeletal animations, πολλαπλά και κινούμενα skins, visibility animations, διάφορα επίπεδα λεπτομέρειας (ανάλογα με το πόσα pixel καταλαμβάνουν στην οθόνη κάθε στιγμή), διαφανείς περιοχές, πολλαπλά collision boxes κλπ.

Με αρχεία .dts συνήθως περιγράφονται τα διάφορα μικρά αντικείμενα του κόσμου τα οποία μπορεί να βρίσκονται πάνω σε έναν παίκτη ή ένα όχημα (και άρα να ακολουθούν τις κινήσεις του) αλλά και τα οποία φυσικά μπορεί να έχουν τα δικά τους animations. Επίσης με .dts αρχεία περιγράφονται τα μοντέλα των παικτών. Τα μοντέλα αυτά είναι και το πραγματικά δυνατό σημείο του dts καθώς μπορούν να έχουν animations βασισμένα στην δομή του σκελετού του μοντέλου και επίσης μπορούν να συνδυάσουν περισσότερα από ένα animations (με την χρήση κατάλληλων scripts) ώστε να σχηματίσουν πιο περίπλοκες χειρονομίες. Τέλος με dts μοντελοποιούνται και τα οχήματα, τα οποία μπορούν να συνδυαστούν πχ με διάφορα όπλα και φυσικά τους ίδιους τους παίκτες.

- Αρχεία τύπου .dif

Τα αρχεία αυτά έρχονται να καλύψουν τα κενά που αφήνουν τα .dts αρχεία. Οι μεγάλες δομές και ειδικά οτιδήποτε είναι αρκετά μεγάλο ώστε να μπορεί ο παίκτης να βρεθεί μέσα σε αυτό μοντελοποιείται με ένα αρχείο .dif. Αυτό γίνεται γιατί με τα μεγάλα αντικείμενα υπάρχει η ανάγκη συνήθως να φαίνεται στην οθόνη μόνο ένα τμήμα τους και άρα πρέπει να υπάρχουν οι κατάλληλες δομές ώστε αυτό να μπορεί να γίνει χωρίς να γίνεται ενδεχομένως render ολόκληρο το σχήμα, γεγονός που θα οδηγούσε σε σπατάλη υπολογιστικής ισχύος.

Επίσης τα αντικείμενα για τα οποία απαιτείται πολύ λεπτομερειακή ανίχνευση σύγκρουσης το dts μοντέλο δεν είναι κατάλληλο καθώς κάνει μια αρκετά απλοϊκή προσέγγιση και δεν μπορεί να περιγράψει με ικανοποιητική ακρίβεια για παράδειγμα το περπάτημα ενός παίκτη πάνω σε ένα πολύ μεγαλύτερο από αυτόν αντικείμενο. Έτσι και εδώ χρησιμοποιούνται αρχεία τύπου dif.

Τέλος τα αρχεία .dif προσφέρουν έναν πιο αποδοτικό τρόπο ώστε να υπολογιστεί η επίδραση στον φωτισμό του γύρω χώρου εξαιτίας του αντικειμένου.

Η TGE μας δίνει επίσης τη δυνατότητα να σχεδιάσουμε τόσο τα επίπεδα (missions) του παιχνιδιού όσο και κάθε είδους γραφικής επιφάνειας αλληλεπίδρασης (graphical user interface - GUI) για αλληλεπίδραση με τον χρήστη. Για να γίνεται αυτό εύκολα και γρήγορα η TGE συνοδεύεται από διάφορους ισχυρούς editors που κάνουν την δουλειά να γίνεται με τη χρήση γραφικού περιβάλλοντος και όχι αποκλειστικά μέσω Torque Script.

- **Mission Editor**

Ο Mission Editor περιλαμβάνει μία ομάδα από 8 επιμέρους εργαλεία με τα οποία μπορούμε να τροποποιήσουμε διάφορα στοιχεία του εικονικού κόσμου (mission). Μπορούμε τόσο να δημιουργήσουμε ένα νέο περιβάλλον, όσο και να τροποποιήσουμε ένα ήδη υπάρχον χωρίς να χρειάζεται να καταφύγουμε στη δημιουργία νέων script.

World editor

Μέσω του World editor μπορούμε να μετακινήσουμε, να περιστρέψουμε και γενικά να επεξεργαστούμε ήδη υπάρχοντα αντικείμενα.

World Editor Inspector

Πέρα από τις λειτουργίες του World Editor, ο World Editor Inspector μας επιτρέπει να δούμε και να τροποποιήσουμε τις ιδιότητες των διαφόρων αντικειμένων της κάθε αποστολής.

World Editor Creator

Με τον World Editor Creator μπορούμε να τοποθετήσουμε νέα αντικείμενα σε όποια θέση του χάρτη θέλουμε, μέσα στην τρέχουσα αποστολή.

Mission Area Editor

Με το εργαλείο αυτό μπορούμε να καθορίσουμε τα όρια της ενεργούς περιοχής του χάρτη καθώς και να καθρεφτίσουμε το έδαφος (terrain).

Terrain Editor

Μας επιτρέπει να καθορίσουμε το ύψος του εδάφους του χάρτη με το χέρι με τη χρήση «πινέλου».

Terrain Terraform Editor

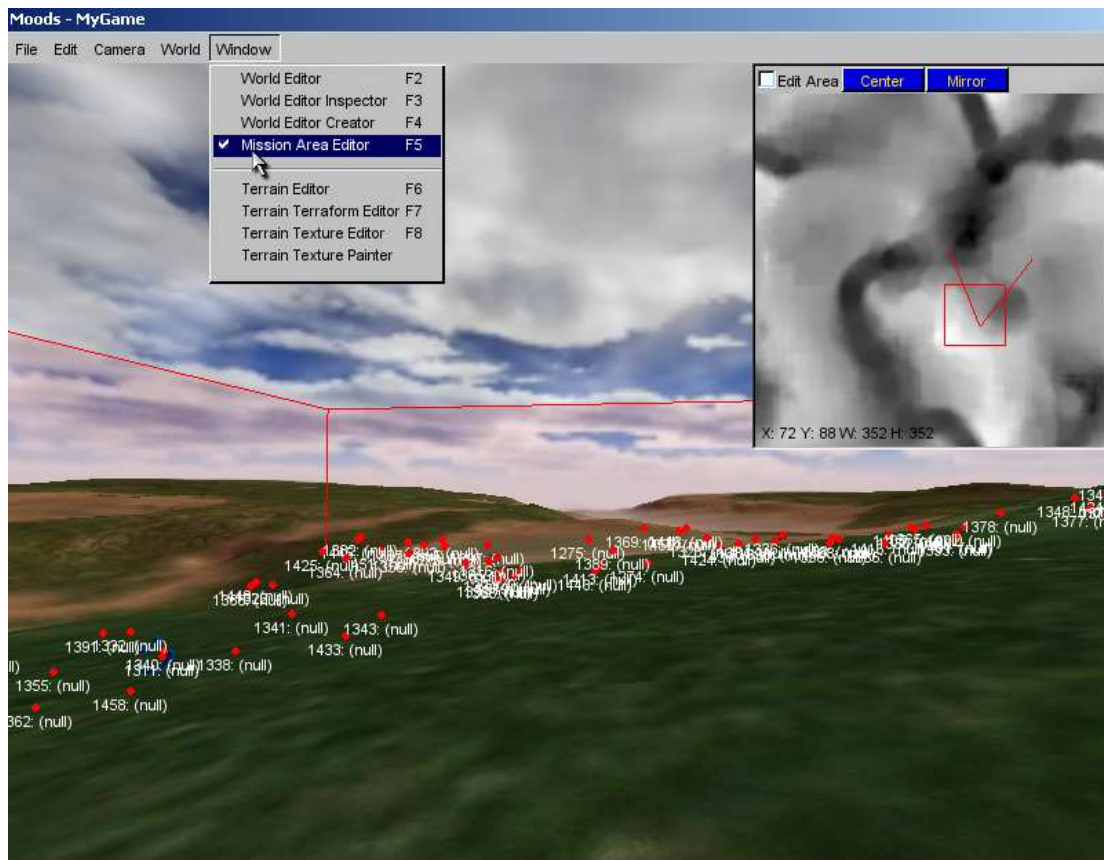
Επιπλέον από τις λειτουργίες του Terrain Editor, μας παρέχει τη δυνατότητα να φορτώσουμε εικόνες ως αρχεία terrain και να δώσουμε έτσι ότι σχέδιο θέλουμε στο έδαφος καθώς επίσης να εφαρμόσουμε διάφορα αλγοριθμικά φίλτρα στο χάρτη (π.χ. για να γίνει πιο ομαλός).

Terrain Texture Editor

Μας επιτρέπει να φορτώσουμε bitmaps για τον χρωματισμό του χάρτη και μας παρέχει διάφορους αλγορίθμους για την επεξεργασία και το συνδυασμό των διαφόρων υφών (textures).

Terrain Texture Painter

Λειτουργεί σαν μία παλέτα απ' όπου μπορούμε να επιλέγουμε την υφή με την οποία θέλουμε κάθε φορά να εργαστούμε.



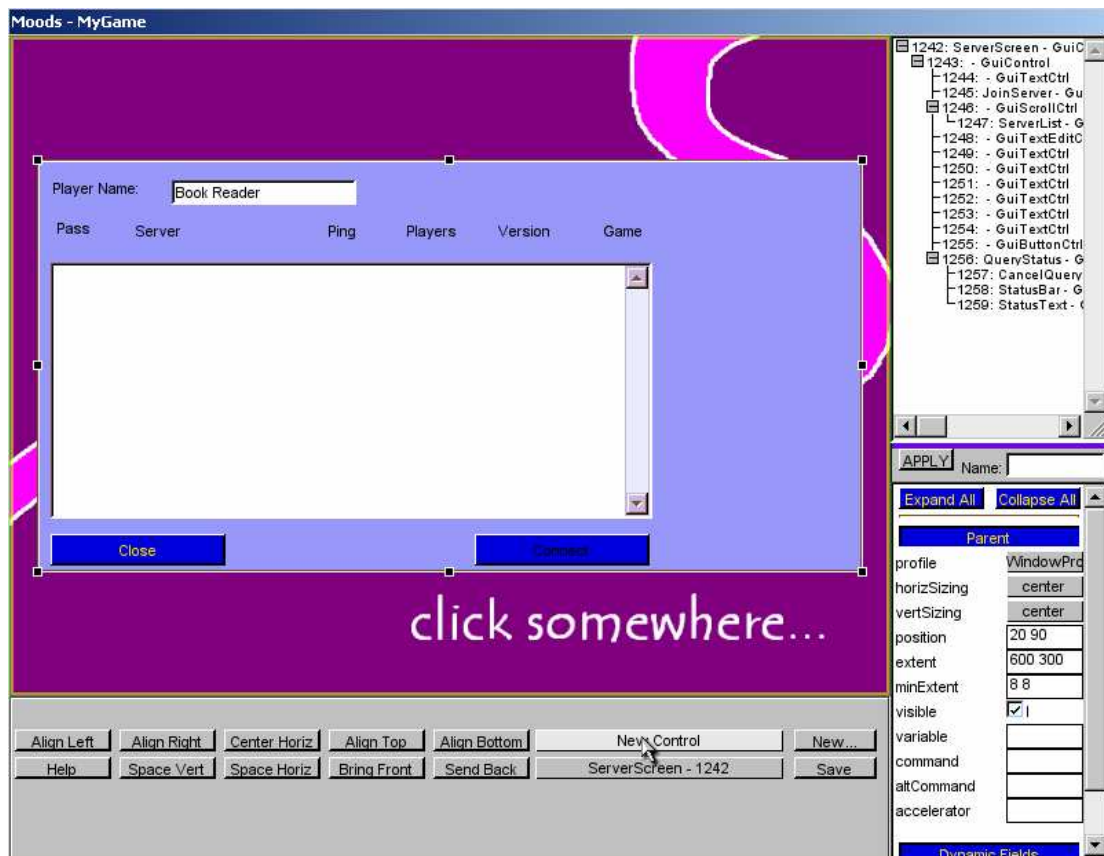
Σχήμα 7 – Χαρακτηριστική όψη του Mission Editor όπου βλέπουμε τον εικονικό κόσμο, τα όρια της mission area και τα id των αντικειμένων που υπάρχουν.

- **Gui Editor**

Όλα τα παιχνίδια χρειάζονται κάποιου είδους διεπαφή με τον χρήστη τόσο για να τον ενημερώσουν είτε για αλλαγές, είτε για τις διάφορες επιλογές του, όσο και για να μπορούν να δέχονται την είσοδό του, όποτε αυτό χρειάζεται. Για παράδειγμα σε λειτουργίες ανταλλαγής μηνυμάτων σε ένα περιβάλλον πολλών παικτών, ο χρήστης πρέπει να μπορεί να βλέπει να μηνύματα που του στέλνουν οι άλλοι παίκτες αλλά και να πληκτρολογεί τα δικά του μηνύματα. Η σημασία λοιπόν των GUIs σε ένα παιχνίδι είναι ιδιαίτερα μεγάλη καθώς είναι ένα χαρακτηριστικό που όλοι οι παίκτες χρησιμοποιούν κάθε φορά που θα ανοίξουν την εφαρμογή.

Η TGE έχει έναν ενσωματωμένο GUI Editor που είναι ιδιαίτερα δυνατός και ευέλικτος. Επιτρέπει απόλυτο έλεγχο στην λειτουργικότητα των interfaces και παρέχει απόλυτη ελευθερία σχετικά με τις σχεδιαστικές επιλογές, έτσι που πρακτικά μπορούμε να δημιουργήσουμε ό,τι φανταστούμε.

Μπορούμε να δημιουργήσουμε τόσο Head Up Displays (HUDs), δηλαδή οτιδήποτε βλέπει ο χρήστης την ώρα που τρέχει η εφαρμογή, όσο και Menus, Splash Screens, Message Boxes κτλ. Μέσα από τον Gui Editor της Torque υπάρχει έτοιμη μία μεγάλη γκάμα controls έτοιμων προς χρήση, όπως buttons, text fields, radio buttons, labels κτλ για τα οποία μπορούμε να ορίσουμε ιεραρχία, λειτουργικότητα και φυσικά θέση και ευθυγράμμιση στον χώρο που καταλαμβάνει το gui που φτιάχνουμε.



Σχήμα 8 – Όψη του Gui Editor όπου φαίνονται τα διάφορα αντικείμενα του μενού, οι ιδιότητές και η ιεραρχία τους καθώς και τα κουμπιά με τις διάφορες ενέργειες που μπορούμε να κάνουμε.

3.2.3 Κονσόλα και TorqueScript

Η βιβλιοθήκη console είναι η βάση για όλες τις εφαρμογές που αναπτύσσονται στην TGE. Η κονσόλα περιέχει έναν μεταγλωττιστή και έναν μεταφραστή και γενικά όλα τα GUIs, τα αντικείμενα και η λογική της εκάστοτε εφαρμογές περνάνε μέσα από αυτήν. Η γλώσσα που χρησιμοποιείται όπως αναφέρθηκε και προηγουμένως είναι η Torque Script η οποία μοιάζει με C++ χωρίς όμως να έχει τύπους δεδομένων και με διάφορα παραπάνω χαρακτηριστικά που βοηθούν την ανάπτυξη ενός παιχνιδιού. Console scripts μπορούν να φορτωθούν στην TGE τόσο μέσω εντολών της κονσόλας όσο και κατευθείαν από αρχεία.

Τι είναι όμως αυτό που κάνει τη δυνατότητα scripting σημαντική; Καταρχάς ως scripting ορίζεται η χρήση προϋπάρχοντων στοιχείων (engine components) για να εκπληρώσουμε ένα νέο στόχο. Με λίγα λόγια χρησιμοποιούμε την scripting γλώσσα για να έχουμε πρόσβαση στα χαρακτηριστικά (features) της μηχανής και για να τα χρησιμοποιήσουμε ώστε να δημιουργήσουμε την κατάλληλη εμπειρία παιχνιδιού.

Γενικά μία scripting γλώσσα μεταφράζεται, δεν μεταγλωττίζεται, και αυτό ενδεχομένως κάνει τις εργασίες που γίνονται μέσω script να είναι πιο αργές από αυτές που υλοποιούνται με κώδικα αλλά είναι σαφές ότι κερδίζουμε σε ευελιξία και ευκολία χρήσης. Ακριβώς επειδή οι scripting γλώσσες μας επιτρέπουν να αλλάζουμε την εφαρμογή χωρίς να χρειάζεται μεταγλώττιση, είμαστε σε θέση να παράγουμε, να ελέγχουμε και φυσικά επιδιορθώσουμε κώδικα εύκολα, γεγονός το οποίο επιταχύνει την διαδικασία ανάπτυξης.

Επίσης οι γλώσσες αυτές επιτρέπουν στον προγραμματιστή να ξεχάσει για λίγο τους αυστηρούς κανόνες και να παράγει κώδικα χωρίς να ενδιαφέρεται για τύπους δεδομένων και διαχείριση μνήμης. Αυτό βέβαια μπορεί να έχει το πλεονέκτημα ότι απλοποιεί πολλές διαδικασίες αλλά από την άλλη επιτρέπει να ξεφύγουν λάθη που ένας μεταγλωττιστής σίγουρα θα εντόπιζε.

Ακόμη ένα πλεονέκτημα του scripting είναι ότι προσφέρεται για την διαδικασία της αποσφαλμάτωσης καθώς όχι μόνο μπορούμε να εντοπίσουμε και να διορθώσουμε τυχόν προβλήματα χωρίς καν να χρειαστεί να γίνει επανεκκίνηση της εφαρμογής αλλά επίσης μπορούμε να δημιουργήσουμε γρήγορα μονάδες ελέγχου με στόχο π.χ. ένα πολύ συγκεκριμένο τμήμα κώδικα.

Τέλος η δυνατότητα να υπάρχουν χαρακτηριστικά της εφαρμογής (που αφορούν τις διεπαφές ή το gameplay) που μπορούν να αλλάξουν μέσω scripts είναι πολύ σημαντική όταν θέλουμε να τεστάρουμε εύκολα τις διάφορες παραμέτρους και τους συνδυασμούς τους κατά τη διάρκεια της διαδικασίας παραγωγής. Αυτή η εναλλακτική προσφέρει επίσης στους τελικούς χρήστες την δυνατότητα να δημιουργούν τις δικές τους τροποποιήσεις, κάτι το οποίο είναι ιδιαίτερα δημοφιλές.

Είναι λοιπόν σαφές ότι μία script γλώσσα σε συνδυασμό με την ίδια τη μηχανή μπορεί να μας βοηθήσει να δημιουργήσουμε μια επεκτάσιμη εφαρμογή και τρόπο απλό και συνάμα ελεγχόμενο. Ποια είναι όμως τα εργαλεία που πρέπει να μας παρέχει μία scripting γλώσσα; Καταρχάς είναι απαραίτητο να υπάρχουν όλες οι κλασσικές προγραμματιστικές δομές όπως ισχυροί τύποι δεδομένων, οι βασικές πράξεις (πρόσθεση κλπ), οι δομές ελέγχου (if-then-else, for, while) και φυσικά η δυνατότητα χρήσης υποπρογραμμάτων και εξωτερικών αρχείων. Επίσης μιας και πρόκειται για ένα εργαλείο που δεν στέκεται μόνο του, είναι απαραίτητη η δυνατότητα πρόσβασης στις δομές της ίδιας της μηχανής. Το scripting σύστημα πρέπει να επιτρέπει την πρόσβαση στο σύστημα παραγωγής γραφικών, ήχου, τεχνητής νοημοσύνης (AI) και εισόδου/εξόδου (I/O) καθώς επίσης πρέπει να καθιστά δυνατή τη δημιουργία και την διαγραφή αντικειμένων και τον ορισμό νέων συναρτήσεων. Άλλα επιθυμητά χαρακτηριστικά είναι η υιοθέτηση του αντικειμενοστραφούς μοντέλου προγραμματισμού μαζί με όλα τα θετικά χαρακτηριστικά του όπως η κληρονομικότητα, ο πολυμορφισμός κτλ, η δυνατότητα να φορτώνονται τμήματα κώδικα σε πραγματικό χρόνο (δυναμικά), με αποτέλεσμα να μπορούμε να τροποποιούμε άμεσα την λειτουργικότητα της εφαρμογής (πάντα χωρίς επανεκκίνηση) και τέλος η δυνατότητα τα scripts να μεταγλωττίζονται σε byte-code ο οποίος στην συνέχεια εκτελείται από μία virtual machine. Το τελευταίο αυτό χαρακτηριστικό αυξάνει ακόμα περισσότερο την απόδοση του συστήματος καθώς σε γενικές γραμμές ο byte-code εκτελείται γρηγορότερα από τον μεταφραζόμενο κώδικα.

Η Torque Script λοιπόν, συνδυάζει πολλά από τα παραπάνω χαρακτηριστικά, με τα κύρια γνωρίσματά της να είναι:

- υποστηρίζει όλους τους γνωστούς τελεστές
- τές καθώς και μερικούς πιο εξελιγμένους σε σχέση με τις κλασικές γλώσσες προγραμματισμού (πχ. για απ'ευθείας χειρισμό συμβολοακολουθιών).
- διαθέτει ένα σύνολο επιπρόσθετων συναρτήσεων (μαθηματικές, χειρισμού συμβολοακολουθιών κτλ) που συμπληρώνουν τις λειτουργίες των τελεστών.
- Διαθέτει όλες τις γνωστές δομές ελέγχου (if-then-else, for, while, switch).
- Επιτρέπει τον ορισμό νέων συναρτήσεων, με την επιλογή επιστροφής ή όχι τιμής.
- Προσφέρει κληρονομικότητα και πολυμορφισμό
- Επιτρέπει την δυναμική φόρτωση (και το αντίστροφο) στην μνήμη συναρτήσεων, ανάλογα με τις εκάστοτε ανάγκες.
- Παρέχει την δυνατότητα ορισμού namespaces (κατά το παράδειγμα της C++). Τα namespaces χρησιμοποιούνται για να αποφεύγονται τα προβλήματα λόγω ύπαρξης συναρτήσεων με διαφορετικά ονόματα, αλλά φυσικά διαφορετική λειτουργικότητα.
- Μεταγλωττίζει και εκτελεί byte-code με αποτέλεσμα την επιτάχυνση της διαδικασίας εκτέλεσης αλλά και την ύπαρξη ενός σημείου όπου μπορεί να γίνει ανίχνευση λαθών.

Βέβαια η χρήση Torque Script δεν είναι απαραίτητη καθώς είναι διαθέσιμος και ο πηγαίος κώδικας της TGE αλλά παρόλα αυτά είναι ένας γρήγορος και κυρίως εύκολος τρόπος για να κάνουμε αλλαγές που δεν επηρεάζουν όλο το σύστημα, χωρίς να χρειάζεται επανεκκίνηση, καθώς τα scripts μεταφράζονται σε πραγματικό χρόνο.

3.2.4 Προσομοίωση γεγονότων

Η TGE είναι χτισμένη γύρω από έναν προσομοιωτή που ενεργοποιείται από διάφορα γεγονότα που συμβαίνουν στον εικονικό κόσμο, χωριστά για τον πελάτη και τον εξυπηρετητή. Τα γεγονότα αυτά χωρίζονται σε διάφορες κατηγορίες και επεξεργάζονται από την μηχανή με την σειρά με την οποία συμβαίνουν και μπορεί να είναι από την κίνηση του ποντικιού του χρήστη μέχρι την ανίχνευση μίας σύγκρουσης.

Ο προγραμματιστής έχει μεγάλη ευχέρεια στην χρήση των γεγονότων καθώς μπορεί να προγραμματίσει κάποιο να συμβεί μετά από ένα χρονικό περιθώριο (και έτσι να πυροδοτήσει κάποια επιθυμητή εξέλιξη), μπορεί να εξετάσει ανά πάσα στιγμή ποια γεγονότα εκκρεμούν και τέλος να ακυρώσει ένα από αυτά. Η δυνατότητες αυτές δίνουν έναν ιδιαίτερα δυναμικό χαρακτήρα στον τρόπο που καθορίζονται οι εξελίξεις και οι αλλαγές μέσα στον εικονικό κόσμο.

Ένα χαρακτηριστικό της TGE που εμπλουτίζει ακόμα περισσότερο τις δυνατότητές της είναι η ύπαρξη ενός σετ από συναρτήσεις οι οποίες καλούνται απευθείας από την μηχανή σαν αντίδραση στα διάφορα γεγονότα που συμβαίνουν στον εικονικό κόσμο. Αυτές οι συναρτήσεις απαντούν ως callbacks και είναι ένα βασικό στοιχείο για να υπάρχει η έννοια της συνέχειας στο παιχνίδι. Για παράδειγμα αν ένας παίκτης συγκρουστεί με κάποιο αντικείμενο καλείται αυτόματα η συνάρτηση `onCollision()` τόσο του παίκτη όσο και του αντικειμένου ώστε να καθοριστεί ποια είναι η συνέπεια αυτού του γεγονότος για τον κόσμο (αύξηση σκορ, τραυματισμός παίκτη, εξαφάνιση του αντικειμένου, γενικός συναγερμός...οτιδήποτε!)

3.2.5 Datablocks

Όπως αναφέρθηκε και προηγουμένως υπάρχει ανάγκη για γρήγορη επικοινωνία μεταξύ του πελάτη και του εξυπηρετητή και αυτό συχνά σημαίνει την ελάχιστη ανταλλαγή δεδομένων μεταξύ τους. Η TGE το καταφέρνει αυτό εισάγοντας την έννοια των datablocks.

Τα datablocks είναι μία ξεχωριστή κατηγορία αντικειμένων, το κύριο γνώρισμα των οποίων είναι ότι έχουν έναν μεγάλο αριθμό χαρακτηριστικών που δεν μεταβάλλονται.

Η χρησιμότητά τους μπορεί να φανεί μέσα από ένα σχετικά απλό παράδειγμα. Ας υποθέσουμε έναν κόσμο με πολλούς παίκτες, όπου ο καθένας από αυτούς έχει την ικανότητα να δημιουργεί οχήματα. Κάθε όχημα που σέβεται τον εαυτό του έχει μεταξύ 4 και 8 τροχών, skins, ειδικά εφέ (ήχοι, dust emitters κλπ) και προφανώς έναν μακρύ κατάλογο από φυσικά χαρακτηριστικά (βάρος, ύψος, τριβή, μέγιστη ταχύτητα και πολλά πολλά άλλα). Η συνολική δομή που περιγράφει ένα τέτοιο όχημα θα χρειαζόταν περίπου 2KB μνήμης για να αποθηκευτεί.

Σε οποιαδήποτε στιγμή του παιχνιδιού, στο οποίο μπορεί να συμμετέχουν και 32 παίκτες, ο καθένας μπορεί να δημιουργήσει ένα νέο όχημα. Με την δεδομένη αρχιτεκτονική πελάτη- εξυπηρετητή αυτό σημαίνει ότι ο εξυπηρετητής πρέπει να δημιουργήσει το αντικείμενο και μετά να το μεταδώσει στον αντίστοιχο πελάτη. Στην χειρότερη περίπτωση όπου όλοι οι παίκτες είναι σε οπτική επαφή μεταξύ τους και θελήσουν ταυτόχρονα να δημιουργήσουν από ένα νέο όχημα, ο εξυπηρετητής θα κληθεί να στείλει ένα σύνολο από 32 x 2KB δεδομένων σε καθέναν από τους 32 παίκτες ώστε να ενημερωθούν για τις αλλαγές. Αυτό μεταφράζεται σε μια κίνηση περίπου 2MB δεδομένων που πρέπει να διακινηθούν άμεσα! Σε συνδυασμό με όλες τις άλλες πληροφορίες που αφορούν τις θέσεις των παικτών κτλ είναι προφανές ότι σύντομα το σύστημα θα υποφέρει από έλλειψη διαθέσιμων πόρων.

Λαμβάνοντας τώρα υπ' όψιν ότι τα datablocks είναι σχεδιασμένα ώστε να έχουν τις παρακάτω ιδιότητες:

- Όλα τα datablocks αντιγράφονται από τον εξυπηρετητή σε κάθε πελάτη.
- Ένα datablock με αριθμό 123 στον εξυπηρετητή είναι σίγουρο ότι θα έχει τον ίδιο αριθμό- ταυτότητα σε όλους τους πελάτες
- Τα datablocks μεταδίδονται στους πελάτες στην αρχή και δεν ανανεώνονται ποτέ ξανά, με αποτέλεσμα να φαίνονται στατικά.

Με την εισαγωγή των datablocks το παραπάνω σενάριο αλλάζει δραστικά. Καταρχάς το datablock θα προδιαγράφει όλα τα χαρακτηριστικά των οχημάτων και θα μεταδίδεται μία και μόνη φορά, στην αρχή. Με αυτά τα δεδομένα, η δημιουργία ενός νέου οχήματος μπορεί να επιτευχθεί απλά με την μετάδοση ενός μικρού όγκου δεδομένων που περιέχει, την ταυτότητα του datablock, την θέση του στο χάρτη και ενδεχομένως κάποιες ακόμη πληροφορίες. Το εκτιμώμενο μέγεθος αυτού του πακέτου πληροφοριών δεν υπερβαίνει τα 64bytes! Είναι μάλλον προφανές ότι αυτή η μέθοδος διαχειρίζεται πολύ πιο λογικά την συνεχή δημιουργία και καταστροφή αντικειμένων που χαρακτηρίζει τα σύγχρονα παιχνίδια, και την ανάγκη να έχουν όλοι οι παίκτες μία ορθή εικόνα για τον κόσμο άμεσα.

3.3 Συμπληρωματικά εργαλεία

3.3.1 Show Tool Pro

Το Show Tool Pro (STP) είναι ένα πρόγραμμα οπτικοποίησης τρισδιάστατων γραφικών για την TGE. Δίνει τη δυνατότητα να δούμε τα μοντέλα των αντικειμένων και των παικτών όπως αυτά θα φαίνονται μέσα στο παιχνίδι προσομοιώνοντας τις ίδιες ακριβώς συνθήκες (εξ' άλλου το ίδιο το Show Tool Pro είναι γραμμένο σαν εφαρμογή της TGE γεγονός που αναδεικνύει πόσο μεγάλο εύρος έχουν οι εφαρμογές που μπορούμε να δημιουργήσουμε).



Το Show Tool Pro επίσης μας επιτρέπει να δούμε δομικά στοιχεία των μοντέλων όπως οι σύνδεσμοί τους, να ελέγξουμε τη σκίαση και το φωτισμό τους, να εξετάσουμε τα collision – boxes των αντικειμένων σε λεπτομέρεια, να δούμε τα textures τους αλλά κυρίως να δούμε και να συνδυάσουμε τις διάφορες εμψυχώσεις (animations) των αντικειμένων με μία πολύ απλή διεπαφή.

Όλα αυτά τα χαρακτηριστικά είναι απαραίτητα για την δημιουργία των παιχνιδιών καθώς θα ήταν πολύ δύσκολο έως αδύνατο (από την άποψη του απαιτούμενου χρόνου) να ελεγχτούν αποτελεσματικά όλα τα παραπάνω μόνο με την χρήση scripts στην TGE.

Αυτό που πρέπει ωστόσο να επισημάνουμε είναι ότι το Show Tool Pro είναι ένα εποπτικό εργαλείο που μας βοηθάει να δούμε τα αντικείμενα μας πριν τα εισάγουμε στον κόσμο. Αυτό δεν σημαίνει όμως ότι μπορούμε να επεξεργαστούμε τα μοντέλα που φορτώνουμε. Για να γίνει κάτι τέτοιο σε περίπτωση που εντοπιστεί κάποιο λάθος, ή απλά το μοντέλο δεν είναι σύμφωνο με αυτό που θέλαμε να σχεδιάσουμε τότε πρέπει αναγκαστικά να γυρίσουμε πίσω στο εργαλείο που χρησιμοποιήθηκε για να φτιαχτεί εξαρχής το εν λόγω μοντέλο.

Συγκεκριμένα οι δυνατότητες που μας προσφέρει το Show Tool Pro είναι οι εξής:

Τρισδιάστατη απεικόνιση

- Μπορούμε να φορτώσουμε πολλαπλά .dts αντικείμενα, το καθένα στην δική του όψη. Τα αρχεία .dsq μπορούν να φορτωθούν ξεχωριστά ή μέσα από ένα απλό script για να φορτωθούν πολλαπλά animations ταυτόχρονα.
- Υποστηρίζονται πολλαπλά rendering formats όπως: textured shaded, untextured shades, textured wireframed, vertices κτλ. Έτσι μπορούμε να απομονώσουμε κάποιο στοιχείο του αντικειμένου και να το ελέγξουμε χωριστά από τα άλλα.
- Εκτός από την προοπτική όψη, μπορούμε να δούμε όλες τις πλευρές του αντικειμένου (κάτοψη, πλάγια όψη κτλ)
- Μπορούμε να μοντάρουμε (mount!?) ένα οποιοδήποτε αντικείμενο πάνω σε κάποιο άλλο, με τη χρήση οποιοδήποτε κόμβου (πχ ένα όπλο στα χέρια ενός παίκτη)
- Μπορούν να καθοριστούν αυτόματα (όπως θα γινόταν στο παιχνίδι) ή να επιλεγθούν με το χέρι τα διάφορα levels of detail του κάθε αντικειμένου.
- Υπάρχει η δυνατότητα να δούμε τα bounding boxes, collision meshes αλλά και τα collision meshes της ορατής περιοχής από τον παίκτη (line of sight).

- Μπορούμε να διαχειριστούμε την φωτεινή πηγή μετακινώντας την ή σβήνοντας την για να είναι δυνατόν να ελέγξουμε ανεμπόδιστα τα textures που έχουμε εφαρμόσει.

Υλικά και Υφή

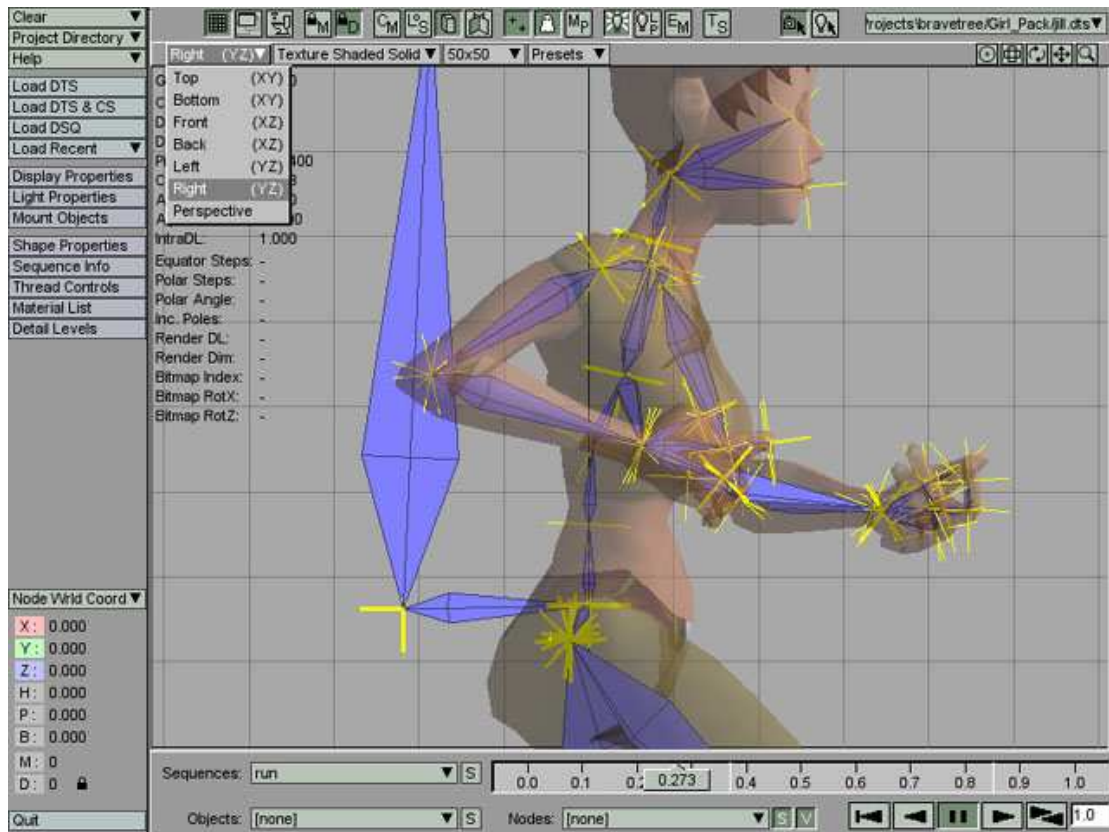
- Τα υλικά που χρησιμοποιούμε για τον χρωματισμό των αντικειμένων μπορούν να φορτώνονται και να εναλλάσσονται με μεγάλη ευκολία, καθώς επίσης είναι ορατές όλες οι ιδιότητές τους όπως αυτές είναι καθορισμένες στο αντίστοιχο dts αρχείο.

Animations

- Τα animations μπορούν να αναπαραχθούν προς τα μπρος ή προς τα πίσω, με καθορισμένο από το χρήστη τον χρονισμό της αναπαραγωγής.
- Πολλαπλά animation threads μπορούν αν συνδυαστούν
- Μπορούμε να δούμε όλες τις παραμέτρους ελέγχου των animations συμπεριλαμβανομένου της λίστας των ελεγχόμενων κόμβων.

Δομική Όψη

- Όλοι οι κόμβοι και η μεταξύ τους ιεραρχία είναι εμφανής. Όταν επιλεγθεί ένας κόμβος τότε μπορούμε να δούμε την θέση του (τόσο σε παγκόσμιες όσο και σε τοπικές συντεταγμένες) καθώς και τον άξονά του.
- Όταν ένας κόμβος ελέγχει κάποιες ακμές, τότε αυτές χρωματίζονται αντίστοιχα.
- Μπορούμε να δούμε ολόκληρη την ιεραρχική δομή ενός .dts αντικειμένου γεγονός το οποίο είναι χρήσιμο για την ανίχνευση λαθών, ενώ επίσης μπορούμε να δούμε όλα τα διαφορετικά levels of detail του αντικειμένου και πιο συγκεκριμένα τη σειρά με την οποία θα σχεδιαστούν τα πολύγωνα.



Σχήμα 9 - Όψη του Show Tool Pro όπου βλέπουμε έναν εικονικό χαρακτήρα και τη δομή του σκελετού του.

3.3.2 Online υποστήριξη – forum.

Μπορεί να φαίνεται στην αρχή ασήμαντο αλλά η TGE είναι μία ιδιαίτερα δημοφιλής μηχανή με εξαιρετικές δυνατότητες και η οποία προσφέρεται (μαζί με τον πηγαίο κώδικά της) σε μία προσιτή, για ένα αρκετά ευρύ κοινό, τιμή. Αυτό είχε σαν αποτέλεσμα να δημιουργηθεί μία ιδιαίτερα ενεργή κοινότητα τόσο επαγγελματιών όσο ερασιτεχνών προγραμματιστών που πειραματίζονται με νέες ιδέες και δεν διστάζουν να τις επικοινωνήσουν και στους άλλους.

Έτσι υπάρχει διαρκώς η ευκαιρία να συζητηθεί κάποιο πρόβλημα με άλλους που ενδεχομένως να το αντιμετωπίσαν στο παρελθόν και κατά συνέπεια να βρεθεί μία καλύτερη λύση, αλλά επίσης υπάρχει η δυνατότητα να ενσωματώσει κανείς έξυπνες ιδέες και τροποποιήσεις. Όπως αναφέραμε και προηγουμένως αυτό είναι ένα από τα πλεονεκτήματα της ύπαρξης της Torquescript, το οποίο φαίνεται η κοινότητα που ασχολείται με την ανάπτυξη TGE εφαρμογών να το έχει εκμεταλλευτεί στο έπακρο.

Τέλος η TGE υποστηρίζεται από την garagegames με ένα μεγάλο αριθμό tutorials, on-line οδηγιών αλλά και πλήρους documentation που καλύπτει κάθε πλευρά της διαδικασίας ανάπτυξης μίας εφαρμογής.

4 Το περιβάλλον Moods

Στα πλαίσια λοιπόν αυτής της θεωρητικής προσέγγισης που αφορά την ρεαλιστική συμπεριφορά των χαρακτήρων μέσα σε ένα εικονικό κόσμο, αναπτύχθηκε με την Torque Game Engine ένα σχετικά από περιβάλλον, όπου όμως μας δίνεται η δυνατότητα να παρατηρήσουμε μερικές από τις αρχές που περιγράψαμε παραπάνω. Θα αναφερόμαστε στο εξής στο περιβάλλον αυτό ως το περιβάλλον του Moods.

Το Moods αναπτύχθηκε στην λογική μίας εφαρμογής πελάτη-εξυπηρετητή, όπου ο εξυπηρετητής μπορεί να είναι αυτόνομος και να τρέχει σε έναν ξεχωριστό υπολογιστή πάνω στο δίκτυο. Ο χρήστης του Moods μπορεί στην αρχή να επιλέξει αν θέλει να συνδεθεί σε κάποιον εξυπηρετητή ή αν προτιμάει να παίζει αυτόνομα (με την αρχιτεκτονική single player που έχουμε ήδη περιγράψει).

4.1 Το σενάριο της εφαρμογής

Το σενάριο που διέπει την εφαρμογή Moods είναι σχετικά απλό. Ο παίκτης, βρίσκεται σε έναν εικονικό κόσμο ο οποίος είναι γεμάτος από όμοια αντικείμενα τριών χρωμάτων. Ο παίκτης μπορεί να μαζέψει τα αντικείμενα αυτά και να παρακολουθήσει την πορεία του σκορ αλλά και της διάθεσής του μιας και τα αντικείμενα αυτά αντιπροσωπεύουν γεγονότα τα οποία επηρεάζουν την διάθεσή του με έναν συγκεκριμένο τρόπο.

Παράλληλα με την συλλογή των αντικειμένων, στον κόσμο κινούνται και δύο NPCs οι οποίοι έχουν διαφορετικούς στόχους και συμπεριφέρονται με διαφορετικό τρόπο. Ο ένας από αυτούς έχει σκοπό να βοηθήσει τον παίκτη να μαζέψει τα αντικείμενα του χρώματος της επιλογής του (και άρα καλείται να πάρει αποφάσεις σχετικά με την συμπεριφορά του παίκτη) ενώ ο δεύτερος έχει τον ακριβώς αντίθετο σκοπό και προσπαθεί να αποτρέψει τον παίκτη απ' το να μαζέψει όλα τα αντικείμενα που έχει σκοπό.



Σχήμα 10 - Η είσοδος του παίκτη στον περιβάλλον Moods

4.2 Θεωρητικά στοιχεία που ενσωματώθηκαν στην εφαρμογή

Στην εφαρμογή Moods προσπαθήσαμε να ενσωματώσουμε διάφορα θεωρητικά στοιχεία από αυτά που έχουμε περιγράψει μέχρι τώρα καθώς και γενικότερα προσπαθήσαμε να δημιουργήσουμε ένα άρτιο περιβάλλον, εύκολα επεκτάσιμο, στο οποίο να μπορούν να μελετηθούν οι συμπεριφορές τόσο ενός παίκτη όσο και ενός NPC.

Επιρροή της διάθεσης

Στην εφαρμογή μας η διάθεση του παίκτη μοντελοποιείται κατάλληλα σαν μία συνεχής μεταβλητή στο διάστημα $[0,1]$ η οποία και μεταβάλλεται με το μάζεμα των αντικειμένων. Όπως έχουμε αναφέρει η διάθεση μπορεί να μεταβάλλει τον τρόπο με τον οποίο ένας παίκτης κινείται στον χώρο. Έτσι στο Moods η διάθεση είναι αυτή που καθορίζει την μέγιστη ταχύτητα με την οποία μπορεί να κινηθεί ο παίκτης. Δηλαδή ένας παίκτης όταν έχει καλή διάθεση (mood κοντά στη μονάδα) μπορεί να κινηθεί με πολύ μεγαλύτερη ταχύτητα απ' όταν η διάθεσή του είναι κακή (mood κοντά στο μηδέν).

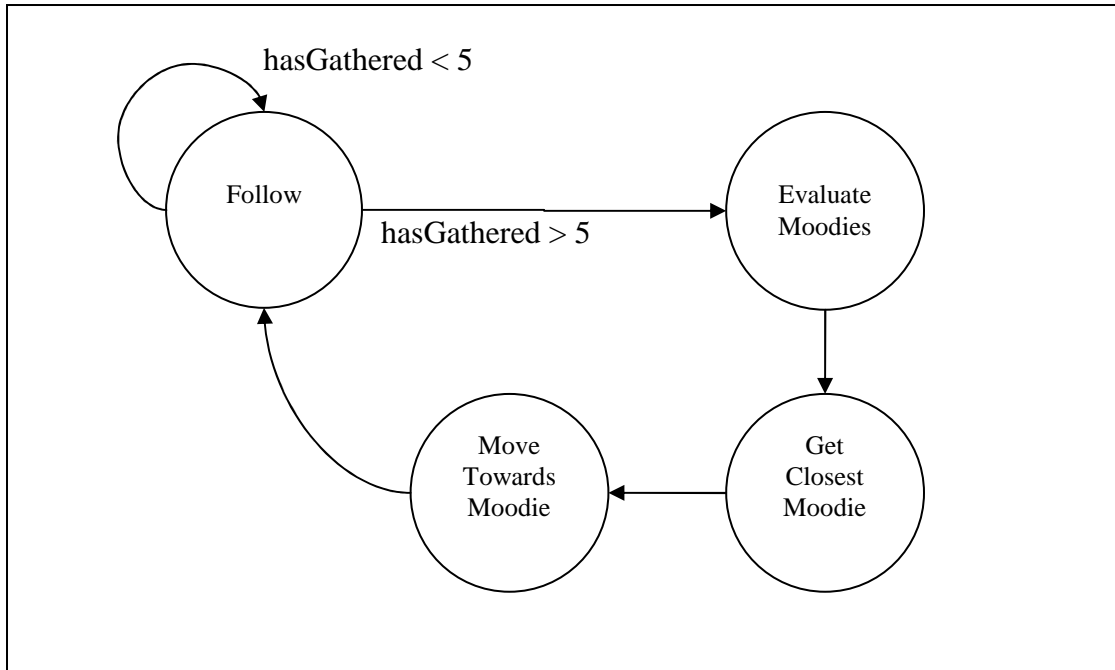
Προσαρμογή Συμπεριφοράς

Οι NPCs του Moods έχουν την ικανότητα να προσαρμόζονται στις κινήσεις του παίκτη κατά τη διάρκεια του παιχνιδιού. Αυτό σημαίνει ότι ο NPC που έχει στόχο να βοηθήσει τον παίκτη παρουσιάζει μία μιμητική συμπεριφορά και σε γενικές γραμμές ακολουθεί το παράδειγμά του. Τον ακολουθεί καθ' όλη την διάρκεια του παιχνιδιού και προσπαθεί να καταλάβει (με τον δικό του ενδεχομένως απλοϊκό τρόπο) τι είναι αυτό που προσπαθεί να πετύχει ο παίκτης, και όταν το καταλάβει τότε προσπαθεί να τον βοηθήσει, μαζεύοντας και αυτός αντικείμενα για λογαριασμό του.

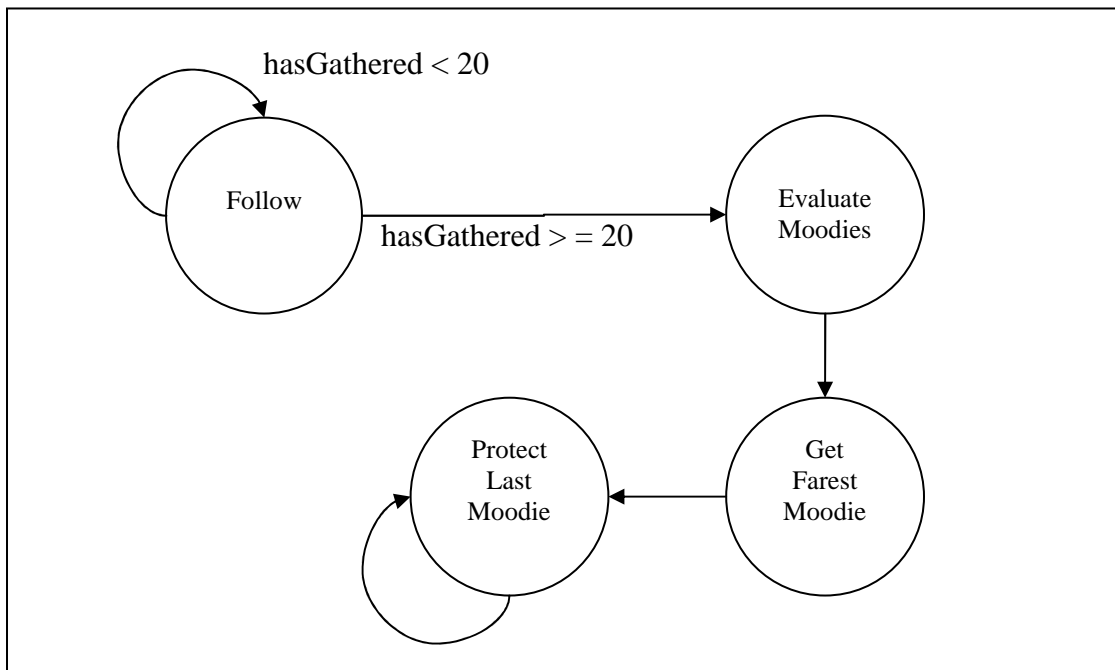
Ο παίκτης δηλαδή κινείται στο χώρο και μαζεύει αντικείμενα (αυτό μπορεί να το κάνει με όποιο τρόπο θέλει αλλά γενικά υποθέτουμε ότι έχει σκοπό να μαζέψει όλα τα αντικείμενα ενός χρώματος, έχοντας όμως το δικαίωμα να αλλάξει ανά πάσα στιγμή την χρωματική του επιλογή), οι NPCs τον ακολουθούν και παρατηρώντας από κοντά τις κινήσεις του αποφασίζουν και προσαρμόζουν την συμπεριφορά τους αντίστοιχα.

Η τρόπος με τον οποίο οι NPCs παίρνουν την απόφασή τους σχετικά με το τι χρώμα μαζεύει ο παίκτης διαφέρει ανάλογα με το αν πρόκειται για τον καλό ή τον κακό χαρακτήρα, αλλά και στις δύο περιπτώσεις πρόκειται για ένα είδος επεκτεταμένης voting machine όπου το πλήθος των αντικειμένων από κάθε κατηγορία που έχει μαζέψει ο παίκτης αποτελεί τους ψήφους. Οι NPCs λοιπόν, έχοντας σαν δεδομένο την αλληλουχία των αντικειμένων που έχει επιλέξει ο παίκτης αποφασίζουν ποιο είναι το νικητήριο χρώμα, το χρώμα δηλαδή που θεωρούν ότι μαζεύει ο παίκτης.

Η αλληλουχία των καταστάσεων από τις οποίες περνάει ο κάθε παίκτης προκειμένου να έχουν ακριβώς αυτή τη συμπεριφορά που περιγράφουμε μπορεί να φανεί στα παρακάτω διαγράμματα κατάστασης (Σχήματα 11 και 12).



Σχήμα 11 – Διάγραμμα καταστάσεων Φίλου-NPC



Σχήμα 12 – Διάγραμμα καταστάσεων Εχθρού-NPC

Οι NPCs δηλαδή αρχικά ακολουθούν τον παίκτη, παρατηρούν τις κινήσεις του μέχρι να θεωρήσουν ότι υπάρχει επαρκές δείγμα αντικειμένων για να μπορέσουν να πάρουν την απόφασή τους και στην συνέχεια υλοποιούν την συμπεριφορά τους. Στον μεν Φίλο-NPC η όλη διαδικασία όπως βλέπουμε είναι αναδρομική, ενώ στον δε Εχθρό-NPC εκτελείται μία φορά.

Εισαγωγή συναισθημάτων

Δεδομένου ότι κινούμαστε σε ένα επιστημονικό πεδίο το οποίο έχει να κάνει με τα συναισθήματα των ανθρώπων και τον τρόπο με τον οποίο τα εκφράζουν και τα αντιλαμβάνονται είναι λογικό να θέλουμε να ξέρουμε όσο περισσότερα πράγματα γίνεται σχετικά με τον τρόπο αυτά δημιουργούνται και αλληλεπιδρούν.

Οι κλασικές μέθοδοι πρόκλησης συναισθημάτων σε ανθρώπους για πειραματικούς σκοπούς, συνήθως υστερούν καθώς τα δυναμικά χαρακτηριστικά των συναισθημάτων δεν μπορούν να εξεταστούν. Γι' αυτό το λόγο έχει προταθεί το περιβάλλον ενός παιχνιδιού για την διεξαγωγή πειραμάτων μιας και έτσι οι συμμετέχοντες είναι κάτι παραπάνω από απλοί παρατηρητές. Βρίσκονται μέσα στο περιβάλλον του πειράματος και συμμετέχουν με διαδραστικό τρόπο.

Η γενικότερη ιδέα είναι η παρουσίαση ενός παιχνιδιού και η παρακολούθηση των αντιδράσεων των παικτών μπροστά σε αναπάντεχα γεγονότα, με σκοπό να τους προκληθούν συναισθήματα θυμού και αναστάτωσης. [9]

Σε αυτά τα γενικά πλαίσια αλλά και δεδομένου ότι ο σχεδιασμός της εφαρμογής έγινε με γνώμονα την προοπτική της επέκτασης, θεωρήθηκε χρήσιμο να εξετάσουμε τις δυνατότητες του εξυπηρετητή όταν αυτός λειτουργεί ανεξάρτητα από τους πελάτες. Γενικά είναι σαφές ότι όλες οι βασικές λειτουργίες περνάνε μέσα από τον εξυπηρετητή μιας και όπως έχουμε πει αυτός αναλαμβάνει οτιδήποτε μπορεί να είναι βαρύ για να το σηκώσει ένας πελάτης. Αυτό όμως μας δίνει την ευκαιρία να δούμε πόσο μπορούμε να επηρεάσουμε τον κόσμο του παίκτη καθώς και τον ίδιο.

Είναι αρκετά ενδιαφέρον λοιπόν όταν αναλύουμε την διάσταση που αφορά τα συναισθήματα να αναφερόμαστε και σε αυτά των παικτών όταν παίζουν ένα παιχνίδι καθώς θέλουμε να αναλύσουμε πώς ο παίκτης βιώνει εντονότερα την εμπειρία αυτή. Μέσα σε αυτή την ανάλυση συχνά θέλουμε να μπορούμε έστω και με δική μας επέμβαση να προκαλούμε διάφορες αντιδράσεις στον παίκτη και να παρατηρούμε πώς αντιμετωπίζει τα διάφορα συμβάντα.

Εκτός λοιπόν από τους υπολογιστές που καταφέρνουν να εκδηλώσουν κάποια μορφή συναισθηματικής κατάστασης, στόχος είναι να σχεδιαστούν κάποτε και οι λεγόμενοι affective υπολογιστές, αυτοί δηλαδή που καταφέρνουν να προκαλέσουν συγκεκριμένα συναισθήματα στον χρήστη. (Το Affective computing περιγράφεται από την Picard ως “*computing that relates to, arises from or deliberately influences emotions*” [12])

Με αυτά τα δεδομένα κατά νου, τροποποιήσαμε τον εξυπηρετητή ώστε να έχει γραφικό περιβάλλον και να επιτρέπει στον διαχειριστή του να επέμβει στο παιχνίδι και να προκαλέσει απροσδόκητα γεγονότα στον εικονικό κόσμο του χρήστη. Οι λειτουργίες που μας παρέχονται είναι οι εξής:

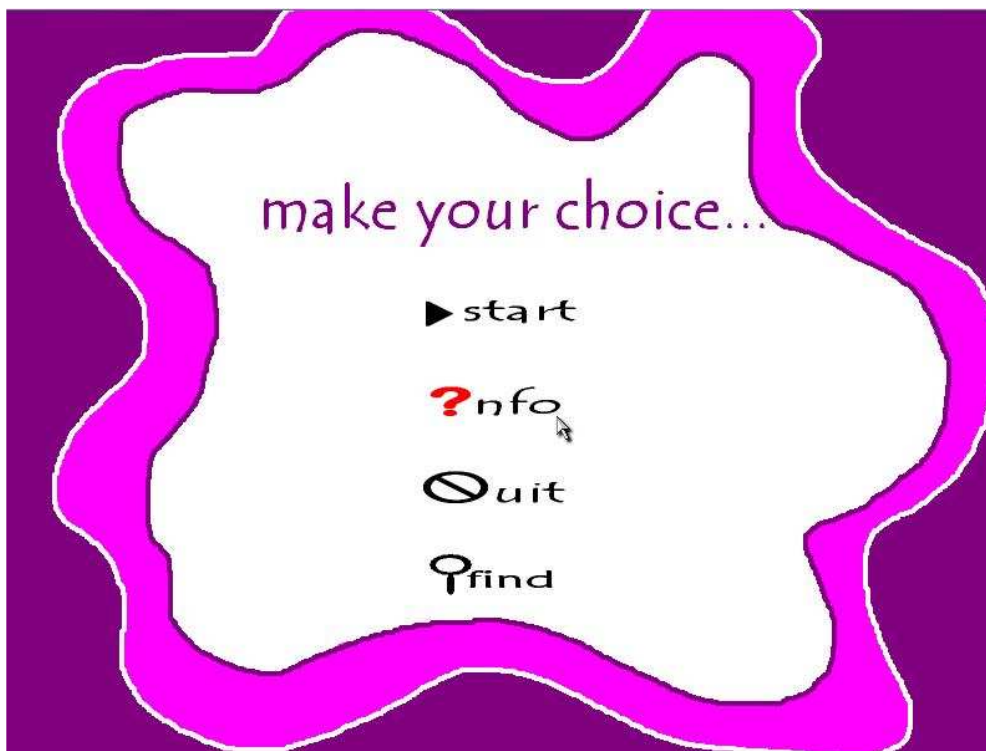
- Μπορούμε να κάνουμε τον παίκτη να κινηθεί με όποιον τρόπο θέλουμε, παρά το τι πατάει ο ίδιος. Έχουμε δηλαδή έναν εξυπηρετητή ο οποίος μπορεί να λειτουργήσει σαν remote control και να μπορούμε να χειριστούμε έναν παίκτη παρά το γεγονός ότι δεν είμαστε στον υπολογιστή στον οποίο τρέχει ο client.
- Μπορούμε να κάνουμε το avatar του παίκτη να παίζει όποιο από τα διαθέσιμα animations θέλουμε.
- Μπορούμε να μεταβάλλουμε την διάθεση του παίκτη και κατ' επέκταση να επηρεάσουμε την ταχύτητα με την οποία κινείται.

Εάν θέλουμε λοιπόν μπορούμε να κάνουμε το παιχνίδι να συμπεριφέρεται με απροσδόκητο τρόπο αλλοιώνοντας π.χ. τις κινήσεις του παίκτη και σε μία μελλοντική επέκταση να δούμε πως αυτό το γεγονός μπορεί να επηρεάσει και τον τρόπο με τον οποίο παίζει ο παίκτης, βοηθώντας μας να καταλάβουμε τι συμπεριφορά θα πρέπει να παρουσιάζει ο ιδανικός NPC ώστε να προκαλέσει ακόμα μεγαλύτερη συναισθηματική εμπλοκή του παίκτη.

4.3 Περιγραφή του περιβάλλοντος

4.3.1 Περιβάλλον client

Το πρώτο πράγμα που βλέπει ένας παίκτης κατά τη εκκίνηση του Moods είναι το menu επιλογών.



Σχήμα 13 – Το βασικό μενού της εφαρμογής

Οι δυνατότητες που παρέχονται στον χρήστη είναι:

1. Εκκίνηση εφαρμογής

Το παιχνίδι ξεκινάει αμέσως με το πάτημα αυτού του πλήκτρου. Ανεβαίνει ο τοπικός εξυπηρετητής, δημιουργείται ο εικονικός κόσμος και εισάγεται το avatar του παίκτη ο οποίος είναι έτοιμος να αρχίσει να κινείται.

2. Πληροφορίες

Ο χρήστης έχει την δυνατότητα να επισκεφθεί μία ιστοσελίδα (υποθέτουμε ότι υπάρχει διαθέσιμη σύνδεση με το διαδίκτυο) ώστε να βρει περισσότερες πληροφορίες σχετικά με την εφαρμογή Moods και τους στόχους της.

3. Εύρεση Εξυπηρετητή

Εάν ο παίκτης θελήσει να συνδεθεί με έναν απομακρυσμένο εξυπηρετητή τότε μπορεί να το κάνει αυτό αφού όμως πρώτα κάνει την κατάλληλη αναζήτηση για να εντοπίσει τους υπάρχοντες εξυπηρετητές πάνω στο δίκτυο στο οποίο είναι συνδεδεμένος. Εάν βρεθεί κάποιος ενεργός εξυπηρετητής τότε ο χρήστης μπορεί να τον επιλέξει να συνδεθεί σε αυτόν, καθορίζοντας παράλληλα και το όνομα με το οποίο θα είναι πλέον γνωστός, και να συνεχίσει μετά στον εικονικό κόσμο. Η διαφορά λοιπόν με την απλή εκκίνηση του παιχνιδιού είναι ότι τώρα δεν ανεβαίνει τοπικός εξυπηρετητής αλλά ο παίκτης συνδέεται σε κάποιον ήδη υπάρχοντα.



Σχήμα 14 - Menu Αναζήτησης Εξυπηρετητών

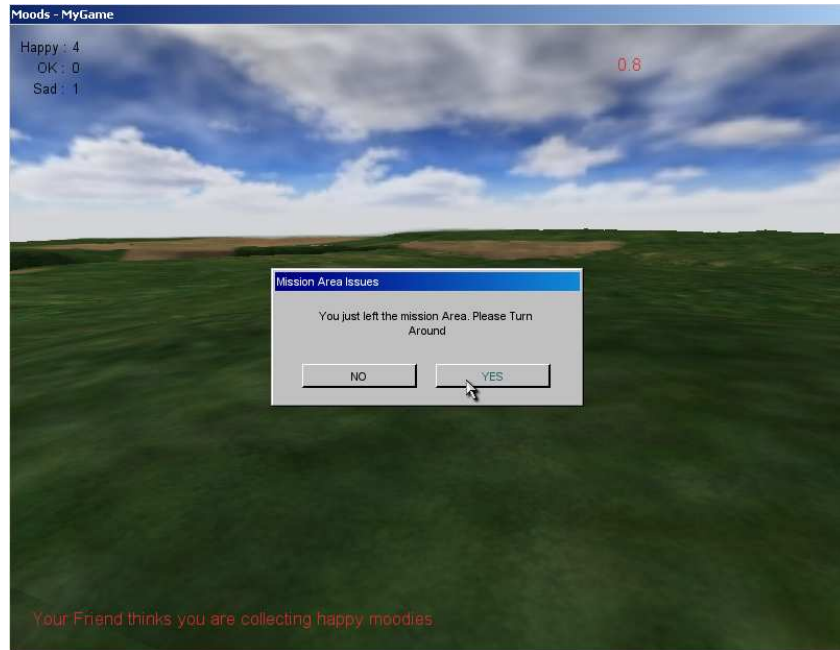
4. Εξόδος

Φυσικά μαζί με όλες τις άλλες επιλογές που παρέχονται στον χρήστη είναι και αυτή της εξόδου από το περιβάλλον του Moods.

Αφού βρεθεί ο παίκτης στον εικονικό κόσμο μπορεί να μετακινηθεί σε μία μεγάλη περιοχή που ονομάζεται mission area και η οποία αντιπροσωπεύει την ενεργό περιοχή του παιχνιδιού, την περιοχή δηλαδή όπου υπάρχουν αντικείμενα και στην οποία καλούνται να κινηθούν ο παίκτης και οι NPCs. Για να μην βρεθεί ο παίκτης εκτός από αυτήν την περιοχή και καταλήξει να περιπλανιέται άσκοπα σε αδρανείς για το παιχνίδι χώρους, όταν πάει να εξέλθει από την mission area τότε εμφανίζεται ένα ενημερωτικό μήνυμα που του ζητάει να γυρίσει πίσω. Αυτό γίνεται μέσω της συνάρτησης `onLeaveMissionArea()` που έχει ο παίκτης και η οποία καλείται αυτόματα από την μηχανή. Στην συνάρτηση αυτή που φαίνεται παρακάτω βλέπουμε την ευκολία που μας παρέχει η TGE στο να εμφανίζουμε μηνύματα σε μορφή `MessageBoxes`.

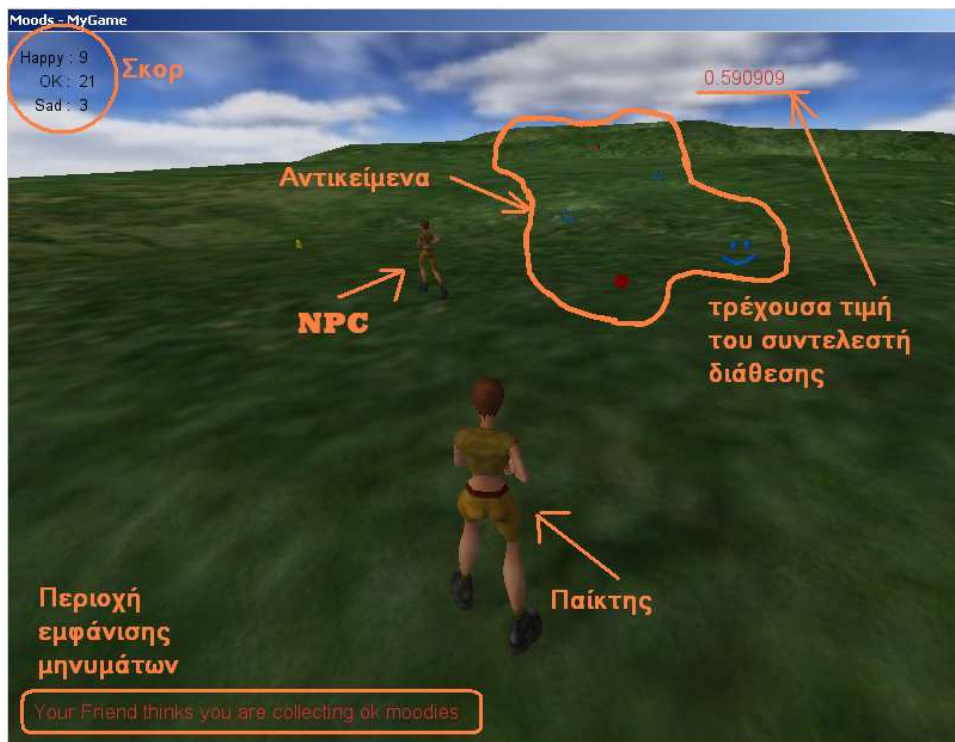
Συνάρτηση `onLeaveMissionArea()`

```
function PlayerAvatar::onLeaveMissionArea(%this)
{
    MessageBoxYesNo( "Mission Area Issues",
                    "You just left the mission Area. Please Turn Around", "", "" );
}
```

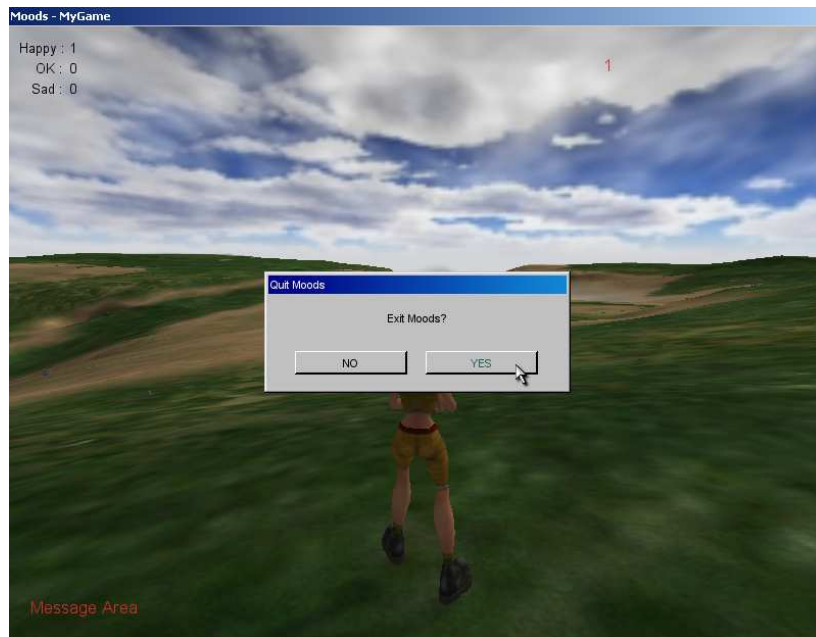


Σχήμα 15 – Προειδοποίηση εξόδου από την mission area

Ο χρήστης λοιπόν κινείται στον εικονικό κόσμο ο οποίος είναι γεμάτος με τριών ειδών αντικείμενα (μπλε, κόκκινα και κίτρινα) τα οποία μπορεί και να μαζέψει. Στην οθόνη κατά τη διάρκεια του παιχνιδιού ο παίκτης μπορεί να δει ανά πάσα στιγμή τον αριθμό των αντικειμένων που έχει συλλέξει καθώς και τον αριθμό που αντιπροσωπεύει την διάθεσή του. Τέλος στο κάτω μέρος της οθόνης υπάρχει χώρος για την εμφάνιση ενημερωτικών μηνυμάτων.



Σχήμα 16 - Τα Huds και τα βασικά στοιχεία του κόσμου.

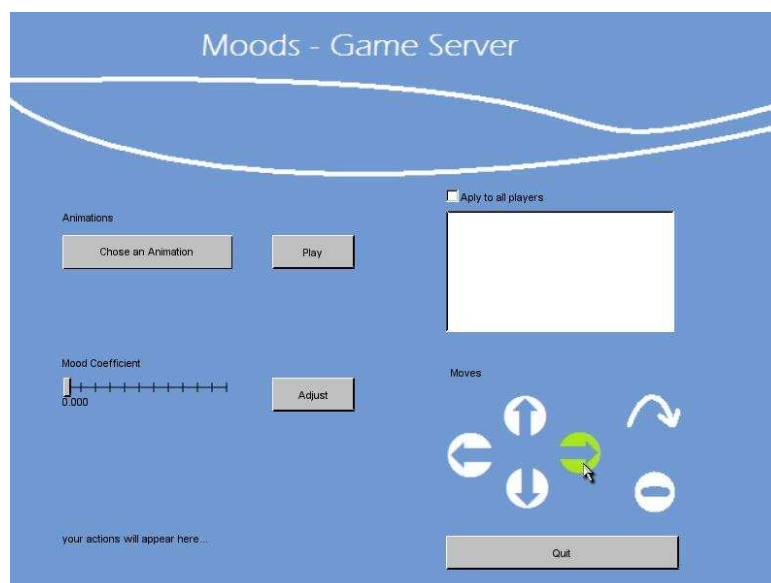


Σχήμα 17 – Παράθυρο εξόδου από την εφαρμογή

Η έξοδος από το παιχνίδι γίνεται με το πλήκτρο Esc, το πάτημα του οποίου εμφανίζει ένα MessageBox επιβεβαίωσης για την έξοδο.

4.3.2 Περιβάλλον server

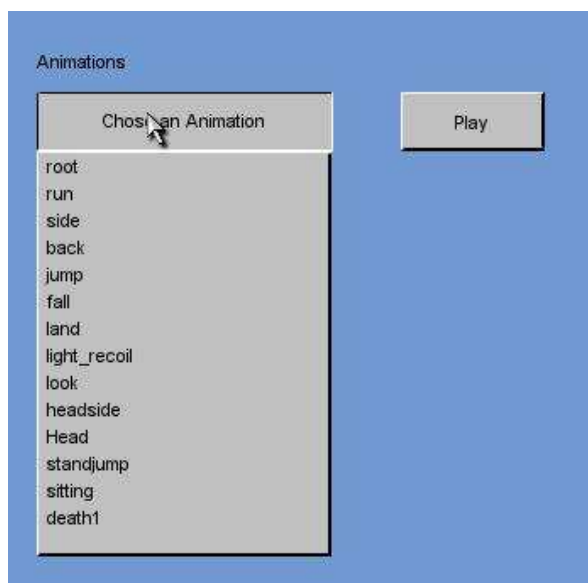
Ο εξυπηρετητής του Moods έχει γραφικό περιβάλλον μόνο στην περίπτωση που είναι αυτόνομος και τρέχει σε έναν ξεχωριστό υπολογιστή. Στην περίπτωση Single Player ο εξυπηρετητής τρέχει στο παρασκήνιο και ο χρήστης δεν έχει πρόσβαση σε αυτόν. Στο παρακάτω σχήμα 18 φαίνεται το Graphical User Interface μέσω του οποίου μπορεί ο χειριστής του εξυπηρετητή να έχει πρόσβαση στις διάφορες λειτουργίες που του παρέχει.



Σχήμα 18 – Γραφικό περιβάλλον εξυπηρετητή

Όπως βλέπουμε οι επιλογές είναι πολυάριθμες. Καταρχάς μπορούμε να δούμε πόσοι χρήστες είναι συνδεδεμένοι την τρέχουσα στιγμή στον εξυπηρετητή. Στην δική μας εφαρμογή θα είναι πάντα ένας, αλλά το χαρακτηριστικό αυτό είναι πολύ χρήσιμο άμα εξετάσουμε τις μελλοντικές δυνατότητες επέκτασης του Moods.

Μια δεύτερη λειτουργία είναι η δυνατότητα να επιλέξουμε έναν παίκτη και να τον κάνουμε να παίξει ένα οποιοδήποτε animation. Οι χαρακτήρες όπως έχουμε αναφέρει παραπάνω περιγράφονται πλήρως με ένα .dts αρχείο στο οποίο συμπεριλαμβάνονται και όλα τα animations που μπορεί να δείξει κάθε χαρακτήρας. Έχοντας πρόσβαση στα ονόματα αυτών των animations μπορούμε να καλέσουμε όποια επιθυμούμε.



Σχήμα 19 - Διαθέσιμα animations για τον χαρακτήρα του Moods

Στην συνέχεια βλέπουμε ότι μπορούμε να μεταβάλουμε μέσω της αντίστοιχης μπάρας την τιμή της διάθεσης του κάθε παίκτη. Αυτό, όπως θα δούμε παρακάτω, έχει άμεση συνέπεια στον τρόπο με τον οποίο κινείται ο παίκτης.

Τέλος με την ομάδα κουμπιών της κάτω αριστερής γωνίας μπορούμε να κινήσουμε έναν παίκτη με όποιον τρόπο θέλουμε (μπρος, πίσω, αριστερά, δεξιά, άλμα και πλήρης ακινητοποίηση).

4.4 Περιγραφή *gameplay*

Εάν ο παίκτης παρατηρήσει τον χώρο στον οποίο βρίσκεται θα δει τρία βασικά χαρακτηριστικά. Καταρχάς παντού τριγύρω του βρίσκονται αντικείμενα τριών χρωμάτων (Moodies) και δύο ακόμη NPCs που τον ακολουθούν όπου πάει (Friend, Enemy). Σε γενικές γραμμές μπορούμε να πούμε ότι ο στόχος του παίκτη είναι να μαζέψει όποια αντικείμενα του αρέσουν και εμείς παρατηρούμε τους δύο NPC οι οποίοι έχουν διαφορετική αντιμετώπιση απέναντι στον παίκτη, να προσπαθούν ο μιν Friend να τον βοηθήσει, ο δε Enemy να τον εμποδίσει.

Παίκτης

Ο παίκτης δημιουργείται κατά την σύνδεση του client με τον server. Η συνάρτηση που καλείται αρχικά είναι η `onClientEnterGame` του αντικειμένου `GameConnection` που αντιστοιχεί στον παίκτη. Το αντικείμενο αυτό παρέχει τις βασικές λειτουργίες επικοινωνίας μεταξύ ενός παίκτη και του εξυπηρετητή με τον οποίο έχει συνδεθεί (αποτελεί έναν άμεσο τρόπο επικοινωνίας σε σχέση με τον έμμεσο τρόπο ανταλλαγής μηνυμάτων). Παρατηρούμε αρχικά ότι στον κάθε παίκτη αντιστοιχίζεται μία κάμερα, μέσα από την οποία βλέπει τον κόσμο στην οθόνη του, και επίσης ότι κατά την δημιουργία του παίκτη προγραμματίζεται η κλήση της συνάρτησης που δημιουργεί τα αντικείμενα (Moodies) στο περιβάλλον. Για την συνάρτηση αυτή θα μιλήσουμε αργότερα όταν περιγράψουμε την γενικότερη λειτουργικότητα των αντικειμένων αυτών.

Συνάρτηση `onClientEnterGame()`

```
function GameConnection::OnClientEnterGame(%this)
{
    %this.camera = new Camera()
    {
        dataBlock = Observer;
    };
    MissionCleanup.Add( %this.camera );

    %spawnPoint = getSpawnPoint();
    %this.CreatePlayer(%spawnPoint);
    (...)
    schedule( 2000, 0, "ServerCmdSpawnItems", %this.player);
}
```

Συνάρτηση δημιουργίας παίκτη

```
function GameConnection::CreatePlayer(%this, %spawnPoint)
{
    if (%this.player > 0)
    {
        Error( "Attempting to create a second Avatar!");
    }

    %player = new Player()
    {
        dataBlock = PlayerAvatar;
        client = %this;
    };

    MissionCleanup.Add( %player );

    %player.SetTransform(%spawnPoint); // where to put it

    %this.camera.SetTransform(%player.GetEyeTransform());
    %player.SetEnergyLevel(100);

    %this.player = %player;
    %this.setControlObject(%player);
}
```

```
%this.happy = 0;  
%this.sad = 0;  
%this.ok = 0;  
}
```

Βλέπουμε λοιπόν ότι κατά την δημιουργία του παίκτη, τον εφοδιάζουμε με έναν δείκτη στον εαυτό του και επίσης αργότερα αρχικοποιούνται και οι δείκτες friend και enemy ώστε να δείχνουν στους δύο NPCs. Επίσης βλέπουμε ότι αρχικοποιούνται οι τιμές των μεταβλητών happy, sad και ok που αντιπροσωπεύουν την συνολική συγκομιδή του παίκτη από τα διάφορα αντικείμενα.

Moodies

Τα αντικείμενα που βρίσκονται στον χώρο (και τα οποία ονομάσαμε Moodies) αντικατοπτρίζουν διάφορα γεγονότα που αλλάζουν την διάθεση του παίκτη. Έτσι τα μπλε αντικείμενα τον κάνουν να είναι σε καλή διάθεση (happyMoodies), τα κόκκινα τον κάνουν να είναι άσχημη διάθεση (sadMoodies) ενώ τα κίτρινα τον κάνουν να είναι δε μια ουδέτερη κατάσταση (okMoodies).



HappyMoodie



OkMoodie



SadMoodie

Σχήμα 20 – Τα τρία αντικείμενα της εφαρμογής

Φυσικά η αλλαγή από τη μία διάθεση στην άλλη δεν γίνεται με τρόπο on-off. Δηλαδή τα αντικείμενα επιδρούν στην διάθεση, δεν την καθορίζουν. Όπως έχουμε ήδη αναφέρει η διάθεση μοντελοποιείται ως μία συνεχής μεταβλητή στο διάστημα [0,1] και κάθε αντικείμενο που μαζεύει ο παίκτης την σπρώχνει προς το ένα ή το άλλο άκρο. Όσον αφορά τώρα τον κώδικα της εφαρμογής η μεταβλητή αυτή είναι global ώστε να μπορούν τα διάφορα στοιχεία να την προσπελαίνουν με το όνομα moodCoefficient και κάθε φορά που ο παίκτης συγκρούεται με ένα αντικείμενο η τιμή της μεταβλητής αυτής ανανεώνεται αντίστοιχα.

Όπως αναφέραμε και προηγουμένως τα αντικείμενα δημιουργούνται μετά την είσοδο του παίκτη στον κόσμο μέσω της συνάρτησης ServerCmdSpawnItems. Η δημιουργία των παικτών γίνεται σε τυχαίες θέσεις, φυσικά εντός της mission area, και μάλιστα φροντίζουμε ώστε τα αντικείμενα να ομαδοποιούνται σε τρεις διαφορετικές κατηγορίες για να μπορούμε ανά πάσα στιγμή να ξέρουμε πόσα αντικείμενα υπάρχουν ακόμα στον χώρο. Και εδώ η δημιουργία γίνεται μέσω datablocks, γεγονός αναμενόμενο αφού έχουμε για κάθε παίκτη 50 αντικείμενα από την κάθε κατηγορία.

Συνάρτηση ServerCmdSpawnItems (π.χ. για τα happyMoodies)

```
function ServerCmdSpawnItems(%client){

    %width = GetWord(MissionArea.area, 2);
    %height = GetWord(MissionArea.area, 3);
    %west = GetWord(MissionArea.area, 0);
    %south = GetWord(MissionArea.area, 1);

    for(%i = 0; %i < 50; %i++)
    {
        %x = GetRandom(%width) + %west;
        %y = GetRandom(%height) + %south;
        %searchMasks = $TypeMasks::PlayerObjectType |
                      $TypeMasks::InteriorObjectType |
                      $TypeMasks::TerrainObjectType |
                      $TypeMasks::ShapeBaseObjectType;
        %scanTarg = ContainerRayCast(%x SPC %y SPC "500", %x SPC %y SPC
                                   "-100", %searchMasks);
        if(%scanTarg &&
            !(%scanTarg.getType() & $TypeMasks::InteriorObjectType))
        {
            %newpos = GetWord(%scanTarg,1) SPC GetWord(%scanTarg,2)
                    SPC GetWord(%scanTarg,3)+1;
        }
        Item::spawnMoodie("happyMoodie", 0,%newpos);
    }
}
```

Συνάρτηση spawnItem()

```
function Item::spawnMoodie(%data, %index, %location)
{
    %obj = new Item() {
        dataBlock = %data;
        count = %index;
    };
    MissionCleanup.add(%obj);

    if ( %data $= "happyMoodie")
        %groupName = "MissionGroup/HappyGroup";
    else if ( %data $= "okMoodie")
        %groupName = "MissionGroup/OkGroup";
    else
        %groupName = "MissionGroup/SadGroup";

    %group = nameToID(%groupName);
    if (%group != -1)
    {
        %group.add(%obj);
    }
    else
    {

```

```

        echo("No Group found");
    {
        %spawn=%location;
        %obj.setTransform(%spawn);
        %obj.playThread(0,"SimpleAnimation");

        return %obj;
    }

```

Κατά τη σύγκρουση του παίκτη με ένα αντικείμενο καλείται αυτόματα η callback συνάρτηση του αντικειμένου με το όνομα onCollision() με παραμέτρους τα αντικείμενα με τα οποία έγινε η σύγκρουση. Στην συνάρτηση αυτή αφού καθοριστεί ότι είναι ο παίκτης που συγκρούστηκε με το αντικείμενο, ανανεώνεται η διάθεσή του και στην συνέχεια ενημερώνεται αντίστοιχα και το σκορ του που φαίνεται στην οθόνη. Επίσης διαγράφεται το αντικείμενο από τον κόσμο και από το αντίστοιχο group (όλα τα moodies είναι οργανωμένα σε ομάδες ανάλογα με τον τύπο τους). Επειδή αυτή η συνάρτηση καλείται ακόμη και όταν συγκρούεται με το αντικείμενο και ένας NPC πρέπει να μεριμνήσουμε για την λειτουργικότητά της και σε αυτή την περίπτωση.

Παράδειγμα συνάρτησης onCollision() για το αντικείμενο happyMoodie

```

function happyMoodie::onCollision(%this,%obj,%col)
{
    if ( %col.getState() != "Dead" )
    {
        %groupName = "MissionGroup/HappyGroup";
        %group = nameToID(%groupName);
        if (%group != -1)
        {
            %group.remove(%obj);
        }
        else
            echo("No Group found");
        %obj.delete();
        if (%col.client)
        {
            %col.incHappy(%col);
            %col.adjustMood(%col);
        }
        else{
            %col.friend.incHappy(%col.friend);
            %col.friend.adjustMood(%col.friend);
        }
    }
}

```

Η συνάρτηση incHappy που καλείται στον παίκτη αυξάνει το σκορ, ενώ η adjustMood είναι αυτή που υπολογίζει την επιρροή του νέου αντικειμένου στον συνολικό συντελεστή διάθεσης.

Συνάρτηση IncHappy()

```
function ShapeBase::incHappy(%this,%user)
{
    %user.happy = %user.happy + 1;
    commandToClient(%user.client, 'IncHappy');
}
```

Συνάρτηση adjustMood()

```
function ShapeBase::adjustMood(%this, %user)
{
    %totalMoodies = %this.happy + %this.sad + %this.ok;
    %avg = (%this.happy + %this.ok*0.5)/%totalMoodies;
    if(%user.client){
        commandToClient(%user.client, 'AdjustMood', %avg);
    }
    %user.client.friend.setSpeed(%avg);
}
```

Συνάρτηση clientCmdIncHappy

```
function clientCmdIncHappy()
{
    happyNum.setValue(happyNum.getValue() + 1);
}
```

Συνάρτηση clientCmdAdjustMood

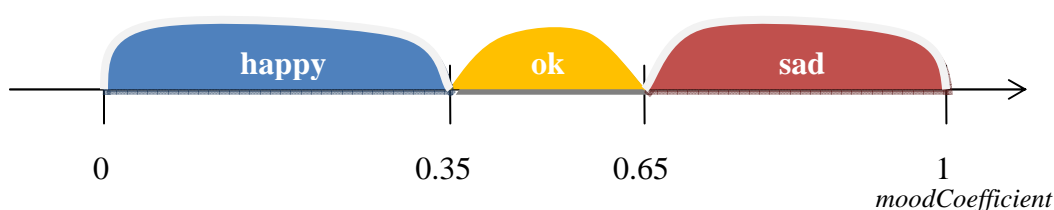
```
function clientCmdAdjustMood(%avg)
{
    $moodCoefficient = %avg;
}
```

Στην συνάρτηση `incHappy` παρατηρούμε και την επιρροή της αρχιτεκτονικής πελάτη-εξυπηρετητή στον κώδικα. Η ανίχνευση της σύγκρουσης του παίκτη με το αντικείμενο γίνεται στον εξυπηρετητή, παρόλα αυτά τα μηνύματα που φαίνονται στην οθόνη είναι στην εμβέλεια του χρήστη και άρα ο εξυπηρετητής δεν έχει πρόσβαση στις μεταβλητές που τις ελέγχουν. Έτσι βλέπουμε την επικοινωνία με μηνύματα μεταξύ πελάτη και εξυπηρετητή: ο εξυπηρετητής στέλνει μήνυμα στον κατάλληλο πελάτη (σε αυτόν δηλαδή που συγκρούστηκε με το αντικείμενο, μιας και θα μπορούσαμε να είμαστε σε multiplayer περιβάλλον) με την παράμετρο `'IncHappy'` και έτσι ο `client` καταλαβαίνει ότι πρέπει να καλέσει την συνάρτηση `clientCmdIncHappy`.

Αυτό είναι γενικά το πρότυπο επικοινωνίας μεταξύ του εξυπηρετητή και του χρήστη. Όταν ο `server` στείλει ένα μήνυμα `commandToClient(%user.client, 'functionName')` τότε ο πελάτης όταν το λάβει αναζητεί μία συνάρτηση με το όνομα `clientCmdFunctionName()` την οποία και στην συνέχεια καλεί. Το αντίστοιχο συμβαίνει και από την μεριά του πελάτη όταν αυτός στέλνει στον εξυπηρετητή ένα μήνυμα τύπου `commandToServer ('functionName')`. Παρατηρούμε ότι στο μήνυμα του πελάτη απουσιάζει η πρώτη παράμετρος η οποία και αντιπροσωπεύει τον

παραλήπτη του μηνύματος. Αυτό συμβαίνει γιατί ενώ ο εξυπηρετητής πρέπει να μπορεί να στείλει σε έναν οποιονδήποτε πελάτη είναι συνδεδεμένος μαζί του, την περίπτωση ενός πελάτη υπάρχει μόνο ένας εξυπηρετητής στον οποίο είναι συνδεδεμένος και έτσι δεν χρειάζεται να ορίσει παραλήπτη.

Στην συνάρτηση `adjustMood()` βλέπουμε τον τρόπο με τον οποίο υπολογίζεται η διάθεση του παίκτη. Αυτό που συμβαίνει είναι ότι κάθε φορά υπολογίζεται ο σταθμισμένος μέσος όρος των αντικειμένων που έχει ήδη μαζέψει ο παίκτης (`moodCoefficient`) και ο αριθμός που προκύπτει αντιστοιχίζεται μία συγκεκριμένη διάθεση. Θεωρούμε ότι τα `happyMoodies` έχουν βάρος +1, τα `sadMoodies` 0, και τα `okMoodies` $\frac{1}{2}$, ενώ αντίστοιχα ο `moodCoefficient` που προκύπτει από την εξής αντιστοίχιση:



Σχήμα 21 – Η αντιστοίχιση του συντελεστή διάθεσης στις τρεις κατηγορίες διάθεσης

Η αντιστοίχιση αυτή δεν έχει κάποια λειτουργικότητα αλλά θα την βρούμε αργότερα μπροστά μας όταν θελήσουν οι NPCs να πάρουν μία απόφαση σχετικά με την διάθεση στην οποία πιστεύουν ότι βρίσκεται ο παίκτης.

Η χρησιμότητα του `moodCoefficient` έγκειται στο ότι επηρεάζει τον τρόπο με τον οποίο κινείται ο παίκτης. Στο Moods αυτό σημαίνει ότι κάθε χρονική στιγμή η μέγιστη ταχύτητα με την οποία μπορεί να κινηθεί ένας παίκτης είναι συνάρτηση του συντελεστή αυτού.

Όταν ο χρήστης πατάει τα πλήκτρα με τα οποία κινείται τότε καλούνται οι κατάλληλες συναρτήσεις ώστε να υπολογίσει η μηχανή πόσο πρέπει να κινήσει το avatar του παίκτη (κάθε πλήκτρο που έχει μία λειτουργικότητα πρέπει να αντιστοιχηθεί με μία συνάρτηση). Κανονικά το μέγεθος της κίνησης καθορίζεται μόνο από τη διάρκεια του πατήματος του πλήκτρου, ενώ τώρα επηρεάζεται και από την τιμή του `moodCoefficient`. Παρακάτω φαίνεται η σύνδεση των πλήκτρων με τις συναρτήσεις και επίσης η προσθήκη του `moodCoefficient` ώστε να αλλάζει η ταχύτητα κίνησης του παίκτη.

Σύνδεση των πλήκτρων κίνησης με τις αντίστοιχες συναρτήσεις

```
PlayerKeymap.Bind(keyboard, up, GoAhead);  
PlayerKeymap.Bind(keyboard, down, BackUp);  
PlayerKeymap.Bind(keyboard, left, GoLeft);  
PlayerKeymap.Bind(keyboard, right, GoRight);
```

Συναρτήσεις κίνησης του παίκτη – επιρροή του moodCoefficient

```
function GoLeft(%val)  
{  
    $mvLeftAction = (%val * $moodCoefficient);  
}
```

```

function GoRight(%val)
{
    $mvRightAction = (%val * $moodCoefficient);
}

function GoAhead(%val)
{
    $mvForwardAction = (%val * $moodCoefficient);
}

function BackUp(%val)
{
    $mvBackwardAction = (%val * $moodCoefficient);
}

```

Με τον τρόπο αυτό καταφέρνουμε να έχουμε, έστω και με σχετικά απλό τρόπο, μία εξάρτηση της κίνησης των χαρακτήρων μέσα στο παιχνίδι από την διάθεσή τους. Αυτό είναι χρήσιμο για τον ρεαλισμό των χαρακτήρων αυτών γιατί όπως είπαμε είναι σημαντικό ακόμα και η κίνησή τους να είναι συμβατή με αυτά που συμβαίνουν. Για παράδειγμα σε ένα πολεμικό παιχνίδι όπου είμαστε συνηθισμένοι να βλέπουμε ένα μετρητή με την «ζωή» που έχει ο παίκτης μας, είναι λογικό να περιμένουμε να κινείται π.χ. πιο αργά όταν είναι σοβαρά τραυματισμένος, απ' ότι όταν έχει πλήρη ενέργεια.

Friend

Από τους δύο NPCs που αρχικά βλέπουμε να ακολουθούν τον παίκτη μας ο ένας από αυτούς είναι θετικά προδιατεθειμένος και θέλει να μας βοηθήσει. Για να γίνει αυτό όμως ο παίκτης έχει αρχικά τον στόχο να καταλάβει τι είναι αυτό που θέλουμε να καταφέρουμε.

Ένα θέμα που πρέπει να συζητηθεί είναι το γεγονός ότι οι NPCs δεν έχουν την δυνατότητα να αντιληφθούν τον κόσμο με τον ίδιο τρόπο όπως ένας άνθρωπος που κινείται στον ίδιο χώρο με αυτούς. Δεν έχουν αισθήσεις, δεν μπορούν να δουν τι συμβαίνει τριγύρω τους και κατ' επέκταση δεν μπορούν να έχουν τον ίδια ευκολία στο να βγάζουν συμπεράσματα σχετικά με το ακριβώς γίνεται τριγύρω τους. Όταν όμως ο στόχος μας είναι να έχουμε ρεαλιστικούς NPCs αυτό είναι ένα εμπόδιο που πρέπει να υπερσκελίσουμε. Θα μπορούσαμε ενδεχομένως να μοντελοποιήσουμε τις αισθήσεις με τέτοιο τρόπο ώστε να μπορούν να τις υλοποιήσουν και οι εικονικοί χαρακτήρες. Αυτό όμως είναι ένα ιδιαίτερα δύσκολο εγχείρημα καθώς πρόκειται για ένα ερευνητικό πεδίο ακόμη ανοικτό, αλλά και πέρα από αυτό θα ήταν σίγουρα ασύμφορο όσον αφορά την υπολογιστική ισχύ που θα απαιτούνταν. Δεν πρέπει εξάλλου να ξεχνάμε ότι όλα αυτά στα οποία αναφερόμαστε θα εφαρμοστούν σε εμπορικές εφαρμογές οι οποίες δεν είναι δυνατόν να έχουν μεγάλες απαιτήσεις καθώς αναφέρονται στον μέσο χρήστη.

Σε αυτά τα πλαίσια είναι μία συνήθης τακτική να επιτρέπεται στους NPCs να «κλέβουν» προκειμένου να έχουν πρόσβαση σε απαραίτητα δεδομένα για την αντίληψη του γύρω κόσμου. Ας μην ξεχνάμε ότι ο σχεδιαστής του παιχνιδιού έχει πρόσβαση σε κάθε δεδομένο που αφορά τον εικονικό περιβάλλον και είναι στο χέρι του να γράψει έτσι τον κώδικα του NPC ώστε και αυτός να έχει αντίστοιχα προνόμια.

Αυτό βέβαια πρέπει να γίνει με τέτοιο τρόπο ώστε να μην είναι εμφανές στους παίκτες γιατί θα έχει το αντίθετο από το επιθυμητό αποτέλεσμα. Έχουμε ήδη αναφέρει ότι κανένας παίκτης δεν θέλει να παίξει ένα παιχνίδι που είναι υπερβολικά δύσκολο, πόσο μάλλον αν έχει την υπόνοια ότι δεν παίζουν όλοι με ίσους όρους.

Στο Moods λοιπόν οι NPCs έχουν μία μικρή βοήθεια ώστε να μπορούν να αντιλαμβάνονται τις κινήσεις του παίκτη. Η βοήθεια αυτή έγκειται στο γεγονός ότι οι χαρακτήρες κατά τη δημιουργία τους έχουν ένα πεδίο στο οποίο αποθηκεύεται η ταυτότητα του παίκτη τον οποίο θέλει να βοηθήσει (πεδίο friend). Αντίστοιχα και ο παίκτης έχει ένα πεδίο στο οποίο αποθηκεύεται η ταυτότητα των δύο NPCs (πεδία friend και enemy αντίστοιχα). Έτσι μπορεί να έχει ο ένας πρόσβαση στη θέση του άλλου και επίσης ο παίκτης μπορεί να ενημερώνει τους δύο NPCs κάθε φορά που μαζεύει ένα αντικείμενο.

Όλα αυτά γίνονται κατά στην πλευρά του server ο οποίος είναι υπεύθυνος για την δημιουργία όλων των χαρακτήρων στο παιχνίδι, κατά την σύνδεση ενός νέου παίκτη όπως είδαμε πριν με την συνάρτηση `onClientEnterGame`.

Δημιουργία του παίκτη και των NPCs και πέρασμα παραμέτρων

Συνάρτηση `onClientEnterGame()`

```
function GameConnection::OnClientEnterGame(%this)
{
    (...)
    %spawnPoint = getSpawnPoint();
    %this.CreatePlayer(%spawnPoint);

    %spawnPoint = VectorSub(%spawnPoint, "2 2 0 0 0 0");
    %this.CreateFriend(%spawnPoint);

    %spawnPoint = VectorSub(%spawnPoint, "2 2 0 0 0 0");
    %this.CreateEnemy(%spawnPoint);
    (...)
}
```

Παρατηρούμε ότι οι εικονικοί χαρακτήρες δημιουργούνται μέσω μιας συνάρτησης του ίδιου του παίκτη, είναι δηλαδή άρρηκτα συνδεδεμένοι μαζί του.

Η συνάρτηση `getSpawnPoint()` επιστρέφει ένα σημείο στο οποίο μπορεί να δημιουργηθεί το avatar του παίκτη. Το σημείο αυτό δεν είναι τυχαίο αλλά υπάρχει μία ομάδα από προκαθορισμένα σημεία στα οποία μπορεί να γίνει η εμφάνισή του και κάθε φορά επιλέγεται με τυχαίο τρόπο ένα από αυτά.

Συνάρτηση `getSpawnPoint()`

```
function getSpawnPoint()
{
    %groupName = "MissionGroup/PlayerSpawns";
    %group = nameToID(%groupName);
    if (%group != -1) {
        %count = %group.getCount();
        if (%count != 0) {
            %index = getRandom(%count-1);
            %spawn = %group.getObject(%index);
        }
    }
}
```

```

        %spawnPoint = %spawn.getTransform();
    }
    else{
        error("No spawn points found in " @ %groupName);
        %spawnPoint = "2 2 220 1 0 0 0";
    }
}
else{
    error("Missing spawn points group " @ %groupName);
    %spawnPoint = "2 2 220 1 0 0 0";
}
return %spawnPoint;
}

```

Κατά την δημιουργία των δύο NPCs περνιέται σαν παράμετρος η θέση στην οποία πρέπει να δημιουργηθεί το avatar τους και το οποίο είναι λίγο πίσω από τον παίκτη. Επίσης βλέπουμε ότι η δημιουργία του παίκτη και των NPCs γίνεται με χρήση datablocks όπως έχουμε περιγράψει.

Συνάρτηση δημιουργίας του Friend NPC : createFriend()

```

function GameConnection::CreateFriend(%this, %spawnPoint)
{
    if (%this.friend > 0)
    {
        echo("Player already has a friend" );
    }
    else
    {
        %friend = new AIPlayer()
        {
            dataBlock = FriendAvatar;
        };
        MissionCleanup.Add( %friend );

        %friend.SetTransform(%spawnPoint);
        %friend.setMoveSpeed(1);
        %friend.SetEnergyLevel(100);

        %this.player.friend = %friend;
        %friend.friend = %this.player;
    }
}

```

Μετά την δημιουργία του παίκτη καλείται η συνάρτηση thinkGood η οποία αντιπροσωπεύει την κύρια λειτουργία σκέψης του NPC. Η λογική που ακολουθεί ο friend NPC έχει ήδη φανεί στο διάγραμμα κατάστασης που παραθέσαμε προηγουμένως και στο οποίο φαίνεται η κατάσταση στην οποία βρίσκεται ο χαρακτήρας σε κάθε στιγμή. Η συνάρτηση thinkGood είναι αναδρομική (προγραμματίζει δηλαδή μία κλήση στον εαυτό της) μιας και θέλουμε να συνεχίζει να σκέφτεται και να εκτιμάει τα γεγονότα του κόσμου για όλη τη διάρκεια του παιχνιδιού.

Η λογική δηλαδή που ακολουθεί ο friend NPC είναι η εξής: αρχικά ακολουθεί τον παίκτη (συνάρτηση Follow) μέχρις ότου αυτός να μαζέψει τα 5 πρώτα αντικείμενα. Τότε κρίνει ότι μπορεί να πάρει μία πρώτη απόφαση σχετικά με το τι αντικείμενα νομίζει ότι προσπαθεί να μαζέψει ο παίκτης (έχουμε υποθέσει ότι ο στόχος του παίκτη είναι να συγκεντρώσει κάποια αντικείμενα, του ίδιου χρώματος). Για να πάρει αυτή την απόφαση (συνάρτηση evaluateMoodies) ο χαρακτήρας λαμβάνει υπ' όψιν όχι μόνο την συνολική συγκομιδή του παίκτη αλλά και τα τελευταία 5 αντικείμενα που έχει μαζέψει. Αυτό γίνεται για να μπορεί ο NPC να ακολουθεί τις διάφορες αποφάσεις του παίκτη, ο οποίος μπορεί κάποια στιγμή να αλλάξει γνώμη και να αρχίσει να μαζεύει άλλου χρώματος αντικείμενα. Η έξοδος της συνάρτησης evaluateMoodies είναι η κατηγορία των αντικειμένων που πιστεύει ο NPC ότι μαζεύει ο χρήστης. Εδώ λοιπόν χρησιμεύει και η αντιστοίχιση του moodCoefficient στις αντίστοιχες κατηγορίες διάθεσης.

Συνάρτηση Evaluate Moodies

```
function AIPlayer::evaluateMoodies()
{
    %max = $TotalHappy;
    %mood1 = 1;
    if ($totalOk >= %max)
    {
        %max = $totalOk;
        %mood1 = 0.5;
    }
    if ($totalSad > %max)
    {
        %max = $totalSad;
        %mood1 = 0;
    }
    $sum = 0;
    for(%i=0; %i<5; %i++)
        %sum = %sum + $last5Moodies[%i];
    %mood2 = %sum / 5;

    %mood = (%mood1 * 0.4) + (%mood2 * 0.6);
    if ( %mood<0.35 )
        return "sad";
    else if (%mood > 0.65)
        return "happy";
    else
        return "ok";
}
```

Συνάρτηση Follow()

```
function AIPlayer::Follow(%this)
{

    %this.stopThread(0);

    %player = %this.friend;
    %y = getWord( %player.getDataBlock().boundingBox, 1);
    %pos = getWords( %player.getTransform(), 0, 2);
    %oldPos = %pos;
```

```

%vec = "0 -1 0";
%vec = MatrixMulVector(%player.getTransform(), %vec);
%pos = "0 0 0";
%pos = VectorAdd(%oldPos, vectorScale(%vec, %y) );

%followPos = %pos;
%pos = getWords(%this.getTransform(), 0, 2);
%vec = VectorSub(%followPos, %pos);
%len = VectorLen(%vec);

%this.setAimObject(%player);
if(%len>getWord(%this.boundingBox, 1))
{
    %this.setMoveDestination(%followPos, true);
    %this.setMoveSpeed(0.5);
}
}

```

Αφού λοιπόν ο χαρακτήρας πάρει την απόφαση ότι ο παίκτης μαζεύει π.χ. happyMoodies (τα οποία τον κάνουν να πηγαίνει και πιο γρήγορα και περιμένουμε να τα προτιμήσει) τότε προχωράει στην επόμενη διαδικασία που είναι να εντοπίσει κάποιο Moodie κοντά στον παίκτη. Αυτό που κάνει είναι να ανατρέξει στην αντίστοιχη ομάδα αντικειμένων και να ελέγξει την απόστασή τους από τον παίκτη μιας και του επιτρέπουμε να μαζέψει ένα αντικείμενο μόνο αν βρίσκεται σε μια περιοχή συγκεκριμένης ακτίνας γύρω από τον παίκτη. Εδώ βλέπουμε και τα σημεία στα οποία μπορούμε να θεωρήσουμε ότι ο παίκτης «κλέβει». Αφού ο NPC ακολουθεί από την αρχή της δημιουργίας του τον παίκτη, τότε είναι προφανές ότι δεν μπορεί να ξέρει που είναι τα αντικείμενα γιατί δεν τα έχει δει όλα. Εμείς όμως του δίνουμε πρόσβαση στην λίστα με τα αντικείμενα για να μπορεί έτσι να εντοπίσει τα κοντινά στον παίκτη τα οποία μπορούμε να υποθέσουμε ότι είναι και εντός του οπτικού του πεδίου. Η αναζήτηση του κοντινότερου αντικειμένου γίνεται με την συνάρτηση `getClosestItem()` η οποία και επιστρέφει την θέση του αντίστοιχου αντικειμένου.

Συνάρτηση `getClosestItem()`

```

function AIPlayer::getClosestItem(%this, %mood)
{
    %player = %this.friend;
    %playerPos = %player.getPosition();

    if ( %mood $= "happy")
        %groupName = "MissionGroup/HappyGroup";
    else if ( %mood $= "ok")
        %groupName = "MissionGroup/OkGroup";
    else
        %groupName = "MissionGroup/SadGroup";

    %group = nameToID(%groupName);
    if (%group != -1)
    {
        %count = %group.getCount();
        %found = false;
        %index = 0;
    }
}

```

```

while ( (%found == false) && (%index < %count) )
{
    %item = %group.getObject(%index);
    %itemPos = %item.getPosition();
    %dist = VectorDist( %itemPos , %playerPos );

    if (%dist <= 100)
        %found = true;
    %index++;
}
if( %found == true){
    return %itemPos;
}
else
{
    return %playerPos;
}
}
}

```

Όταν ο χαρακτήρας εντοπίσει ένα αντικείμενο κοντά στον παίκτη πάει και το μαζεύει (γεγονός που κάνει το σκορ του παίκτη να αυξάνεται και την διάθεσή του να μεταβάλλεται) και στην συνέχεια ξαναξεκινάει την διαδικασία. Εάν η αναζήτηση αντικειμένου είναι άκαρπη (είτε ο παίκτης πρόλαβε και τα μάζεψε όλα, είτε απλά δεν υπάρχουν αντικείμενα τριγύρω) τότε η συνάρτηση επιστρέφει την θέση του παίκτη ώστε παρά το γεγονός ότι δεν υπάρχει αντικείμενο, τουλάχιστον να συνεχίσει να ακολουθεί τον παίκτη.



Σχήμα 22 – Η λειτουργία του Φίλου- NPC : έχοντας εκτιμήσει ότι ο παίκτης μαζεύει happyMoodies, απομακρύνεται από αυτόν και πηγαίνει να τον βοηθήσει μαζεύοντας και αυτός happyMoodies.



Σχήμα 23 – Η λειτουργία του Φίλου- NPC : ενώ υπάρχουν happyMoodies κάπου στο χώρο, ο Φίλος- NPC δεν πηγαίνει να τα μαζέψει καθώς είναι έξω από την ακτίνα δράσης του.

Ένα ακόμη θέμα που έχουμε αφήσει χωρίς εξήγηση είναι πώς μαθαίνει ο NPC για τα αντικείμενα που έχει μαζέψει ο παίκτης. Σίγουρα η απάντηση «το είδε» δεν μας ικανοποιεί αφού εξηγήσαμε ότι οι εικονικοί χαρακτήρες δεν αντιλαμβάνονται τον χώρο γύρω τους. Εδώ λοιπόν βρίσκεται ακόμη ένα σημείο στο οποίο ο σχεδιασμός του χαρακτήρα τον κάνει να έχει πλεονέκτημα.

Ο παίκτης όταν μαζεύει ένα αντικείμενο τότε καλεί την συνάρτηση Inform και των δύο NPC με παράμετρο την κατηγορία του αντικειμένου που μάζεψε, μιας και έχει έναν δείκτη προς αυτούς και έτσι μπορεί να προσπελάσει τις συναρτήσεις τους. Με την συνάρτηση αυτή οι NPCs μπορούν να ενημερώσουν τις μεταβλητές που διαθέτουν και με τις οποίες παρακολουθούν τις επιλογές του παίκτη και να είναι πάντα ενημερωμένοι όταν έρθει η ώρα να πάρουν κάποια απόφαση.

Με την μέθοδο InsertMoodie ενημερώνεται ο πίνακας lifo που κρατάνε οι NPCs για τα 5 αντικείμενα που έχει μαζέψει ο παίκτης.

Συνάρτηση Inform()

```
function AIPlayer::Inform(%this, %name)
{
    $hasGathered ++;
    if (%name $= "happyMoodie" )
```

```

    {
        $totalHappy++;
        insertMoodie(1);
    }
    else if( %name $= "okMoodie" )
    {
        $totalOk++;
        insertMoodie(0.5);
    }
    else if( %name $= "sadMoodie"){
        $totalSad++;
        insertMoodie(0);
    }
}

```

Enemy

Ο δεύτερος NPC είναι αρνητικά προδιατεθειμένος απέναντι στον παίκτη και ο στόχος του είναι να τον εμποδίσει απ' το να μαζέψει όλα τα αντικείμενα. Αν και σε γενικές γραμμές οι συναρτήσεις που χρησιμοποιεί είναι οι ίδιες (Follow, Inform, evaluateMoodies) ο τρόπος με τον οποίο σκέφτεται και ενεργεί είναι διαφορετικός όπως φάνηκε και από το παραπάνω διάγραμμα καταστάσεων.

Σε αντίθεση λοιπόν με τον friend NPC ο enemy, παρακολουθεί για αρκετή ώρα τον παίκτη πριν πάρει την απόφαση σχετικά με την κατηγορία που θεωρεί ότι μαζεύει. Αυτό συμβαίνει γιατί όπως φαίνεται και από το διάγραμμα ο enemy παίρνει μία φορά αυτή την απόφαση και μετά απομακρύνεται και αναζητάει το Moodie την αντίστοιχης κατηγορίας που βρίσκεται πιο μακριά από τον παίκτη.

Αφού το εντοπίσει (με αντίστοιχο τρόπο με την getClosestItem) τότε κινείται προς αυτό και το προστατεύει. Φυσικά έπρεπε να λάβουμε ειδική μέριμνα ώστε εάν συγκρουστεί ο enemy με κάποιο αντικείμενο αυτό ούτε να εξαφανιστεί από τον κόσμο (όπως γίνεται με τον παίκτη και τον friend) αλλά ούτε να μεταβάλλεται τίποτα στο σκορ.



Σχήμα 24 – Λειτουργία Εχθρού NPC: Ο εχθρός NPC έχει εκτιμήσει ότι τελικά ο παίκτης μαζεύει HappyMoodies και εκείνη τη στιγμή ξεκινάει να κατευθύνεται προς το πιο απομακρυνόμενο. Εάν προλάβει να φτάσει πρώτος (όπως έγινε εδώ) τότε εμποδίζει τον Φίλο-NPC αλλά και τον ίδιο τον παίκτη απ' το να μαζέψουν το αντικείμενο αυτό.

5 Ελεκτάσεις και νέοι στόχοι

Όπως και κάθε πειραματική εφαρμογή έτσι και το περιβάλλον Moods δεν είναι τέλειο. Υπάρχουν πολλά ακόμη χαρακτηριστικά που μπορούμε να ενσωματώσουμε και πολλές ακόμη λειτουργίες που μπορούμε να υλοποιήσουμε και να εξετάσουμε μέσα από αυτό. Παρακάτω αναφέρουμε μερικά βασικά στοιχεία που θα μπορούσαν να επεκτείνουν τις δυνατότητες του Moods.

5.1 Βελτίωση αντίληψης των *Non Player Characters*

Ένα βασικό στοιχείο της εφαρμογής είναι ο τρόπος με τον οποίο οι εικονικοί χαρακτήρες αντιλαμβάνονται τον κόσμο και προσαρμόζουν την συμπεριφορά τους. Αυτή τη στιγμή ο κάθε εικονικός χαρακτήρας ενημερώνεται για τις κινήσεις του παίκτη όπου και αν βρίσκεται. Αυτό όμως δεν είναι ιδιαίτερα ρεαλιστικό. Αν για παράδειγμα ο παίκτης καταφέρει να ξεφύγει από τον Εχθρό-NPC τότε θα έπρεπε ο δεύτερος να μην έχει πρόσβαση στις ενέργειες του παίκτη (για όσο καιρό αυτός παραμένει εκτός του οπτικού του πεδίου). Αντίστοιχα αν ο Φίλος-NPC κατά την αναζήτηση αντικειμένων απομακρυνθεί πολύ, πάλι ενδεχομένως να χάσει την πληροφορία που αφορά κάποιες από τις ενέργειες του παίκτη.

Η υλοποίηση ενός τέτοιου χαρακτηριστικού θα απαιτούσε την εισαγωγή της έννοιας του οπτικού πεδίου όχι μόνο σε θεωρητικό επίπεδο αλλά και στον κώδικα της εφαρμογής. Θα μπορούσαμε δηλαδή να επιτρέπουμε στον κάθε χαρακτήρα να βλέπει μόνο ότι είναι μπροστά του και μέχρι κάποια απόσταση και ακτίνα, μιας και κάπως έτσι αντιλαμβάνονται και οι άνθρωποι τον κόσμο. Είναι δηλαδή λίγο άδικο να επιτρέπουμε στους εικονικούς χαρακτήρες να αντιλαμβάνονται γεγονότα που συμβαίνουν πίσω τους μόνο και μόνο επειδή βρίσκονται αρκετά κοντά στον παίκτη.

Αυτή η ιδέα θα μπορούσε να συνδυαστεί με την βελτίωση της προσαρμογής του χαρακτήρα στον παίκτη και έτσι να επιτρέψει κατά κάποιον τρόπο στους NPCs να μαθαίνουν από τα λάθη τους. Αυτή τη στιγμή στο περιβάλλον Moods ο Φίλος-NPC αναζητάει αντικείμενα σε μία περιορισμένη ακτίνα γύρω από τον χρήστη. Ας υποθέσουμε ότι η ακτίνα αυτή δεν είναι σταθερή αλλά αποτελεί ακόμη μία επιλογή που πρέπει να κάνει ο χαρακτήρας με κριτήριο πάντα την επίτευξη των δύο αντικρουόμενων στόχων του: α) να ακολουθεί τον παίκτη από κοντά για να μπορεί να ξέρει τις κινήσεις του και β) να ψάξει να βρει αντικείμενα του χρώματος που προτιμά ο παίκτης.

Με δεδομένο τώρα ότι ο χαρακτήρας όταν απομακρύνεται πολύ δεν ενημερώνεται μπορούμε να χρησιμοποιήσουμε το πλήθος των πληροφοριών που χάνει επιλέγοντας μία συγκεκριμένη ακτίνα δράσης για να αξιολογήσουμε την ποιότητα της απόφασής του μιας και η κατάλληλη ακτίνα δράσης δεν είναι κάτι στατικό αλλά εξαρτάται από τον τρόπο με τον οποίο παίζει ο παίκτης.

Αν για παράδειγμα έχουμε έναν παίκτη νωθρό που κινείται αργά (και αν μαζεύει και sadMoodies ακόμη αργότερα) και έχοντας καταλάβει την λειτουργία του Φίλου – NPC ο παίκτης αυτός προτιμάει να κάθεται και να τον αφήνει να κάνει όλη τη δουλειά, τότε ο χαρακτήρας μπορεί να αυξήσει (σταδιακά) την ακτίνα δράσης του, καθώς σίγουρα ακόμα και αν απομακρυνθεί η κατάσταση του παίκτη δεν θα έχει μεταβληθεί σημαντικά (ούτε από άποψη μετακίνησης στο χώρο, ούτε από άποψη συγκομιδής αντικειμένων). Στην αντίθετη περίπτωση όμως που ο παίκτης κινείται γρήγορα και θέλει να κάνει αυτός το κύριο μέρος της δουλειάς (και γιατί όχι, μπορεί

να αλλάζει και πιο συχνά γνώμη) τότε ο χαρακτήρας πρέπει να παραμένει κοντά του μιας και αν απομακρυνθεί πολύ θα δει όταν επιστρέψει ότι η κατάσταση του παίκτη είναι σημαντικά διαφορετική και άρα θα αξιολογήσει ως λανθασμένη την απόφασή του να αυξήσει την ακτίνα.

Ένας τέτοιος τρόπος σκέψης θα έδινε πολύ πιο δυναμικά χαρακτηριστικά στην συμπεριφορά των εικονικών χαρακτήρων και θα αύξαινε σημαντικά την πειστικότητά τους.

5.2 Χρήση του απομακρυσμένου ελέγχου για ένα Turing Test

Στην υλοποίησή μας ο εξυπηρετητής μας παρέχει την δυνατότητα να κινήσουμε τον παίκτη κάποιου πελάτη με όποιον τρόπο θέλουμε. Αυτό μπορεί να επεκταθεί ώστε να μπορεί να χειριστεί τεχνητά κάποιος εξωτερικός παράγοντας – άνθρωπος, έναν NPC, από τον εξυπηρετητή.

Μία τέτοια δυνατότητα θα ήταν χρήσιμη για να μπορέσουμε να ελέγξουμε κατά πόσο οι NPCs μας είναι αρκετά ρεαλιστικοί για να ξεγελάσουν έναν άνθρωπο. Μπορούμε δηλαδή να φανταστούμε ένα πείραμα τύπου Turing στο οποίο οι παίκτες που παίζουν με το Moods καλούνται στο τέλος (χωρίς να ξέρουν από πριν ότι θα τους γίνει αυτή η ερώτηση) να απαντήσουν αν κατά τη γνώμη τους οι NPCs ήταν πράγματι Non Player Characters ή αν τους χειριζόταν κάποιος άνθρωπος.

Ένα τέτοιο feedback θα ήταν σίγουρα σημαντικό καθώς πρέπει τελικά να ξέρουμε πόσο όλη αυτή η δουλειά που γίνεται προς μία κατεύθυνση είναι αποδοτική και έχει το επιθυμητό αποτέλεσμα: την δημιουργία δηλαδή εικονικών χαρακτήρων που προσομοιώνουν στον μέγιστο βαθμό την ανθρώπινη συμπεριφορά με τρόπο όμως ρεαλιστικό και πειστικό για τον άνθρωπο που έρχεται σε επαφή μαζί τους.

6 Γλωσσάρι

Avatar	Η εικόνα που αντιπροσωπεύει έναν άνθρωπο σε ένα εικονικό περιβάλλον
Animation	Ακολουθία κίνησης ενός χαρακτήρα στην οποία παρουσιάζεται μία απλή στοιχειώδης κίνηση όπως για παράδειγμα η κίνηση του κεφαλιού ή το περπάτημα. Οι ακολουθίες που διαθέτει ο χαρακτήρας είναι αυτές που του δίνουν ζωή και γι' αυτό επίσης καλούνται και εμψυχώσεις.
Chat	Τύπος επικοινωνίας δημοφιλής στα παιχνίδια με πολλούς παίκτες, όπου ανταλλάσσονται σύντομα μηνύματα σε έναν κατάλληλο χώρο στην οθόνη, και τα οποία είναι ορατά συνήθως σε όλους τους παίκτες.
Client-Server Architecture	Τρόπος διασύνδεσης εφαρμογών όπου οι σημαντικές λειτουργίες γίνονται σε έναν κεντρικό υπολογιστή (server) και οι clients απλά συνδέονται με αυτός.
Head Up Display (HUD)	Οι πληροφορίες που βρίσκονται πάντα διαθέσιμες στην οθόνη ενός παίκτη όταν αυτός παίζει ένα παιχνίδι (π.χ. σκορ, διαθέσιμη ενέργεια κλπ).
Gameplay	Όλες οι εμπειρίες του παίκτη όταν παίζει ένα παιχνίδι, καθώς και οι ενέργειές του μέσα στον κόσμο.
Graphical User Interface (GUI)	Το γραφικό περιβάλλον με το οποίο αλληλεπιδρά ένας χρήστης με μία εφαρμογή (μενού, κουμπιά κλπ).
Mission	Μία συγκεκριμένη αποστολή σε ένα παιχνίδι, συχνά έχει την έννοια της πίστας.
Non Player Character (NPC)	Ένας παίκτης που τον χειρίζεται ο υπολογιστής
Multiplayer Games	Παιχνίδια στα οποία πολλοί παίκτες μαζί κινούνται στον ίδιο χώρο και αλληλεπιδρούν ο ένας με τον άλλο. Συνήθως πρόκειται για δικτυακά παιχνίδια όπου ο κάθε παίκτης συνδέεται με έναν κεντρικό εξυπηρετητή.
Script Language	Κάποια γλώσσα προγραμματισμού με την οποία ελέγχεται απ' ευθείας ένα εκτελέσιμο πρόγραμμα χωρίς να χρειάζεται μεταγλώττιση. Τα scripts εκτελούνται συνήθως κατευθείαν από τα αρχεία κειμένου στα οποία είναι γραμμένα.

7 Βιβλιογραφικές Αναφορές

- [1]. Baillie-de Byl P. 2004, *Programming Believable Characters fro Computer Games*, Charles River Media.
- [2]. Bartneck C., 2002, *Integrating the OCC Model of Emotions in Embodied Characters*, *Proceedings of the Workshop on Virtual Conversational Characters: Applications, Methods, and Research Challenges*, Melbourne.
- [3]. Bunney G.; 2006, Romano T.; *AI Research*;
- [4]. Darryl C., *Enhancing Gameplay: Challenges for Artificial Intelligence in Digital Games*
- [5]. Eckschlager M.; Lankes M.; Bernhaupt R., *Real or Unreal? An evaluation Setting for Emotional Characters using Unreal*.
- [6]. Egges A.; Kshirsagar S.; Magnenat-Thalman N., *A Model for Personality and Emotion Simulation*, MIRALab - University of Geneva
- [7]. Ekman P.; Friesen W.V.; Ellsworth P., 1972, *Emotion in the human face: guidelines for research and an integration of findings*, New York: Pergamon Press.
- [8]. Finney K. C., 2005, *Advanced 3D game programming All in One*, Thomson Course Technology PTR
- [9]. Kaiser S.; Wehrle T.; Edwards P., 1994, *Multimodal emotion measurement in an interactive computer game*
- [10]. Maurina E. F., 2006, *The game programmer's guide to Torque*, GG Press
- [11]. Ortony A.; Clore G. L.; Collins, A. 1988. *The Cognitive Structure of Emotions*, Cambridge, UK: Cambridge University Press.
- [12]. Picard R. W., 1997, *Affective Computing*, MIT Press.
- [13]. Rabin S., 2005, *Introduction to game development*, Charles River Media
- [14]. Ruebenstrunk G., 1998, *Emotional Computers, Computer models of emotions and their meaning for emotion-psychological research*
- [15]. Schreirer J.; Fernandez R.; Klein J.; Picard R., *Frustrating the user on purpose: a Stepp toward building an affective computer*.
- [16]. Sweetser P.; Johnson D.; Sweetser J.; Wiles J., 2003, *Creating Engaging Artificial Characters for Games*, *Proceedings of the Second International Conference on Entertainment Computing*, Kluwer
- [17]. Weizenbaum J.; *ELIZA—a computer program for the study of natural language communication between man and machine*, *Communications of the ACM*, Volume 9, Issue 1, pp. 36 - 45, 1996.
- [18]. Zubek R.; Khoo A., *Making the Human Care: On Building Engaging Bots*

Χρήσιμες Ιστοσελίδες:

<http://garagegames.com/developer/torque/tge/>