



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΜΕΛΕΤΗΣ ΠΛΟΙΟΥ & ΘΑΛΑΣΣΙΩΝ ΜΕΤΑΦΟΡΩΝ

**Εφαρμογή Σχεδίασης Ναυπηγικών Γραμμών Με Χρήση Του
Λογισμικού Rhinoceros 3D**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Αθανάσιου Ντάλλα

Επιβλέπων : Γκίνης Αλέξανδρος

Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΜΕΛΕΤΗΣ ΠΛΟΙΟΥ & ΘΑΛΑΣΣΙΩΝ ΜΕΤΑΦΟΡΩΝ**

Εφαρμογή Σχεδίασης Ναυπηγικών Γραμμών Με Χρήση Του Λογισμικού Rhinoceros 3D

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Αθανάσιου Ντάλλα

Επιβλέπων : Γκίνης Αλέξανδρος
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Τριμελής εξεταστική επιτροπή:

1. Γκίνης Αλέξανδρος, Αναπληρωτής Καθηγητής Ε.Μ.Π.
2. Ζαραφωνίτης Γεώργιος, Καθηγητής Ε.Μ.Π.
3. Παπαδόπουλος Χρήστος, Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

Περίληψη

Ο σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός Plugin για το λογισμικό Rhinoceros 3D, το οποίο μέσω κατάλληλων εργαλείων θα μοντελοποιεί τη γάστρα ενός πλοίου χρησιμοποιώντας ναυπηγικές γραμμές. Το Plugin θα περιλαμβάνει μια εντολή που καλείται Shiplines και η οποία θα ανοίγει ένα παράθυρο (Interface), με όλα τα απαραίτητα εργαλεία για την παραγωγή και επεξεργασία των βασικών ναυπηγικών γραμμών.

Συγκεκριμένα, υλοποιήθηκαν εργαλεία που κατασκευάζουν καμπύλες από ένα σύνολο σημείων, που προέρχονται είτε από αρχεία που εισάγει ο χρήστης είτε από τομές άλλων καμπυλών με κάποιο επίπεδο καθ'υπόδειξη του χρήστη. Οι καμπύλες αυτές χωρίζονται σε τέσσερις βασικές κατηγορίες (Sections, Waterlines, Buttocks και Boundaries).

Επιπλέον το Shiplines περιλαμβάνει αρκετά εργαλεία επεξεργασίας των καμπυλών, προκειμένου ο χρήστης να πετύχει το επιθυμητό τελικό αποτέλεσμα στη σχεδιάσή του, όπως για παράδειγμα ο καθορισμός της γεωμετρικής συνέχειας σε ένα συγκεκριμένο σημείο παρεμβολής ή ο καθορισμός των συντεταγμένων ενός οποιουδήποτε σημείου.

Abstract

The scope of this thesis was the development of a Rhino Plugin, which, through appropriate tools, would model the hull of a ship using curves. The Plugin contains a command called Shiplines, that will allow the user to create and process the basic curves of a ship hull, through an interface containing all the necessary tools.

Specifically, a variety of classes and methods for creating curves from lists of points have been implemented. The creation is done either from the user through files containing lists of points or automatically from the intersection points of a plane and other curves, which is done by the software. These curves are divided into four basic categories (Sections, Waterlines, Buttocks and Boundaries).

In addition Shiplines includes several curve editing tools, which allow the user to achieve the desired end result for their design. These tools include for example specifying the geometric continuity at a particular intersection point of the curve or changing the original coordinates of a point.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον κ.Αλέξανδρο Γκίνη για την πολύτιμη βοήθειά του και τις συμβουλές του καθ'όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας, η οποία δε θα ήταν δυνατόν να ολοκληρωθεί χωρίς αυτόν.

Επίσης ευχαριστώ τον κύριο Παπατζανάκη Γιώργο για την βοήθεια, τις συμβουλές και την υποστήριξη που παρείχε.

Τέλος, ευχαριστώ την οικογένειά μου και τους φίλους μου που με στήριξαν όλο αυτόν τον καιρό.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Αντικείμενο διπλωματικής.....	2
1.2	Συνεισφορά.....	2
1.3	Οργάνωση κειμένου.....	3
2	Θεωρητικό υπόβαθρο.....	4
2.1	Παρεμβολή σημείων με τη χρήση καμπυλών B-Spline.....	4
2.2	Επιλογή παραμετροποίησης.....	5
2.3	Καθορισμός συνέχειας.....	6
3	Αρχιτεκτονική.....	7
3.1	Το Plugin.....	7
3.2	Αρχιτεκτονικό πρότυπο MVVM.....	8
3.2.1	Γενικά στοιχεία για το πρότυπο MVVM.....	8
3.2.2	Επιμέρους ανάλυση των τμημάτων του MVVM.....	9
3.3	Ανάλυση κλάσεων.....	10
3.3.1	Shiplines.....	13
3.3.2	GenericCurve.....	15
3.3.3	ExtendedCurve.....	16
3.3.4	ExtendedPoint.....	16
3.3.5	CurveCreation.....	17
3.3.6	FitFOBFOSPMB.....	18
3.3.7	ViewModelBase.....	18
3.3.8	ViewModelExtendedPoint.....	18
3.3.9	ViewModelGenericCurve.....	19
3.3.10	ViewModelMain.....	20
3.3.11	ViewModelCreateTab.....	22
3.3.12	ViewModelEditTab.....	23
3.3.13	ViewModelGeneralTab.....	24
3.3.14	ViewModelOptionsTab.....	25
3.3.15	CurveConduit.....	25
3.3.16	CurvatureGraph.....	25
3.3.17	ClearCollection.....	26

3.3.18	ProjectStorage.....	26
3.3.19	Readtxtpoints.....	26
3.3.20	RelayCommand.....	26
3.3.21	RemoveDuplicatesFromList.....	27
3.3.22	SortObservableCollectionClass.....	27
3.3.23	SortPointClass.....	27
3.3.24	TangentVector.....	28
3.3.25	AppearanceWindow.....	28
3.4	User Interface.....	28
3.5	Αρχεία εισόδου.....	32
3.6	Αρχεία αποθήκευσης.....	33
4	Υλοποίηση.....	34
4.1	Προγραμματιστικά εργαλεία και εγκατάσταση.....	34
5	Παράδειγμα λειτουργίας.....	35
5.1	Δεδομένα υπο σχεδίαση πλοίου.....	35
5.2	Αναλυτική διαδικασία δημιουργίας ναυπηγικών καμπυλών.....	36
6	Επίλογος.....	48
6.1	Σύνοψη και συμπεράσματα.....	48
6.2	Μελλοντικές επεκτάσεις.....	49
7	Βιβλιογραφία.....	50

1

Εισαγωγή

Τα τελευταία χρόνια και ιδιαίτερα τη δεκαετία που διανύουμε, χάρη στην ανάπτυξη της τεχνολογίας των υπολογιστών και της ηλεκτρονικής, παρατηρείται μια τεράστια και διαρκώς αυξανόμενη ανάπτυξη νέων τεχνικών σχεδίασης προϊόντων. Επιπλέον το έντονο ανταγωνιστικό περιβάλλον, επιβάλλει στις επιχειρήσεις γρήγορους, ακριβείς, καινοτόμους και με μικρό κόστος σχεδιασμούς για την κάλυψη των αναγκών της αγοράς, με αποτέλεσμα η δημιουργία αναλυτικών ψηφιακών μοντέλων να γίνεται όλο και πιο σημαντική. Τα συστήματα σχεδίασης με τη βοήθεια H/Y (CAD) έχουν πλέον παραμερίσει παραδοσιακές προσεγγίσεις σχεδίασης προϊόντων, όπως η σχεδίαση με το χέρι και έχουν γίνει πλέον αναπόσπαστο τμήμα της σύγχρονης προσέγγισης σχεδίασης και παραγωγής σε πολλούς διαφορετικούς τομείς όπως: ο βιομηχανικός σχεδιασμός, η ναυπηγική, η αεροναυπηγική, ο αρχιτεκτονικός σχεδιασμός κ.α.

Πιο συγκεκριμένα στη ναυπηγική ένα από τα βασικά βήματα στη σχεδίαση ενός πλοίου μέσω H/Y είναι ο καθορισμός της μορφής της γάστρας του πλοίου, αρχικά με ναυπηγικές γραμμές και στη συνέχεια με επιφάνειες. Τα συστήματα CAD επιτρέπουν τη βελτιστοποίηση της σχεδίασης με διάφορα κριτήρια, όπως για παράδειγμα η ελαχιστοποίηση της αντίστασης του πλοίου, η ελαχιστοποίηση του βάρους της μεταλλικής κατασκευής ή η μεγιστοποίηση της επιβιωσιμότητας μετά από βλάβη κλπ.

Ένα τέτοιο σύγχρονο ολοκληρωμένο ναυπηγικό λογισμικό τρισδιάστατης μοντελοποίησης πλοίων είναι το Rhinoceros®. Ένα βασικό χαρακτηριστικό του εν λόγω προγράμματος μοντελοποίησης είναι η ευκολία δημιουργίας και επεξεργασίας καμπυλών ελευθέρως μορφής, όπως οι ναυπηγικές καμπύλες της γάστρας του πλοίου. Ένα άλλο εξίσου σημαντικό χαρακτηριστικό είναι η δυνατότητα δημιουργίας Plugins, δηλαδή λογισμικών που επιτρέπουν την προγραμματιστική υλοποίηση επιπλέον εντολών, λειτουργιών και δυνατοτήτων τόσο για την αυτοματοποίηση διαδικασιών σχεδίασης αλλά και για την αποφυγή χρήσης διαφορετικών λογισμικών πακέτων κατά τη διάρκεια της προμελέτης ενός προϊόντος όπως ένα πλοίο.

Κάποια από τα πιο γνωστά Plugin για το Rhinoceros είναι το Grasshopper, το οποίο αναπτύχθηκε από την εταιρία που δημιούργησε και το ίδιο το Rhinoceros Robert McNeel & Associates και βρίσκει ποικίλες εφαρμογές λόγω της ευκολίας χρήσης του αλλά και λόγω των

επιπλέον δυνατοτήτων που παρέχει ιδιαίτερα στον αρχιτεκτονικό σχεδιασμό. Ένα άλλο πολύ σημαντικό Plugin που χρησιμοποιείται αποκλειστικά για ναυπηγικές εφαρμογές είναι το Orca3D®, το οποίο παρέχει λύσεις για τη σχεδίαση πλοίων, τους απαραίτητους υδροστατικούς υπολογισμούς του καθώς και εκτιμήσεις για την αντίσταση πρόωσης και ισχύ ρυμούλκησης και τέλος του συνολικού κόστους της κατασκευής του σκάφους.

Γενικά η σχεδίαση ενός πλοίου είναι ένα ιδιαίτερα σύνθετο και πολύπλοκο ζήτημα και πολλές φορές μπορεί τα υπάρχοντα ναυπηγικά λογισμικά πακέτα να μην μπορούν να καλύψουν τις ανάγκες για σχεδίαση σύγχρονων και πρωτοπόρων σκαφών. Γι'αυτό το λόγο είναι επιτακτική ανάγκη η συνεχής εξέλιξη των λογισμικών σχεδίασης και μελέτης που θα βελτιστοποιούν με ποικίλα κριτήρια κάθε φορά τη σχεδίαση και θα παρέχουν λύσεις που θα εξασφαλίζουν τον ανταγωνισμό μεταξύ των κατασκευαστικών εταιριών.

1.1 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία λοιπόν καταπιάνεται ακριβώς με αυτό, το πρόβλημα της δημιουργίας ενός Plugin για το Rhinoceros το οποίο θα καλύπτει τις ανάγκες του χρήστη για γρήγορη και εύκολη σχεδίαση των ναυπηγικών γραμμών ενός πλοίου. Το συγκεκριμένο Plugin θα μειώνει δραστικά τον χρόνο καθώς και τη διαδικασία εκμάθησης που θα χρειαζόταν κάποιος για να δημιουργήσει το πλέγμα καμπυλών ενός πλοίου χρησιμοποιώντας μόνο τα εργαλεία που παρέχει το Rhinoceros από μόνο του.

Για την επίτευξη αυτού του στόχου η παρούσα εργασία μελετά τη διαδικασία κατασκευής και επεξεργασίας καμπυλών από την πλευρά του προγραμματιστή, καθώς και τη διαδικασία δημιουργίας των απαραίτητων εργαλείων που θα είναι διαθέσιμα σε ένα παράθυρο (interface) για την υλοποίηση από της πλευράς του χρήστη της σχεδίασης. Το Plugin αυτό θα μπορούσε να θεωρηθεί και ένα ολοκληρωμένο πρόγραμμα παραγωγής των ναυπηγικών γραμμών ενός πλοίου, καθώς περιέχει σχεδόν όλα τα απαραίτητα εργαλεία για τη σχεδίαση γαστρών συμβατικών πλοίων. Στη συνέχεια ο χρήστης αν επιθυμεί και την δημιουργία των επιφανειών θα μπορεί να χρησιμοποιήσει τα εργαλεία που παρέχει το Rhinoceros.

1.2 Συνεισφορά

Πιο συγκεκριμένα κατά τη διαδικασία εκπόνησης της εργασίας μελετήθηκαν διάφορες πτυχές του προβλήματος σχεδίασης πλοίων και πως θα μπορούσε ένα πρόγραμμα να μειώσει τον χρόνο εκμάθησης που θα χρειαζόταν ένας καινούργιος χρήστης αλλά και να επιταχύνει τον ρυθμό σχεδίασης ενός πιο έμπειρου. Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Δημιουργία ενός interface WPF (Windows Presentation Foundation) που θα λειτουργεί σε υπολογιστές με λειτουργικό σύστημα Windows στους οποίους θα υπάρχει και το Rhino 5.

2. Ως γλώσσα προγραμματιστικής υλοποίησης επιλέχθηκε η C#, μια ιδιαίτερα σύγχρονη και διαρκώς εξελισσόμενη γλώσσα, που έχει αναπτυχθεί από τη Microsoft.
3. Καταστρώθηκαν αλγόριθμοι και μεθοδολογίες για την δημιουργία και επεξεργασία των ναυπηγικών γραμμών ενός πλοίου.
4. Το Plugin που δημιουργήθηκε είναι πλήρως επεκτάσιμο.
5. Το πρόγραμμα εκτελεί πολλές διαδικασίες αυτοματοποιημένα γλιτώνοντας χρόνο από τον χρήστη.
6. Το τελικό αποτέλεσμα μπορεί να μετατραπεί σε μορφή επεξεργάσιμη από το Rhinocerps (export) εάν οι δυνατότητες του Plugin δεν καλύπτουν τον χρήστη.

1.3 Οργάνωση κειμένου

Το παρών κεφάλαιο είναι μια σύντομη εισαγωγή στο χώρο του προβλήματος και δίνει τον ορισμό του στα πλαίσια της εργασίας μαζί με μερικά στοιχεία γι' αυτήν. Στη συνέχεια στο Κεφάλαιο 2 θα παρουσιαστούν οι θεωρητικές γνώσεις που απαιτούνται για την κατανόηση της θεωρίας πίσω από τα εργαλεία της εφαρμογής. Στο Κεφάλαιο 3 γίνεται αναλυτική παρουσίαση της αρχιτεκτονικής του συστήματος και των τμημάτων του. Το Κεφάλαιο 4 αφορά την υλοποίηση, το Κεφάλαιο 5 μελετά τη σχεδίαση των καμπυλών ενός συμβατικού πλοίου χρησιμοποιώντας το Plugin και το Κεφάλαιο 6 συνοψίζει τα αποτελέσματα στα οποία κατέληξε η εργασία. Το Κεφάλαιο 7 περιέχει την βιβλιογραφία που αξιοποιήθηκε.

2

Θεωρητικό υπόβαθρο

Σε αυτό το σημείο θα γίνει μια συνοπτική παρουσίαση των θεωρητικών περιοχών οι οποίες είναι συναφείς με το αντικείμενο της διπλωματικής. Η συλλογιστική με την οποία έγινε ο κώδικας ήταν η δημιουργία καμπυλών από σημεία παρεμβολής και η παρέμβαση για τον καθορισμό της συνέχειας στα σημεία αυτά.

2.1 Παρεμβολή σημείων με τη χρήση καμπυλών B-Spline

Σε αυτό το σημείο της διπλωματικής θα γίνει μια σύντομη περιγραφή του προβλήματος δημιουργίας μιας καμπύλης που διέρχεται από τα σημεία p_0, p_1, \dots, p_n με $n > 1$. Προκειμένου να κατασκευάσουμε μια καμπύλη B-Spline, πρέπει να προσδιορίσουμε:

- Την τάξη της καμπύλης (k).
- Τον αριθμό και τις συντεταγμένες των σημείων ελέγχου.
- Το κομβοδιάγραμμα.

Μια καμπύλη B-Spline με $n + 1$ σημεία ελέγχου μπορεί να χρησιμοποιηθεί για την παρεμβολή $n + 1$ δεδομένων σημείων $p_i, i = 0, \dots, n$. Αν η καμπύλη έχει ακριβώς $n + 1$ σημεία ελέγχου, οι βαθμοί ελευθερίας της (τα άγνωστα σημεία ελέγχου) συμπίπτουν ακριβώς με τον αριθμό των εξισώσεων που καλούμαστε να επιλύσουμε, ενώ στην περίπτωση επιπλέον σημείων ελέγχου διαθέτουμε επιπλέον βαθμούς ελευθερίας για να διαμορφώσουμε το σχήμα της καμπύλης.

Στην παρούσα διπλωματική επιλέγουμε ως τάξη των καμπυλών $k = 3$. Τα πολυωνυμικά κομμάτια $3^{\text{ου}}$ βαθμού είναι τα μικρότερα βαθμού πολυώνυμα που μπορούν να εξασφαλίσουν συνέχεια $2^{\text{ης}}$ τάξης (C^2) και ταυτόχρονα είναι αρκετά ευέλικτα, καθώς είναι τα μικρότερου βαθμού πολυώνυμα με δυνατότητα ύπαρξης σημείου καμπής.

Όσον αφορά το κομβοδιάγραμμα, αφού θα χρησιμοποιήσουμε $n + 1$ σημεία ελέγχου σε μια καμπύλη k τάξης, είναι προφανές ότι ο αριθμός των κόμβων θα είναι $n + k + 1$, δηλαδή για

καμπύλες $3^{\text{ου}}$ βαθμού που μελετάμε στην παρούσα διπλωματική $n + 5$. Ο βαθμός d της καμπύλης συνδέεται με την τάξη της καμπύλης με την $d = k - 1$.

2.2 Επιλογή παραμετροποίησης

Η επιλογή της παραμετροποίησης έχει να κάνει με την επιλογή του μήκους των υποδιαστημάτων στο κομβοδιάγραμμα. Υπάρχουν αρκετές επιλογές παραμετροποίησης όπως η ομοιόμορφη παραμετροποίηση, η centripetal και η χορδική παραμετροποίηση την οποία θα χρησιμοποιήσουμε στα πλαίσια της διπλωματικής εργασίας για την κατασκευή των καμπυλών.

Στην χορδική παραμετροποίηση επιθυμούμε το συνολικό κομβοδιάγραμμα να κινείται στο διάστημα $[0,1]$. Η παραμετροποίηση ορίζεται ως εξής:

$$u_0 = u_1 = \dots = u_k - 1 = 0$$

$$u_{n+1} = u_{n+2} = \dots = u_{n+k} = 1$$

και

$$u_j = \frac{\|b_{j-k+1} - p_{j-k}\|_2}{\sum_{i=k}^{n+1} \|b_{i-k+1} - p_{i-k}\|_2}$$

όπου με $\|\cdot\|_2$ συμβολίζουμε την ευκλείδεια απόσταση. Πιο γενικά μπορούμε να γράψουμε για τις σχετικές αποστάσεις των υποδιαστημάτων ότι θα πρέπει να ισχύει:

$$\frac{\Delta_i}{\Delta_{i+1}} = \frac{\|\Delta p_i\|_2}{\|\Delta p_{i+1}\|_2}$$

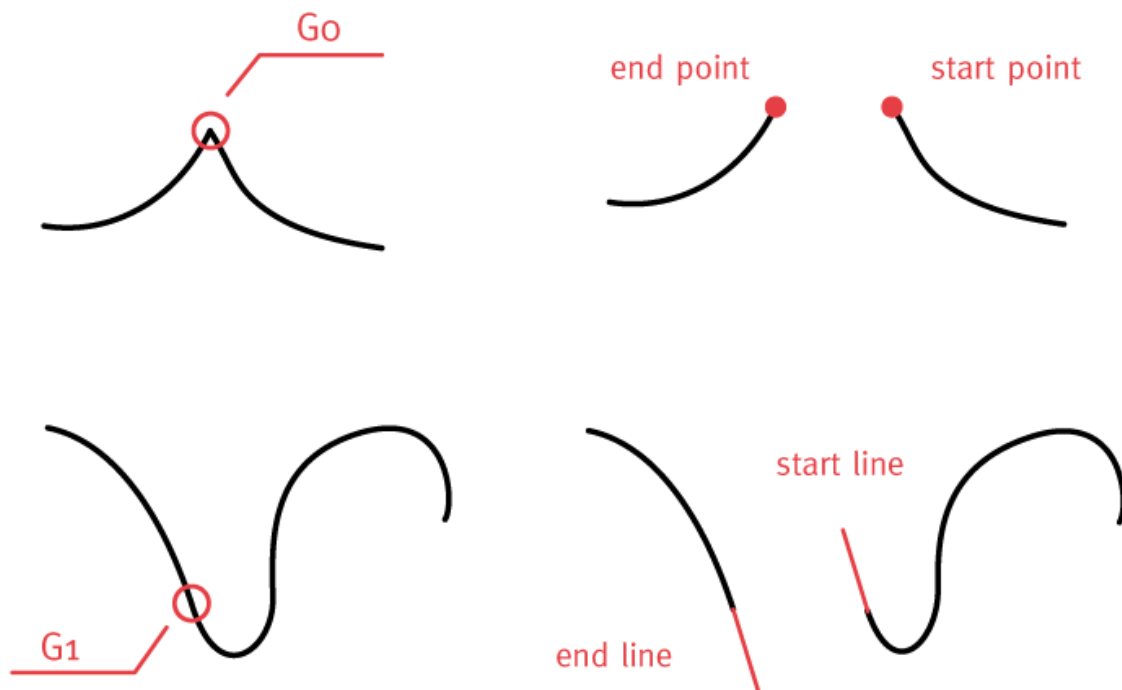
Δηλαδή, με εξαίρεση τις πολλαπλότητες στα άκρα, η χορδική παραμετροποίηση ορίζει τα παραμετρικά διαστήματα σε μήκος ανάλογο του ευθύγραμμου τμήματος που ενώνει δύο διαδοχικά σημεία ως προς το συνολικό μήκος της πολυγωνικής γραμμής που περνά απ' όλα τα σημεία.

2.3 Καθορισμός συνέχειας

Χρησιμοποιώντας πολλές καμπύλες B-Spline σε αλληλουχία έχουμε τη δυνατότητα να αποκτήσουμε τοπικό έλεγχο. Κάτι τέτοιο υλοποιείται μοιράζοντας τα δύο ακραία σημεία ελέγχου σε γειτονικές καμπύλες, κάτι που προφανώς επηρεάζει το πρώτο και τελευταίο τμήμα της κάθε καμπύλης. Με αυτόν τον τρόπο είναι αναπόφευκτο πως θα δημιουργηθούν ασυνέχεις στο κοινό σημείο ελέγχου.

Στην κοινή κορυφή η συνέχεια που θα έχουμε είναι G^0 με αποτέλεσμα η μετάβαση μεταξύ των συμμετεχόντων τμημάτων να μην είναι ομαλή. Εάν ορίσουμε τις εφαπτομένες των δύο καμπυλών στο κοινό σημείο να είναι ίσες τότε έχουμε συνέχεια G^1 .

Η παραπάνω ανάλυση φαίνεται και πιο καθαρά στο παρακάτω σχήμα:



Σχήμα 2.1: Παραδείγματα G^0 και G^1 συνέχειας

3

Αρχιτεκτονική

Σε αυτό το κεφάλαιο περιγράφεται η αρχιτεκτονική του προγράμματος και γίνεται αναλυτική παρουσίαση του σχεδιασμού του. Θα γίνει παρουσίαση των μεθόδων, κλάσεων καθώς την τρόπο σύνδεσης των παραπάνω με το παράθυρο (interface) που θα χειρίζεται ο χρήστης.

3.1 To Plugin

Το Plugin ουσιαστικά αποτελείται από μια εντολή (command) του Rhinoceros την «Shiplines», η οποία όταν δοθεί ανοίγει ένα παράθυρο παράλληλα με το κύριο πρόγραμμα που περιέχει Tabs, Buttons, ComboBoxes, CheckBoxes και οτιδήποτε άλλο χρειάζεται ο χρήστης για να επεξεργαστεί και να δημιουργήσει ναυπηγικές γραμμές. Το Plugin προϋποθέτει όχι μόνο την ύπαρξη του Rhinoceros 5 στον υπολογιστή αλλά και τη λειτουργία του καθ'όλη τη διάρκεια που θέλουμε να το χρησιμοποιήσουμε. Σε περίπτωση που τερματίσει η λειτουργία του Rhinoceros τερματίζει αντίστοιχα και η λειτουργία του Shiplines.

Σε επίπεδο λογισμικού κάθε εργαλείο που υπάρχει στο παράθυρο συνδέεται είτε με κάποια μέθοδο που υπάρχει στον κώδικα είτε με κάποια λιστα στην οποία αποθηκεύονται τα αντικείμενα όπως οι καμπύλες ή τα σημεία που δημιουργούνται από τον χρήστη. Παρ'όλο που τα δεδομένα αυτά αποθηκεύονται σε λίστες ανεξάρτητες του Rhinoceros, οποιαδήποτε αλλαγή πραγματοποιηθεί απεικονίζεται άμεσα και αυτόματα στα Viewports του.

Ο κώδικας χωρίζεται όπως θα δούμε και παρακάτω σε τρεις κατηγορίες, όπως προδιαγράφεται από το MVVM (Model-View-ViewModel) πρότυπο αρχιτεκτονικής και επομένως κάθε κλάση έχει κατηγοριοποιηθεί ανάλογα με το ρόλο που κατέχει στην λειτουργία του προγράμματος είτε ως Model είτε ως ViewModel, είτε ως View.

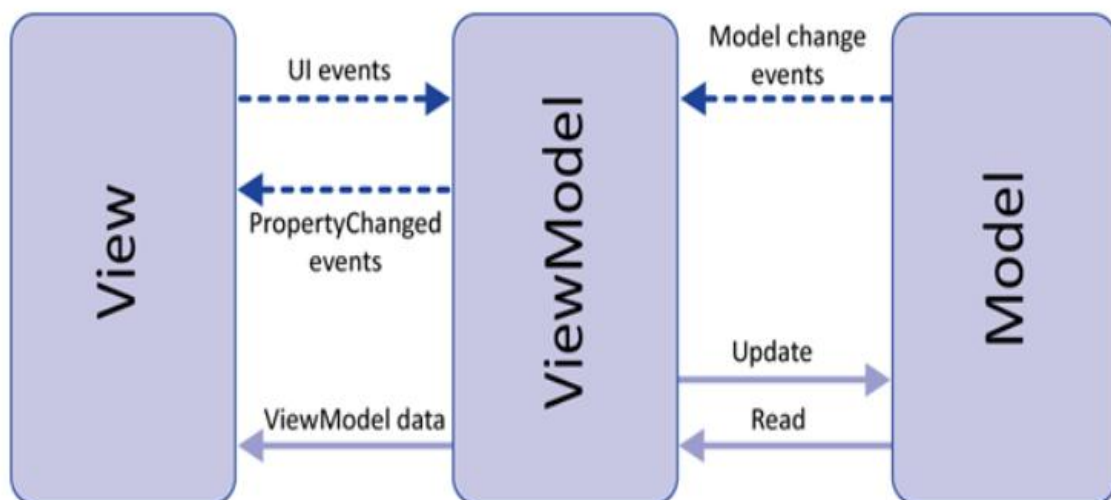
3.2 Αρχιτεκτονικό πρότυπο MVVM

3.2.1 Γενικά στοιχεία για το πρότυπο MVVM

Παραδοσιακά η ανάπτυξη παραθύρων (interfaces) γινόταν με την σύνδεση όλης της λογικής (logical code) με το παράθυρο με τέτοιο τρόπο ώστε να μην γίνεται κάποια στιγμή στο μέλλον να αποσυμπλεχθούν αυτά τα δύο είτε για να γίνουν αλλαγές στη λογική είτε για να γίνουν αλλαγές μικρές και μεγάλες στο παράθυρο. Επιπλέον, η άμεση σύμπλεξη της λογικής με το παράθυρο επιβράδυνε τον ρυθμό ανάπτυξης των προγραμμάτων, καθώς δεν ήταν δυνατόν δύο προγραμματιστές να δουλεύουν ταυτόχρονα στο ίδιο παράθυρο και επιπλέον τα λάθη ενός προγραμματιστή επιβάρυναν και δημιουργούσαν δυσκολίες σε όλους τους υπόλοιπους.

Για να αποφευχθούν αυτά τα προβλήματα αναπτύχθηκαν διάφορα αρχιτεκτονικά πρότυπα που διαχώριζαν τη λογική από το παράθυρο όπως το MVC και αργότερα το MVVM. Στο MVVM που επικεντρώνεται και η συγκεκριμένη διπλωματική ο κώδικας που συνδέει την λογική, που αναφέρεται και ως Model, με το παράθυρο, που αντίστοιχα αναφέρεται και ως View, καλείται ViewModel και έχει ως σκοπό να συνδέει το Model με το View, χωρίς όμως να δημιουργεί σχέση εξάρτησης, με αποτέλεσμα η διαχείριση, ανάπτυξη και συντήρηση του κώδικα να είναι πολύ πιο εύκολη.

Για παράδειγμα εάν αλλάξει κάποιο Property που βρίσκεται στο ViewModel, τότε το View ενημερώνεται αυτόματα μέσω DataBinding και Notification. Αν ο χρήστης πραγματοποιήσει αλλαγές στο View, όπως για παράδειγμα το export των καμπυλών που έχει φτιάξει σε ένα ξεχωριστό αρχείο .3dm τότε εκτελείται μια εντολή Command στο ViewModel για να εκτελεσει την ζητούμενη μέθοδο. Σε περίπτωση που θέλαμε να αλλάξουμε κάποια δεδομένα στο Model θα ήταν αντίστοιχα το ViewModel αυτό που θα πραγματοποιούσε τις αλλαγές. Επομένως, οι κλάσεις που αφορούν το View δεν 'γνωρίζουν' την ύπαρξη των κλάσεων που απαρτίζουν το Model.



3.2.2 Επιμέρους ανάλυση των τμημάτων του MVVM

Σε αυτό το σημείο γίνεται μια σύντομη ανάλυση του σκοπού των τμημάτων του αρχιτεκτονικού προτύπου MVVM που αναφλερθηκαν και στις προηγούμενες παραγράφους.

3.2.2.1 Model

Το Model αποτελείται από τα δεδομένα ή την πληροφορία που μας ενδιαφέρει καθώς και τη λογική του προγράμματος. Ο τρόπος με τον οποίο εκτίθεται το Model στο ViewModel είναι μέσω κατάλληλων Notifications είτε του χρήστη είτε ενσωματωμένα σε λίστες, όπως τα ObservableCollections.

3.2.2.2 View

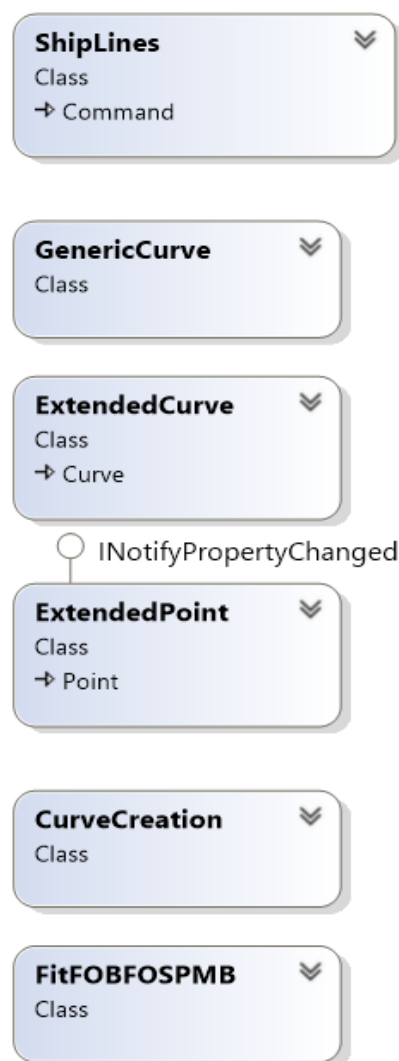
Το View είναι το κομμάτι του κώδικα με το οποίο αλληλεπιδρά ο χρήστης και ουσιαστικά προβάλλει τα δεδομένα μέσω κάποιου κατάλληλου εργαλείου (Control). Το View επίσης περιέχει και μεθόδους για τον έλεγχο των δεδομένων εισαγωγής του χρήστη, όπως για παράδειγμα η μη-αποδοχή χαρακτήρων σε κελιά που περιέχουν αριθμούς ή η απενεργοποίηση εργαλείων που σε άλλη περίπτωση θα οδηγούσαν σε σφάλμα και στον τερματισμό του προγράμματος.

3.2.2.3 ViewModel

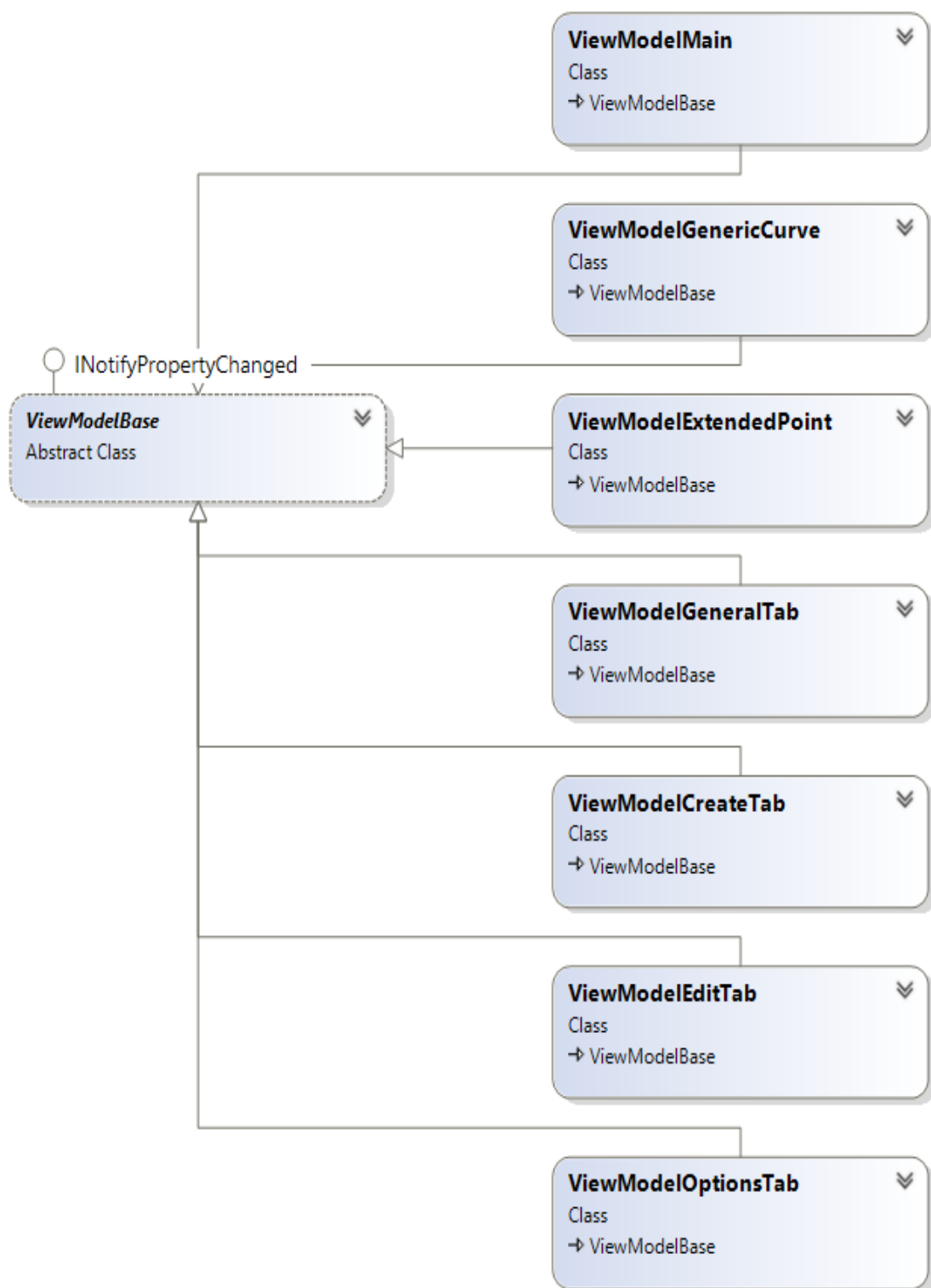
Το ViewModel αποτελεί τον συνδετικό κρίκο του Model με το View, χωρίς να εκθέτει οποιαδήποτε πληροφορία ή διαδικασία του ενός στο άλλο. Το ViewModel περιέχει κλάσεις, μεθόδους και εντολές βοηθητικές για το View, όπως η πρόκληση κάποιου Event και άλλες που διαμορφώνουν την πληροφορία στο Model ως αποτέλεσμα αλλαγών που έχει κάνει ο χρήστης στο View.

3.3 Ανάλυση κλάσεων

Στην προηγούμενη παράγραφο έγινε μια πιο γενική ανάλυση του προγράμματος και σε ποια τμήματα χωρίζεται ανάλογα με τη λειτουργία. Τώρα θα παρουσιαστούν αναλυτικά οι συνιστώσες κάθε τμήματος, ο τρόπος διασύνδεσής τους καθώς και διάφορες σχεδιαστικές λεπτομέρειες.



Σχήμα 3.2: Διάγραμμα κλάσεων του Model



Σχήμα 3.3: Διάγραμμα κλάσεων του ViewModel



Σχήμα 3.4: Διάγραμμα υπόλοιπων κλάσεων Helpers, Controls, Appearance και RhinoDisplay

Όπως παρατηρούμε και στο πιο πάνω σχήμα στον κώδικα υπάρχουν και κλάσεις που δεν ανήκουν στο Model ή στο ViewModel. Οι κλάσεις αυτές έχουν βοηθητικό ρόλο στο πρόγραμμα και ανήκουν σε μια από τις εξής τέσσερις κατηγορίες: Helpers, Controls Appearance ή RhinoDisplay.

3.3.1 Shiplines

Η κλάση Shiplines ανήκει στο Model και κληρονομεί τις μεθόδους και τις ιδιότητες της βασικής κλάσης Command που έχει δημιουργηθεί από τους δημιουργούς του Rhinoceros. Στη συγκεκριμένη κλάση περιέχονται οι περισσότερες μέθοδοι για την δημιουργία και επεξεργασία των καμπυλών. Περιέχει τα εξής πεδία:

_datagridpointslist: Λίστα με τα σημεία της καμπύλης που επεξεργαζόμαστε.

_windowInstance: Το παράθυρο που δημιουργείται μέσω της εντολής Shiplines.

B: Τιμή του πλάτους του υπο σχεδίαση πλοίου.

Boundaries: Λίστα που περιέχει ναυπηγικές καμπύλες τύπου Boundary.

Buttocks: Λίστα που περιέχει ναυπηγικές καμπύλες τύπου Buttock.

currentDoc: Το αρχείο του Rhinoceros στο οποίο μπορούμε να προσθέσουμε σημεία, καμπύλες κ.α.

curvecolor: Το χρώμα με το οποίο φαίνεται στα Viewport του Rhinoceros η καμπύλη που θέλουμε να επεξεργαστούμε.

curvetoedit: Η καμπύλη που επεξεργαζόμαστε.

D: Τιμή του κοίλου του υπο σχεδίαση πλοίου.

L: Τιμή του μήκους μεταξύ καθέτων του υπο σχεδίαση πλοίου.

objMoved: Είναι μεταβλητή που καθορίζει αν κάποιο σημείο καμπύλης έχει μετακινηθεί.

PMBAftLocation: Τιμή της θέσης του παράλληλου τμήματος του υπο σχεδίαση πλοίου πρύμα της μέσης τομής.

PMBForLocation: Τιμή της θέσης του παράλληλου τμήματος του υπο σχεδίαση πλοίου πλώρα της μέσης τομής.

pointsDeleted: Λίστα με σημεία που έχουν διαγραφεί.

pointsTransformed: Λίστα με σημεία που έχουν μετακινηθεί.

Sections: Λίστα που περιέχει ναυπηγικές καμπύλες τύπου Section.

Waterlines: Λίστα που περιέχει ναυπηγικές καμπύλες τύπου Waterline.

Επιπλέον περιέχει τις ακόλουθες μεθόδους:

_datagridpointslist_CollectionChanged: Εκτελείται όταν μεταβάλεται η λίστα των σημείων την καμπύλης που επεξεργαζόμαστε είτε με προσθήκη είτε με αφαίρεση σημείου και ενημερώνει την αντίστοιχη στο ViewModel.

AddPMBClick: Μέθοδος που προσθέτει τις καμπύλες που οριοθετούν το παράλληλο τμήμα του πλοίου.

Boundaries_CollectionChanged: Ενημερώνει την αντίστοιχη λίστα Boundaries στο ViewModel όταν κάποια καμπύλη προσθέτεται ή αφαιρείται από τη λίστα Boundaries της κλάσης Shiplines.

Buttocks_CollectionChanged: Ενημερώνει την αντίστοιχη λίστα Buttock στο ViewModel όταν κάποια καμπύλη προσθέτεται ή αφαιρείται από τη λίστα Buttocks της κλάσης Shiplines.

CreateCurves: Δημιουργία καμπυλών είτε με κάποιο βήμα σε ένα συγκεκριμένο διάστημα, είτε σε κάποια θέση που θα ορίσει ο χρήστης.

CreateShaftCurve: Κατασκευάζει το ημικύκλιο της χοάνης της έλικας.

CurvatureHairDensity: Καθορίζει την πυκνότητα των διανυσμάτων καμπυλότητας.

CurvatureHairLength: Καθορίζει το μήκος των διανυσμάτων καμπυλότητας.

CurvatureHide: Όταν επεξεργαζόμαστε μια ναυπηγική καμπύλη αποκρύπτει τα διανύσματα καμπυλότητας όλων των καμπυλών που την απαρτίζουν.

CurvatureShow: Όταν επεξεργαζόμαστε μια ναυπηγική καμπύλη εμφανίζει τα διανύσματα καμπυλότητας όλων των καμπυλών που την απαρτίζουν.

DoneEdit: Όταν ολοκληρωθεί η επεξεργασία μιας ναυπηγικής καμπύλης επαναφέρει τα Viewports στα αρχικά του Rhinoceros, κρύβει τα σημεία παρεμβολής της καμπύλης και αφαιρεί την δυνατότητα επεξεργασίας της.

EditMode: Ορίζει ανάλογα με τον τύπο της ναυπηγικής καμπύλης το Viewport και εμφανίζει τα σημεία παρεμβολής για να μπορεί ο χρήστης να τα επεξεργαστεί.

Export3dmFile: Εξάγει ένα αρχείο .3dm με τις ναυπηγικές καμπύλες που έχουν δημιουργηθεί για να τις επεξεργαστεί κάποιος χρησιμοποιώντας τα εργαλεία του Rhinoceros.

FitClick: Η επιλεγμένη κατηγορία ναυπηγικών καμπυλών ξαναδημιουργείται έτσι ώστε τα σημεία παρεμβολής να

OnCurvetoeditPointPropertyChanged: Εκτελείται κάθε φορά που αλλάζει μια ιδιότητα ενός σημείου της ναυπηγικής καμπύλης και ενημερώνει το αντίστοιχο σημείο που υπάρχει στη λίστα του ViewModel ώστε να παραμένουν συγχρονισμένες.

OnPlaceholderPointPropertyChanged: Εκτελείται όταν αλλάζει μια ιδιότητα κάποιου σημείου στην λίστα που βρίσκεται στο ViewModel όταν επεξεργαζόμαστε την καμπύλη.

OnRhinoDeleteObject: Εκτελείται κάθε φορά που διαγράφεται ένα σημείο.

OnRhinoIdle: Εκτελείται επαναλαμβανόμενα όταν επεξεργαζόμαστε κάποια καμπύλη προκειμένου οποιαδήποτε αλλαγή να αποτυπώνεται αυτόματα στο αντίστοιχο Viewport.

OnRhinoMoveObject: Εκτελείται κάθε φορά που μετακινείται ένα σημείο.

OpenFile: Ανοίγει και διαβάζει ένα αρχείο .txt με ένα σύνολο συντεταγμένων για να κατασκευάσει και να εμφανίσει μια καμπύλη.

RangedCurveCreation: Κατασκευάζει ναυπηγικές καμπύλες σε ένα διάστημα με ένα συγκεκριμένο βήμα με σημεία που έχουν προκύψει από την τομή ενός επιπέδου με άλλες ναυπηγικές καμπύλες διαφορετικού τύπου.

Sections_CollectionChanged: Ενημερώνει την αντίστοιχη λίστα Section στο ViewModel όταν κάποια καμπύλη προσθέτεται ή αφαιρείται από τη λίστα Sections της κλάσης Shiplines.

SetKnuckleAngle: Ορίζει τις γωνίες που θα έχουν οι δύο καμπύλες στο κοινό σημείο, όταν υπάρχει G^0 συνέχεια, δηλαδή Knuckle.

SetPoint: Καθορίζει μια επιπλέον ιδιότητα του σημείου αν θα είναι ένα απλό σημείο παρεμβολής Ordinary, αν θα έχουμε συνέχεια G^0 Knuckle, ή αν θα έχουμε G^1 συνέχεια Tangent.

SetShipDimensions: Ορίζει τις βασικές διαστάσεις (μήκος, πλάτος κοίλο) του πλοίου.

Waterlines_CollectionChanged: Ενημερώνει την αντίστοιχη λίστα Waterline στο ViewModel όταν κάποια καμπύλη προσθέτεται ή αφαιρείται από τη λίστα Waterlines της κλάσης Shiplines.

3.3.2 GenericCurve

Η κλάση GenericCurve περιέχει όλη την πληροφορία που χρειάζεται για να περιγραφεί πλήρως μια ναυπηγική γραμμή. Ανήκει στο Model και περιέχει τα εξής πεδία:

BoundaryColor: Το χρώμα με το οποίο θα προβάλλονται οι καμπύλες Boundary στο Rhinoceros.

ButtockColor: Το χρώμα με το οποίο θα προβάλλονται οι καμπύλες Buttock στο Rhinoceros.

SectionColor: Το χρώμα με το οποίο θα προβάλλονται οι καμπύλες Section στο Rhinoceros.

WaterlineColor: Το χρώμα με το οποίο θα προβάλλονται οι καμπύλες Waterline στο Rhinoceros.

Περιέχει τις εξής ιδιότητες:

Curve: Λίστα με τις καμπύλες που απαρτίζουν μια ναυπηγική γραμμή.

CurveCategory: Η κατηγορία στην οποία ανήκει η συγκεκριμένη καμπύλη, δηλαδή Section, Waterline, Buttock ή Boundary.

CurveName: Το όνομα της ναυπηγικής καμπύλης που για τις Sections, Waterlines και Buttocks ταυτίζεται με τη θέση, ενώ οι καμπύλες Boundary έχουν προκαθορισμένα ονόματα.

CurvePlane: Το επίπεδο στο οποίο βρίσκεται η καμπύλη.

CurvePoint3dList: Λίστα με σημεία παρεμβολής της καμπύλης.

CurvePointGuidList: Λίστα με το μοναδικό ID που έχει κάθε σημείο.

CurvePosition: Η θέση της καμπύλης ως προς σύστημα συντεταγμένων ορισμένο από τον χρήστη.

ListCurvatureGraph: Λίστα με τις καμπυλότητες των επιμέρους καμπυλών της ναυπηγικής καμπύλης.

ListCurveConduit: Λίστα με τις απεικονίσεις των καμπυλών στο Rhinoceros.

ViewportName: Το όνομα του Viewport στο οποίο ανήκει η καμπύλη.

Περιέχει επιπλέον τις εξής μεθόδους:

CreateBoundaryPlane: Η μέθοδος αυτή ορίζει το επίπεδο στο οποίο θα ανήκει κάθε Boundary ναυπηγική γραμμή.

CreateCurveConduit: Η μέθοδος αυτή κατασκευάζει την απεικόνιση των καμπυλών στο Rhinoceros.

CreateCurvePlane: Η μέθοδος καθορίζει το επίπεδο στο οποίο θα ανήκει κάθε ναυπηγική γραμμή, ανεξάρτητα της κατηγορίας που ανήκει.

CreateIDList: Κατασκευάζει την λίστα με τα μοναδικά ID που έχουν τα σημεία παρεμβολής των καμπυλών.

CreatePointdList: Κατασκευάζει μια λίστα που περιέχει σημεία του Rhinoceros και τα οποία έχουν δημιουργηθεί από αρχείο .txt που έχει εισάγει ο χρήστης.

3.3.3 *ExtendedCurve*

Όπως και οι παραπάνω δύο κλάσεις, η κλάση ExtendedCurve ανήκει στο Model και αποτελεί μια επέκταση της ήδη υπάρχουσας κλάσης Curve του Rhinoceros. Περιέχει την εξής ιδιότητα:

_NurbsCurve: Μια μη ομοιόμορφη ρητή καπύλη B-Spline (Non Uniform Rational B-Spline)

και την εξής μέθοδο:

InterpolateExtendedPoints: Η μέθοδος αυτή χρησιμοποιεί την μέθοδο παρεμβολής σημείων που υπάρχει στις βιβλιοθήκες του Rhinoceros και την επεκτείνει έτσι ώστε να λαμβάνονται επιπλέον ιδιότητες των σημείων, κατά τη δημιουργία των καμπυλών όπως για παράδειγμα η συνέχεια.

3.3.4 *ExtendedPoint*

Αντίστοιχα με την πάνω κλάση, αυτή η κλάση αποτελεί μια επέκταση της κλάσης Point που υπάρχει στις βιβλιοθήκες του Rhinoceros προκειμένου να προστεθεί περισσότερη πληροφορία σε κάθε σημείο. Περιέχει τις παρακάτω ιδιότητες:

BreakCurvePoint: Καθορίζει σημεία στα οποία θα διακόπτεται η καμπύλη και δεν θα γίνεται παρεμβολή των αντίστοιχων σημείων.

Index: Αρίθμηση του σημείου.

KnuckleAngle: Σε σημεία ασυνέχειας της καμπύλης η τιμή αυτή είναι γωνία που έχει η πρώτη καμπύλη στο σημείο αυτό.

KnuckleAngleUpper: Εάν το σημείο είναι σημείο ασυνέχειας η ιδιότητα αυτή αποτελεί την τιμή της γωνίας της δεύτερης καμπύλης που συνδέεται στο σημείο αυτό.

Location: Ένα αντικείμενο τύπου Point3d από τη βιβλιοθήκη του Rhinoceros, το οποίο περιλαμβάνει τις συντεταγμένες του σημείου.

PointCurve: Ο τύπος της καμπύλης που ανήκει το σημείο.

PointType: Μια επιπλέον ιδιότητα του σημείου που καθορίζει και την συνέχεια της καμπύλης. Υπάρχουν τρεις διαφορετικοί τύποι Ordinary, Tangent και Knuckle.

TangentAngle: Εάν το σημείο είναι τύπου Tangent, η συγκεκριμένη ιδιότητα είναι η τιμή της εφαπτομένης στο κοινό σημείο των δύο καμπυλών.

3.3.5 CurveCreation

Η κλάση αυτή ανήκει στο Model και περιλαμβάνει μόνο μεθόδους για την κατασκευή καμπυλών, από σημεία τομής ενός επιπέδου με άλλες καμπύλες. Οι μέθοδοι της κλάσης αυτής είναι οι εξής:

GetButtockIntersectionPoints: Η κλάση αυτή βρίσκει τις τομές επιπέδου με κάθετο διάνυσμα παράλληλο με τον άξονα y με τις ομάδες ναυπηγικών καμπυλών που έχει ορίσει ο χρήστης.

GetSectionIntersectionPoints: Η κλάση αυτή βρίσκει τις τομές επιπέδου με κάθετο διάνυσμα παράλληλο με τον άξονα x με τις ομάδες ναυπηγικών καμπυλών που έχει ορίσει ο χρήστης.

GetWaterlineIntersectionpoints: Η κλάση αυτή βρίσκει τις τομές επιπέδου με κάθετο διάνυσμα παράλληλο με τον άξονα z με τις ομάδες ναυπηγικών καμπυλών που έχει ορίσει ο χρήστης.

GetButtockPoints: Η μέθοδος αυτή εκτελείται αμέσως μετά την εκτέλεση της GetButtockIntersectionPoints, με σκοπό να εισάγει όλες τις απαραίτητες ιδιότητες στα σημεία, να βάλει τα σημεία στην κατάλληλη σειρά ώστε να δημιουργηθεί η επιθυμητή καμπύλη και να αφαιρέσει τυχόν διπλά σημεία.

GetSectionPoints: Η μέθοδος αυτή εκτελείται αμέσως μετά την εκτέλεση της GetSectionIntersectionPoints, με σκοπό να εισάγει όλες τις απαραίτητες ιδιότητες στα σημεία, να βάλει τα σημεία στην κατάλληλη σειρά ώστε να δημιουργηθεί η επιθυμητή καμπύλη και να αφαιρέσει τυχόν διπλά σημεία.

GetWaterlinePoints: Η μέθοδος αυτή εκτελείται αμέσως μετά την εκτέλεση της GetWaterlineIntersectionPoints, με σκοπό να εισάγει όλες τις απαραίτητες ιδιότητες στα σημεία, να βάλει τα σημεία στην κατάλληλη σειρά ώστε να δημιουργηθεί η επιθυμητή καμπύλη και να αφαιρέσει τυχόν διπλά σημεία.

3.3.6 *FitFOBFOSPMB*

Η μέθοδος αυτή ανήκει στο Model και λειτουργεί βοηθητικά με δύο μεθόδους, όταν ο χρήστης εισάγει το FOS (Flat of Side) και το FOB (Flat of Bottom) αντίστοιχα. Οι μέθοδοι είναι οι παρακάτω:

FitFOBtoPMB: Η εκτέλεση αυτής της μεθόδου γίνεται αφού ο χρήστης εισάγει το FOB από αρχείο και έχει ως σκοπό να μεταφέρει την καμπύλη ώστε να περνάει από τα σωστά σημεία του παράλληλου τμήματος και με τις σωστές κλίσεις.

FitFOStoPMB: Η εκτέλεση αυτής της μεθόδου γίνεται αφού ο χρήστης εισάγει το FOS από αρχείο και έχει ως σκοπό να μεταφέρει την καμπύλη ώστε να περνάει από τα σωστά σημεία του παράλληλου τμήματος και με τις σωστές κλίσεις.

3.3.7 *ViewModelBase*

Αυτή η κλάση ανήκει στο ViewModel και λειτουργεί ως κλάση που κληρονομούν όλες κλάσεις του ViewModel προκειμένου να αποκτήσουν την δυνατότητα να στέλνουν Notifications όταν μια ιδιότητα της κλάσης αλλάζει. Περιέχει μια μέθοδο RaisePropertyChanged καθώς και ένα Event το PropertyChanged.

3.3.8 *ViewModelExtendedPoint*

Η συγκεκριμένη κλάση χρησιμοποιείται για τη σύζευξη της κλάσης των σημείων του Model με το View. Ανήκει στο ViewModel και περιλαμβάνει κυρίως ιδιότητες που θα παρουσιάζονται στο παράθυρο και ο χρήστης θα μπορεί να αλλάξει. Στη συνέχεια μέσω κατάλληλων μεθόδων οι αλλαγές αυτές να πραγματοποιούνται και στα αντίστοιχα αντικείμενα του Model. Οι ιδιότητες της συγκεκριμένης κλάσης είναι οι εξής:

FirstAngle: Η γωνία της πρώτης καμπύλης στο σημείο αυτό σε περίπτωση που το σημείο δεν είναι Ordinary.

FirstAngleEnabled: Καθορίζει αν θα μπορεί να μεταβληθεί από το χρήστη η παραπάνω ιδιότητα.

Index: Αρίθμηση του σημείου.

PointType: Ο τύπος του σημείου Ordinary, Tangent ή Knuckle.

SecondAngle: Η γωνία της δεύτερης καμπύλης στο σημείο αυτό σε περίπτωση που το σημείο δεν είναι Ordinary.

SecondAngleEnabled: Καθορίζει αν θα μπορεί να μεταβληθεί από τον χρήστη η παραπάνω ιδιότητα.

X: Η x συντεταγμένη του σημείου.

XEnabled: Καθορίζει αν θα μπορεί να μεταβληθεί από τον χρήστη η x συντεταγμένη του σημείου αυτού μέσω του παραθύρου του Plugin.

Y: Η y συντεταγμένη του σημείου.

YEnabled: Καθορίζει αν θα μπορεί να μεταβληθεί από τον χρήστη η y συντεταγμένη του σημείου αυτού μέσω του παραθύρου του Plugin.

Z: Η z συντεταγμένη του σημείου.

ZEnabled: Καθορίζει αν θα μπορεί να μεταβληθεί από τον χρήστη η z συντεταγμένη του σημείου αυτού μέσω του παραθύρου του Plugin.

Η μοναδική μέθοδος που περιέχει η κλάση είναι η εξής:

CreateViewModelPoint: Κατασκευάζει ένα αντικείμενο τύπου ViewModelExtendedPoint.

3.3.9 ViewModelGenericCurve

Η κλάση ViewModelGenericCurve ανήκει στο ViewModel και περιέχει την πληροφορία της αντίστοιχης κλάσης του Model που θα θέλαμε να προβάλουμε στο παράθυρο του Plugin. Περιέχει μόνο ιδιότητες, οι οποίες φαίνονται παρακάτω:

CurveName: Το όνομα με το οποίο φαίνεται στο παράθυρο η καμπύλη.

CurvePosition: Η θέση της καμπύλης.

CurveType: Ο τύπος της καμπύλης (Section, Waterline, Buttock ή Boundary)

Index: Χρησιμοποιείται για την αρίθμηση.

IsEditable: Καθορίζει αν η ναυπηγική καμπύλη μπορεί να επεξεργαστεί.

IsSelected: Καθορίζει αν η ναυπηγική καμπύλη έχει επιλεγθεί.

IsVisible: Καθορίζει αν η ναυπηγική καμπύλη θα φαίνεται στα Viewports του Rhinoceros.

3.3.10 ViewModelMain

Η κλάση ViewModelMain είναι πολύ σημαντική και απτελεί ουσιαστικά τον πυρήνα του ViewModel ολόκληρου, διαχειρίζεται την έκθεση των πληροφοριών του Model στο View και την μεταφορά αλλαγών από το View στο Model και αντίστροφα. Η κλάση έχει τις εξής ιδιότητες:

DeleteBoundaries: Η εντολή (ICommand) που χρησιμοποιείται για να διαγράψουμε ναυπηγικές καμπύλες τύπου Boundary.

DeleteButtocks: Η εντολή (ICommand) που χρησιμοποιείται για να διαγράψουμε ναυπηγικές καμπύλες τύπου Buttock.

DeleteSections: Η εντολή (ICommand) που χρησιμοποιείται για να διαγράψουμε ναυπηγικές καμπύλες τύπου Section.

DeleteWaterlines: Η εντολή (ICommand) που χρησιμοποιείται για να διαγράψουμε ναυπηγικές καμπύλες τύπου Waterline.

EditBoundaries: Η εντολή (ICommand) που χρησιμοποιείται για να επεξεργαστούμε ναυπηγικές καμπύλες τύπου Boundary.

EditButtocks: Η εντολή (ICommand) που χρησιμοποιείται για να επεξεργαστούμε ναυπηγικές καμπύλες τύπου Buttock.

EditSections: Η εντολή (ICommand) που χρησιμοποιείται για να επεξεργαστούμε ναυπηγικές καμπύλες τύπου Section.

EditWaterlines: Η εντολή (ICommand) που χρησιμοποιείται για να επεξεργαστούμε ναυπηγικές καμπύλες τύπου Waterline.

ExportLinesCommand: Η εντολή (ICommand) που χρησιμοποιείται για την εξαγωγή του τελικού αρχείου .3dm.

HideBoundaries: Η εντολή (ICommand) που χρησιμοποιείται για να κρύψει ναυπηγικές καμπύλες τύπου Boundary από τα Viewport του Rhinoceros.

HideButtocks: Η εντολή (ICommand) που χρησιμοποιείται για να κρύψει ναυπηγικές καμπύλες τύπου Buttock από τα Viewport του Rhinoceros.

HideSections: Η εντολή (ICommand) που χρησιμοποιείται για να κρύψει ναυπηγικές καμπύλες τύπου Section από τα Viewport του Rhinoceros.

HideWaterlines: Η εντολή (ICommand) που χρησιμοποιείται για να κρύψει ναυπηγικές καμπύλες τύπου Waterline από τα Viewport του Rhinoceros.

LoadClick: Η εντολή (ICommand) που χρησιμοποιείται για να ανοίξουμε ένα αρχείο .xml προκειμένου να το διαβάσει το Plugin. Όπως θα αναφερθεί και παρακάτω η αποθήκευση της σχεδίασης που έχει γίνει χρησιμοποιώντας το Shiplines γίνεται με αρχεία τύπου xml.

ObservableCollectionBoundaries: Λίστα με την πληροφορία των ναυπηγικών καμπυλών Boundaries που θέλουμε να εκθέσουμε στο View.

ObservableCollectionButtocks: Λίστα με την πληροφορία των ναυπηγικών καμπυλών Buttocks που θέλουμε να εκθέσουμε στο View.

ObservableCollectionSections: Λίστα με την πληροφορία των ναυπηγικών καμπυλών Sections που θέλουμε να εκθέσουμε στο View.

ObservableCollectionWaterlines: Λίστα με την πληροφορία των ναυπηγικών καμπυλών Waterlines που θέλουμε να εκθέσουμε στο View.

SaveClick: Η εντολή (ICommand) που χρησιμοποιείται για την αποθήκευση ενός αρχείου .xml με όλη την απαραίτητη πληροφορία που έχει δημιουργηθεί.

ShowBoundaries: Η εντολή (ICommand) που χρησιμοποιείται για να εμφανίσει ναυπηγικές καμπύλες τύπου Boundary από τα Viewport του Rhinoceros, εάν τις είχαμε κρύψει.

ShowButtocks: Η εντολή (ICommand) που χρησιμοποιείται για να εμφανίσει ναυπηγικές καμπύλες τύπου Buttock από τα Viewport του Rhinoceros, εάν τις είχαμε κρύψει.

ShowSections: Η εντολή (ICommand) που χρησιμοποιείται για να εμφανίσει ναυπηγικές καμπύλες τύπου Section από τα Viewport του Rhinoceros, εάν τις είχαμε κρύψει.

ShowWaterlines: Η εντολή (ICommand) που χρησιμοποιείται για να εμφανίσει ναυπηγικές καμπύλες τύπου Waterline από τα Viewport του Rhinoceros, εάν τις είχαμε κρύψει.

Tabs: Λίστα που περιέχει τις κλάσεις με την πληροφορία για τα Tabs του παραθύρου.

Η κλάση ViewModelMain περιέχει επίσης και αρκετές μεθόδους, τις οποίες θα αναλύσουμε παρακάτω:

CanEditClick: Είναι μια μέθοδος που καθορίζει αν θα μπορούμε να επεξεργαστούμε κάποια καμπύλη ή όχι.

CanHideClick: Αντίστοιχα με την παραπάνω μέθοδο, η μέθοδος αυτή καθορίζει αν θα μπορούμε να κρύψουμε μια καμπύλη ή όχι.

CanShowClick: Η μέθοδος αυτή καθορίζει αν θα μπορούμε να εμφανίσουμε μια καμπύλη που είχαμε κρύψει.

DeleteClick: Εκτελείται όταν θέλουμε να διαγράψουμε μια ναυπηγική καμπύλη.

EditClick: Εκτελείται όταν θέλουμε να επεξεργαστούμε μια ναυπηγική καμπύλη.

HideClick: Εκτελείται όταν θέλουμε να κρύψουμε μια ναυπηγική καμπύλη.

ObservableCollectionBoundaries_CollectionChanged: Εκτελείται όταν αλλάζει η λίστα του ViewModel για τις καμπύλες Boundary, προκειμένου να ενημερώσει την αντίστοιχη λίστα στο Model.

ObservableCollectionButtocks_CollectionChanged: Εκτελείται όταν αλλάζει η λίστα του ViewModel για τις καμπύλες Buttock, προκειμένου να ενημερώσει την αντίστοιχη λίστα στο Model.

ObservableCollectionSections_CollectionChanged: Εκτελείται όταν αλλάζει η λίστα του ViewModel για τις καμπύλες Section, προκειμένου να ενημερώσει την αντίστοιχη λίστα στο Model.

ObservableCollectionWaterlines_CollectionChanged: Εκτελείται όταν αλλάζει η λίστα του ViewModel για τις καμπύλες Waterline, προκειμένου να ενημερώσει την αντίστοιχη λίστα στο Model.

OnIsSelectedBoundaryPropertyChanged: Εκτελείται κάθε φορά που ο αριθμός των επιλεγμένων καμπυλών τύπου Boundary αλλάζει.

OnIsSelectedButtockPropertyChanged: Εκτελείται κάθε φορά που ο αριθμός των επιλεγμένων καμπυλών τύπου Buttock αλλάζει.

OnIsSelectedSectionPropertyChanged: Εκτελείται κάθε φορά που ο αριθμός των επιλεγμένων καμπυλών τύπου Section αλλάζει.

OnIsSelectedWaterlinePropertyChanged: Εκτελείται κάθε φορά που ο αριθμός των επιλεγμένων καμπυλών τύπου Waterline αλλάζει.

OnIsVisiblePropertyChanged: Εκτελείται κάθε φορά που εμφανίζουμε ή κρύβουμε τις επιλεγμένες ναυπηγικές καμπύλες.

ShowClick: Εκτελείται όταν θέλουμε εμφανίσουμε μια ναυπηγική καμπύλη, που είχαμε κρύψει.

3.3.11 ViewModelCreateTab

Η συγκεκριμένη κλάση περιέχει τις ιδιότητες που χρησιμοποιούνται από τα εργαλεία του Create Tab του παραθύρου, παρακάτω αναφέρουμε τις ιδιότητες:

AddPMBClick: Η εντολή (ICommand) που κατασκευάζει το παράλληλο τμήμα του πλοίου.

CreateBoundaries: Καθορίζει εάν ο χρήστης έχει επιλέξει να δημιουργήσει κάποια καμπύλη τύπου Boundary.

CreateButtocks: Καθορίζει εάν ο χρήστης έχει επιλέξει να δημιουργήσει κάποια καμπύλη τύπου Buttock.

CreateCurvesCommand: Η εντολή (ICommand) για τη δημιουργία ναυπηγικών καμπυλών χρησιμοποιώντας τα σημεία, που προέκυψαν από τις τομές άλλων ναυπηγικών καμπυλών με κάποιο επίπεδο.

CreateSections: Καθορίζει εάν ο χρήστης έχει επιλέξει να δημιουργήσει κάποια καμπύλη τύπου Section.

CreateShaftCurve: Η εντολή (ICommand) για τη δημιουργία του ημικύκλιου της χοάνης της έλικας.

CreateWaterlines: Καθορίζει εάν ο χρήστης έχει επιλέξει να δημιουργήσει κάποια καμπύλη τύπου Waterline.

OpenFileCommand: Η εντολή (ICommand) για την εισαγωγή αρχείου .txt κατά τη δημιουργία της λίστας των σημείων μιας ναυπηγικής καμπύλης.

3.3.12 ViewModelEditTab

Η συγκεκριμένη κλάση περιέχει τις ιδιότητες που χρησιμοποιούνται από τα εργαλεία του Edit Tab του παραθύρου, παρακάτω αναφέρουμε τις ιδιότητες και τις μεθόδους:

FinishEditing: Η εντολή (ICommand) που εκτελείται όταν τελειώνουμε την επεξεργασία μιας ναυπηγικής καμπύλης.

FitButtocks: Καθορίζει αν οι ναυπηγικές καμπύλες τύπου Buttock θα ξαναδημιουργηθούν, έτσι ώστε τα καινούργια σημεία να είναι τομές επιπέδου κάθετο στον y, με τις επιλεγμένες καμπύλες.

FitCurves: Η εντολή (ICommand) για να δημιουργηθούν ξανά οι ναυπηγικές καμπύλες, ανάλογα με τον τύπο που έχει επιλέξει ο χρήστης.

FitSections: Καθορίζει αν οι ναυπηγικές καμπύλες τύπου Buttock θα ξαναδημιουργηθούν, έτσι ώστε τα καινούργια σημεία να είναι τομές επιπέδου κάθετο στον x, με τις επιλεγμένες καμπύλες.

FitWaterlines: Καθορίζει αν οι ναυπηγικές καμπύλες τύπου Buttock θα ξαναδημιουργηθούν, έτσι ώστε τα καινούργια σημεία να είναι τομές επιπέδου κάθετο στον z, με τις επιλεγμένες καμπύλες.

In interpolateButtocks: Καθορίζει αν για τη δημιουργία ναυπηγικών καμπυλών θα χρησιμοποιηθούν τομές των καμπυλών Buttock με κάποιο επίπεδο.

InterpolateSections: Καθορίζει αν για τη δημιουργία ναυπηγικών καμπυλών θα χρησιμοποιηθούν τομές των καμπυλών Section με κάποιο επίπεδο.

InterpolateWaterlines: Καθορίζει αν για τη δημιουργία ναυπηγικών καμπυλών θα χρησιμοποιηθούν τομές των καμπυλών Waterline με κάποιο επίπεδο.

KnuckleAngle: Η γωνία της πρώτης καμπύλης σε ένα σημείο Knuckle.

KnuckleAngleUpper: Η γωνία της δεύτερης καμπύλης σε ένα σημείο Knuckle.

PointsDataGrid: Η λίστα με τα σημεία που προβάλλεται στο παράθυρο όταν επεξεργαζόμαστε κάποια ναυπηγική καμπύλη.

SetKnuckle: Καθορίζει εάν ο χρήστης θέλει να αλλάξει ένα απλό σημείο ή ένα σημείο Tangent και να το μετατρέψει σε Knuckle.

SetKnuckleAngles: Καθορίζει εάν ο χρήστης θέλει να ορίσει και τις γωνίες σε ένα σημείο Knuckle.

SetOrdinary: Καθορίζει εάν ο χρήστης θέλει να αλλάξει ένα σημείο Knuckle ή ένα σημείο Tangent και να το μετατρέψει σε Ordinary.

SetPointTypeCommand: Η εντολή (ICommand) για να αλλάξουμε τον τύπο του σημείου.

SetTangent: Καθορίζει εάν ο χρήστης θέλει να αλλάξει ένα σημείο Knuckle ή ένα σημείο Ordinary και να το μετατρέψει σε Tangent.

TangentAngle: Η γωνία των καμπυλών σε ένα σημείο Knuckle.

Οι μέθοδοι της κλάσης είναι οι εξής:

CanDoneClick: Η μέθοδος αυτή καθορίζει αν θα μπορούμε να τελειώσουμε την επεξεργασία κάποιας καμπύλης.

CanSetKnuckleAngle: Η μέθοδος αυτή καθορίζει αν μπορούμε να ορίσουμε τις δύο γωνίες σε ένα σημείο τύπου Knuckle.

CanSetPointType: Η μέθοδος αυτή καθορίζει αν μπορούμε να ορίσουμε τον τύπο ενός σημείου.

DoneClick: Η εντολή (ICommand) τερματίζει τη διαδικασία επεξεργασίας μιας καμπύλης, κρύβοντας τα σημεία παρεμβολής και αφαιρώντας οποιαδήποτε δυνατότητα επεξεργασίας της καμπύλης.

PointsDataGrid_CollectionChanged: Εκτελείται όταν ο χρήστης αλλάζει κάποια ιδιότητα ενός σημείου μέσω του παραθύρου (UI) του Plugin.

3.3.13 ViewModelGeneralTab

Η συγκεκριμένη κλάση περιέχει τις ιδιότητες που χρησιμοποιούνται από τα εργαλεία του General Tab του παραθύρου, παρακάτω αναφέρουμε τις ιδιότητες αυτές:

Beam: Το πλάτος του πλοίου.

Beaminput: Η εισαγωγή του πλάτους του πλοίου ως δεδομένο από τον χρήστη.

ChangeShipDimensions: Η εντολή (ICommand) για την αλλαγή των διαστάσεων του πλοίου.

Depth: Το κοίλο του πλοίου.

Depthinput: Η εισαγωγή του κοίλου του πλοίου ως δεδομένο από τον χρήστη.

LBP: Το μήκος μεταξύ καθέτων του πλοίου.

LBPinput: Η εισαγωγή του μήκους μεταξύ καθέτων του πλοίου ως δεδομένο από τον χρήστη.

PMBAftLocation: Τιμή της θέσης του παράλληλου τμήματος του υπο σχεδίαση πλοίου πρύμα της μέσης τομής.

PMBForLocation: Τιμή της θέσης του παράλληλου τμήματος του υπο σχεδίαση πλοίου πλώρα της μέσης τομής.

3.3.14 ViewModelOptionsTab

Η συγκεκριμένη κλάση περιέχει τις ιδιότητες που χρησιμοποιούνται από τα εργαλεία του Options Tab του παραθύρου, παρακάτω αναφέρουμε τις ιδιότητες αυτές:

ApplyColorChanges: Η εντολή (ICommand) για την αλλαγή του χρώματος με το οποίο θα φαίνονται οι καμπύλες.

ColorsList: Λίστα με όλα τα χρώματα με τα οποία μπορούμε να εμφανίσουμε τις καμπύλες.

3.3.15 CurveConduit

Η κλάση αυτή κληρονομεί από τη βιβλιοθήκη του Rhinoceros την κλάση DisplayConduit και ανήκει στην κατηγορία RhinoDisplay, αλλά θα μπορούσε να τοποθετηθεί και στο Model. Είναι πολύ καθοριστική για τη λειτουργία του προγράμματος, καθώς καθορίζει τον τρόπο με τον οποίο εμφανίζονται οι καμπύλες στα Viewports του Rhinoceros και κατ'επέκταση τον τρόπο με τον οποίο αλληλεπιδρά ο χρήστης με αυτές. Οι ιδιότητες και οι μέθοδοι που έχει η κλάση αυτή είναι οι παρακάτω:

Color: Το χρώμα με το οποίο θα προβάλεται η καμπύλη στα Viewports.

και η μέθοδος:

DrawOverlay: Εμφανίζει την καμπύλη στα Viewports με το χρώμα που έχει επιλέξει ο χρήστης.

3.3.16 CurvatureGraph

Η κλάση CurvatureGraph ανήκει και αυτή στην κατηγορία RhinoDisplay, και σχετίζεται με την εμφάνιση της καμπυλότητας. Λειτουργεί με τρόπο όμοιο με αυτόν που εμφανίζονται οι καμπύλες στα Viewports και γι'αυτό έχουν τις ίδιες ιδιότητες και μεθόδους. Οι ιδιότητα είναι:

Color: Το χρώμα με το οποίο θα εμφανίζονται τα διανύσματα της καμπυλότητας.

η μέθοδος είναι:

DrawOverlay: Εμφανίζει τα διανύσματα καμπυλότητας της καμπύλης που επεξεργαζόμαστε.

3.3.17 ClearCollection

Η κλάση ClearCollection ανήκει στην κατηγορία με τις βοηθητικές κλάσεις Helpers και περιέχει μόνο μια μέθοδο:

CollectionClear: Διαγράφει το περιεχόμενο μιας λίστας με ναυπηγικές καμπύλες.

3.3.18 ProjectStorage

Η κλάση ProjectStorage χρησιμοποιείται για την αποθήκευση και φόρτωση των αρχείων xml που χρησιμοποιεί το Plugin. Οι μέθοδοι της κλάσης αυτή είναι οι εξής:

LoadProject: Χρησιμοποιείται για να φορτώσουμε το αρχείο που αποθηκεύσαμε. Το αρχείο αυτό επισημαίνεται ότι, διαβάζεται μόνο από το Plugin και όχι από το Rhinoceros.

SaveCurve: Καθορίζει ποιες πληροφορίες μιας ναυπηγικής καμπύλης πρέπει να αποθηκευθούν.

SaveProject: Αποθηκεύει όλα τα δεδομένα που χρειάζονται για τις ναυπηγικές γραμμές, προκειμένου να μπορούν να ξαναχρησιμοποιηθούν σε ένα αρχείο .xml.

3.3.19 ReadtxtPoints

Η κλάση αυτή λειτουργεί βοηθητικά, ανήκει στους Helpers και αποτελείται από μια μέθοδο:

readpointsfromtxt: Διαβάζει ένα αρχείο .txt με συντεταγμένες σημείων και δημιουργεί ένα πίνακα με τις συντεταγμένες αυτές για να χρησιμοποιηθεί από άλλη μέθοδο για τη δημιουργία καμπύλης.

3.3.20 *RelayCommand*

Προκειμένου να συνδέσουμε κάποιο εργαλείο (π.χ. Button) σε κάποια ιδιότητα κλάσης που ανήκει στο ViewModel χρησιμοποιούμε το interface ICommand. Η κλάση RelayCommand κληρονομεί το interface ICommand, και αποτελείται από δύο τμήματα, τα οποία είναι:

CanExecuteChanged: Εκτελείται όταν πραγματοποιούνται αλλαγές που καθορίζουν εάν η εντολή μπορεί να εκτελεστεί.

CanExecute: Καθορίζει αν η εντολή μπορεί να εκτελεστεί. Εάν η μέθοδος επιστρέψει false τότε το εργαλείο (Control) στο παράθυρο θα απενεργοποιηθεί.

Execute: Εκτελεί την επιλεγμένη μέθοδο την οποία θέλουμε να τη συνδέσουμε με το εργαλείο του παραθύρου.

3.3.21 *RemoveDuplicatesFromList*

Η κλάση αυτή ανήκει στους Helpers και περιλαμβάνει μια μέθοδο:

RemoveDuplicates: Η μέθοδος αυτή αναζητεί σε μία λίστα σημείων διπλά σημεία και σε περίπτωση που βρει κάποιο το αφαιρεί.

3.3.22 *SortObservableCollectionClass*

Η κλάση SortObservableCollectionClass ανήκει στους Helpers και έχει ως στόχο μέσω δύο μεθόδων να ελέγξει και να διορθώσει αν χρειάζεται την σειρά με την οποία τοποθετούνται στις λίστες τους τα σημεία, καθώς έχει ιδιαίτερη σημασία για την σωστή δημιουργία της καμπύλης:

SortDataGridList: Χρησιμοποιείται για την διαμόρφωση της κατάταξης των σημείων που εμφανίζονται στο DataGrid του παραθύρου.

SortObservableCollection: Χρησιμοποιείται για τον έλεγχο και διόρθωση της σειράς με την οποία έχουν τοποθετηθεί οι καμπύλες στις λίστες τους. Συνήθως τοποθετούνται με αύξουσα σειρά θέσης.

3.3.23 *SortPointsClass*

Κάθε κατηγορία καμπύλης έχει δική της μέθοδο για τη διαμόρφωση της κατάταξης των σημείων της. Η συγκεκριμένη κλάση ανήκει στους Helpers και περιλαμβάνει τρεις μεθόδους για τη διαμόρφωση της λίστας των σημείων.

SortedButtockPointsList: Διαμορφώνει την κατάταξη των σημείων σε καμπύλες τύπου Buttock. Η μέθοδος αυτή τοποθετεί τα σημεία με βάση την μικρότερη ευκλείδεια απόσταση.

SortedSectionPointsList: Διαμορφώνει την κατάταξη των σημείων σε καμπύλες τύπου Section. Η μέθοδος αυτή τοποθετεί τα σημεία με βάση την μικρότερη ευκλείδεια απόσταση.

SortedWaterlinePointsList: Διαμορφώνει την κατάταξη των σημείων σε καμπύλες τύπου Waterline. Η μέθοδος αυτή τοποθετεί τα σημεία με βάση την μικρότερη ευκλείδεια απόσταση.

3.3.24 *TangentVector*

Η κλάση TangentVector είναι η τελευταία κλάση που ανήκει στους Helpers και περιλαμβάνει μια μέθοδο:

GetTangentVector3d: Υπολογίζει διάνυσμα με μέτρο μονάδα, σε επίπεδο και με γωνία που ορίζει ο χρήστης.

3.3.25 *AppearanceWindow*

Η κλάση αυτή χρησιμοποιείται για την αλλαγή του χρώματος των καμπυλών που προβάλλονται στα Viewports του Rhinoceros μέσω της μεθόδου:

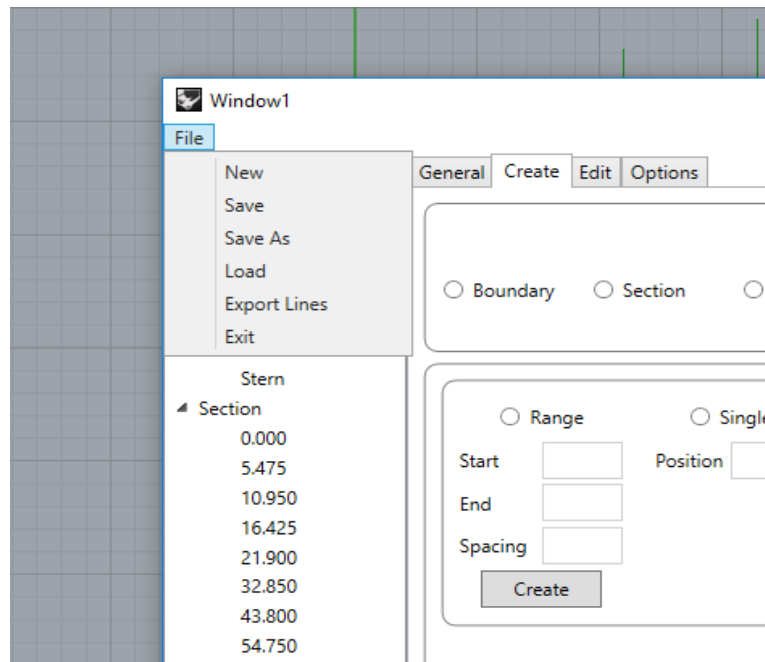
ApplyColorChanges: Αλλάζει το χρώμα που προβάλλονται οι καμπύλες.

3.4 *User Interface*

Ένα πολύ σημαντικό στοιχείο του Plugin είναι το παράθυρο (UI), καθώς είναι το μέσο με το οποίο θα αλληλεπιδρά ο χρήστης με το μοντέλο. Το framework που χρησιμοποιήθηκε είναι το WPF της Microsoft, το οποίο αξιοποιεί την XAML. Η XAML είναι μια γλώσσα βασισμένη στην XML και χρησιμοποιείται εκτενώς στο .NET Framework 3.0 και .NET Framework 4.0. Πιο συγκεκριμένα στο WPF η XAML αποτελεί το βασικό εργαλείο για να ορίσουμε τα διάφορα στοιχεία του παραθύρου και για να γίνει η σύνδεση του ViewModel με το View μέσω κατάλληλων DataBindings.

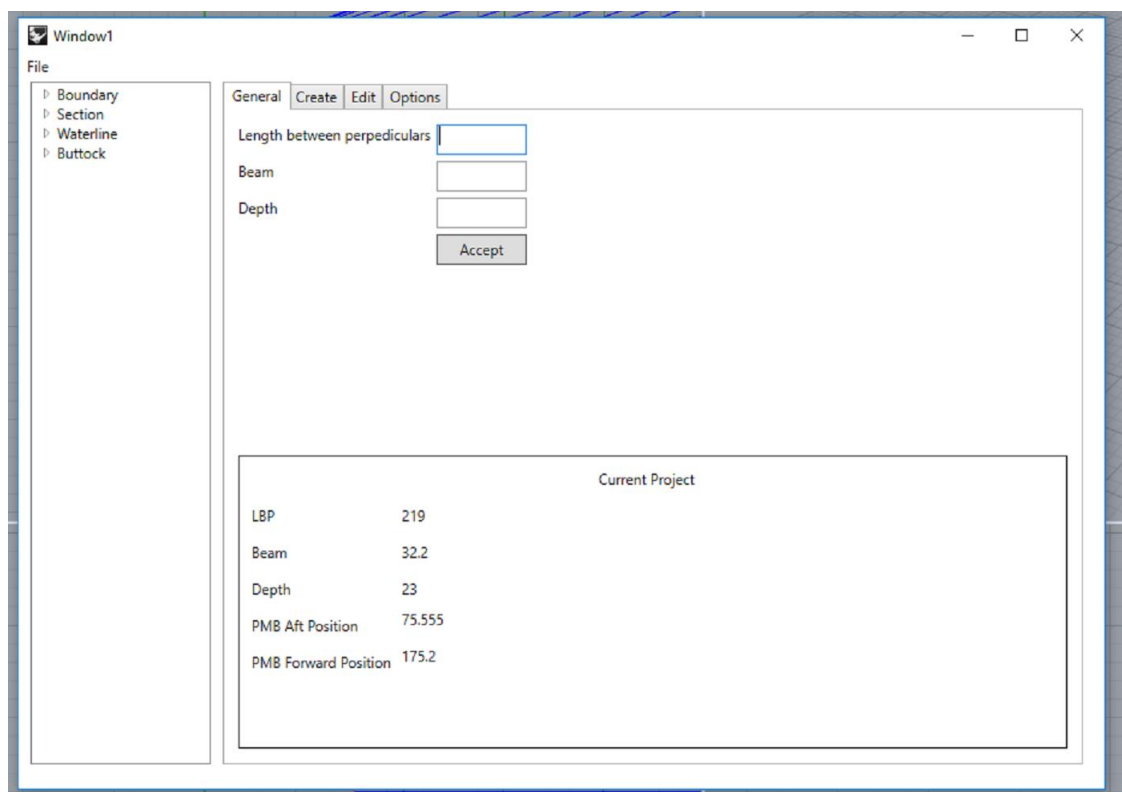
Τα βασικά στοιχεία του παραθύρου είναι ένα δέντρο (TreeView), στο οποίο εμφανίζονται Τα ονόματα των ναυπηγικών καμπυλών που έχουν δημιουργηθεί, τα Tabs με εργαλεία για την επεξεργασία και δημιουργία καμπυλών και ένα Dropdown-Menu με εργαλεία για την αποθήκευση, φόρτωση, δημιουργία και εξαγωγή αρχείων.

Τα παραπάνω στοιχεία φαίνονται στην παρακάτω εικόνα:



Σχήμα 3.4: Κύρια στοιχεία του παραθύρου

Κάθε Tab περιέχει ένα σύνολο εργαλείων που εξειδικεύονται σε συγκεκριμένες λειτουργίες. Το πρώτο Tab καλείται General και περιέχει συνοπτικά τις βασικές διαστάσεις του πλοίου, όπως τις έχει ορίσει ο χρήστης καθώς και εργαλεία για να αλλάξει ο χρήστης τις διαστάσεις αυτές.



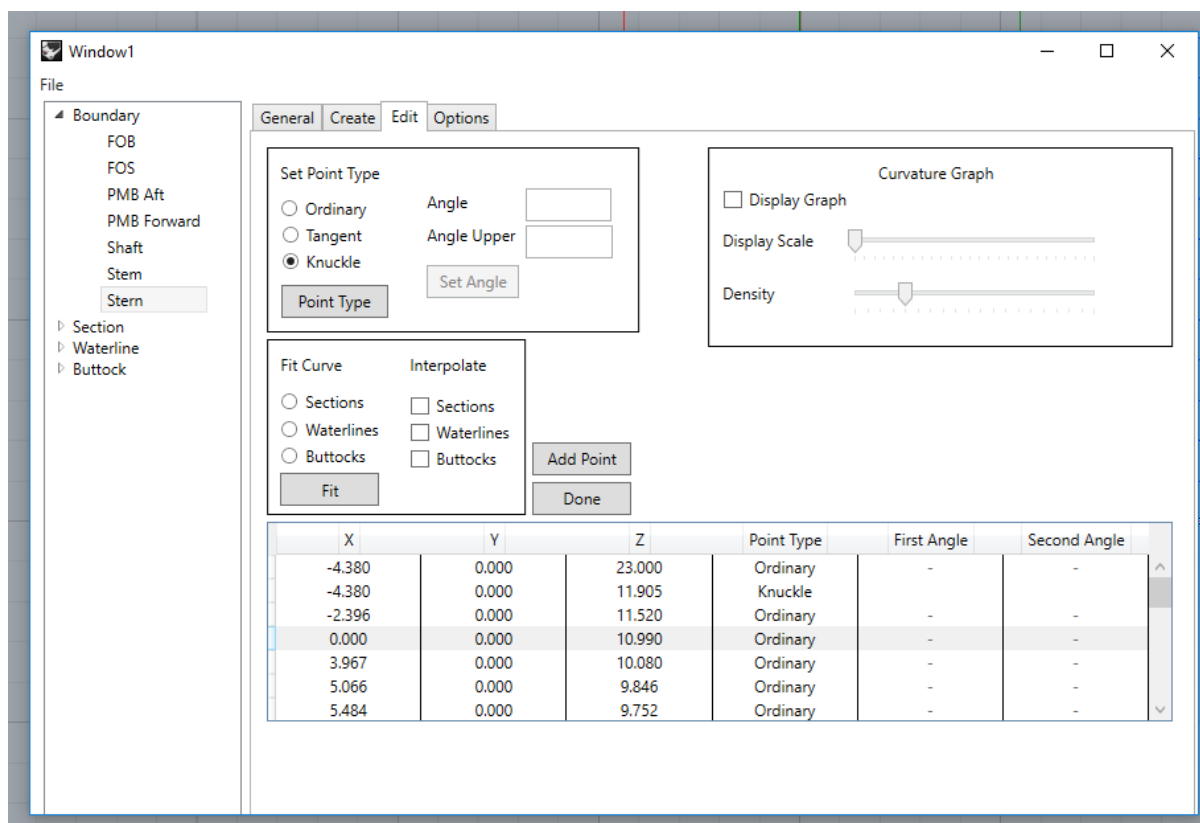
Σχήμα 3.5: Περιεχόμενο του General Tab

Το δεύτερο Tab ονομάζεται Create και περιέχει όλα τα εργαλεία που θα επιτρέψουν στο χρήστη να δημιουργήσει ναυπηγικές καμπύλες εύκολα και γρήγορα. Όπως φαίνεται και στο Σχήμα 3.6 παρακάτω υπάρχει ένα πλήθος από CheckBoxes, Buttons και TextBoxes για τη συμπλήρωση δεδομένων από τον χρήστη.

Σχήμα 3.6: Περιεχόμενο του Create Tab

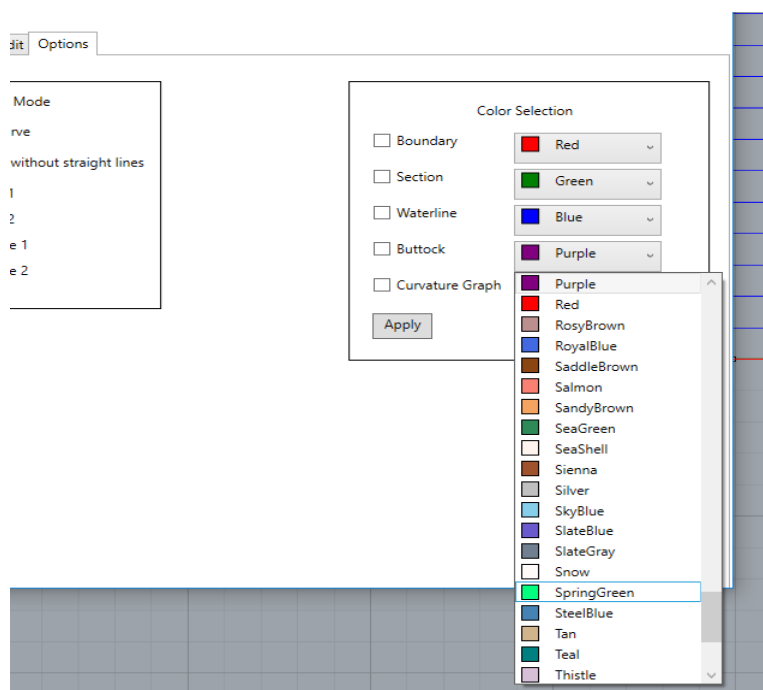
Όλα τα παραπάνω controls λειτουργούν με τέτοιο τρόπο ώστε να καθοδηγούν τον χρήστη στο επόμενο βήμα, με αποτέλεσμα η εκμάθηση να είναι εύκολη και να μειώνεται δραστικά ο αριθμός των πιθανών σφαλμάτων.

Στη συνέχεια γίνεται μια παρουσίαση ενός εξίσου σημαντικού Tab, του Edit. Το Edit Tab σχετίζεται με την επεξεργασία της καμπύλης. Περιέχει εργαλεία για την εμφάνιση και επεξεργασία των διανυσμάτων καμπυλότητας, για τον ορισμό του τύπου των σημείων, για την δημιουργία των καμπυλών ξανά, ώστε να χρησιμοποιούν κατάλληλα σημεία τομής με άλλες καμπύλες, καθώς και ένα πίνακα με όλα τα σημεία της υπο επεξεργασία καμπύλης. Ο πίνακας αυτός υλοποιείται με τη χρήση ενός DataGrid και στα κελιά του παρουσιάζονται οι συντεταγμένες των σημείων, ο τύπος τους καθώς και γωνίες που σχετίζονται με τον κάθε τύπο αν έχουν οριστεί. Στο επόμενο σχήμα (Σχήμα 3.7) φαίνονται ξεκάθαρα όλα τα προαναφερθέντα εργαλεία.



Σχήμα 3.7: Περιεχόμενο του Edit Tab

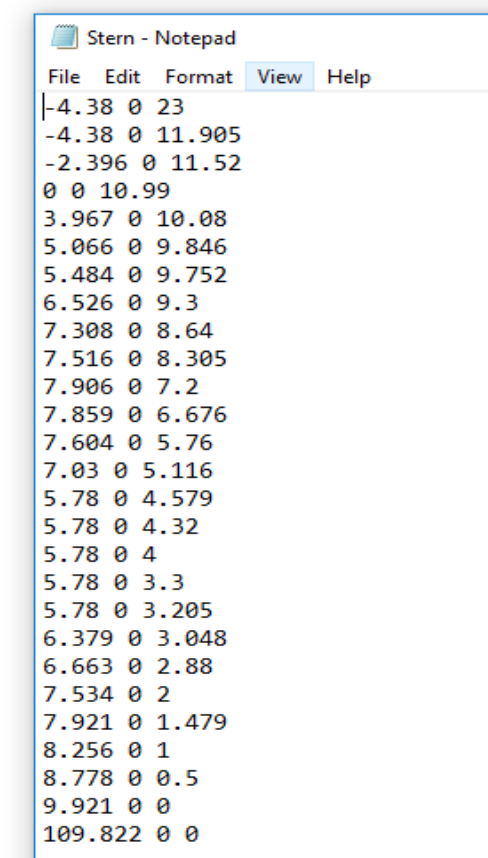
Τέλος τελευταίο Tab είναι το Options το οποίο περιέχει εργαλεία για την αλλαγή του χρώματος με το οποίο προβάλλονται οι καμπύλες στα Viewports.



Σχήμα 3.8: Περιεχόμενο του Options Tab

3.5 Αρχεία εισόδου

Για τη δημιουργία των καμπυλών είναι απαραίτητο αρχικά να δοθούν κάποια αρχεία εισόδου με τις συντεταγμένες των σημείων που θα απαρτίζουν την επιθυμητή καμπύλη. Τα αρχεία είναι μορφής TXT, σε κάθε γραμμή υπάρχουν τρία νούμερα χωρισμένα με ένα κενό που αντιστοιχούν στις x, y και z συντεταγμένες. Όπως φαίνεται στο παρακάτω παράδειγμα.



Σχήμα 3.9: Παράδειγμα αρχείου εισόδου

3.6 Αρχεία αποθήκευσης

Το Shiplines αποθηκεύει την πληροφορία που έχει δημιουργηθεί σε αρχεία μορφής XML. Κάθε Element του αρχείου αυτού θα αντιστοιχεί είτε σε μια καμπύλη είτε σε ένα σημείο και τα Attributes θα περιέχουν όλες τις ιδιότητες, όπως συντεταγμένες που θέλουμε να σώσουμε. Παρακάτω υπάρχει ένα παράδειγμα (Σχήμα 3.10) της μορφής του αρχείου.

```
<Boundaries>
  <Curve Name="PMB Aft" CurveCategory="Boundary">
    <Point X="75.555" Y="0" Z="0" PointType="Ordinary" Angle="0" BreakCurvePoint="false" />
    <Point X="75.555" Y="13.160000000000002" Z="0" PointType="Tangent" Angle="0" BreakCurvePoint="false" />
  </Curve>
</Boundaries>
```

Σχήμα 3.10: Αρχείο που δημιουργείται κατά την αποθήκευση

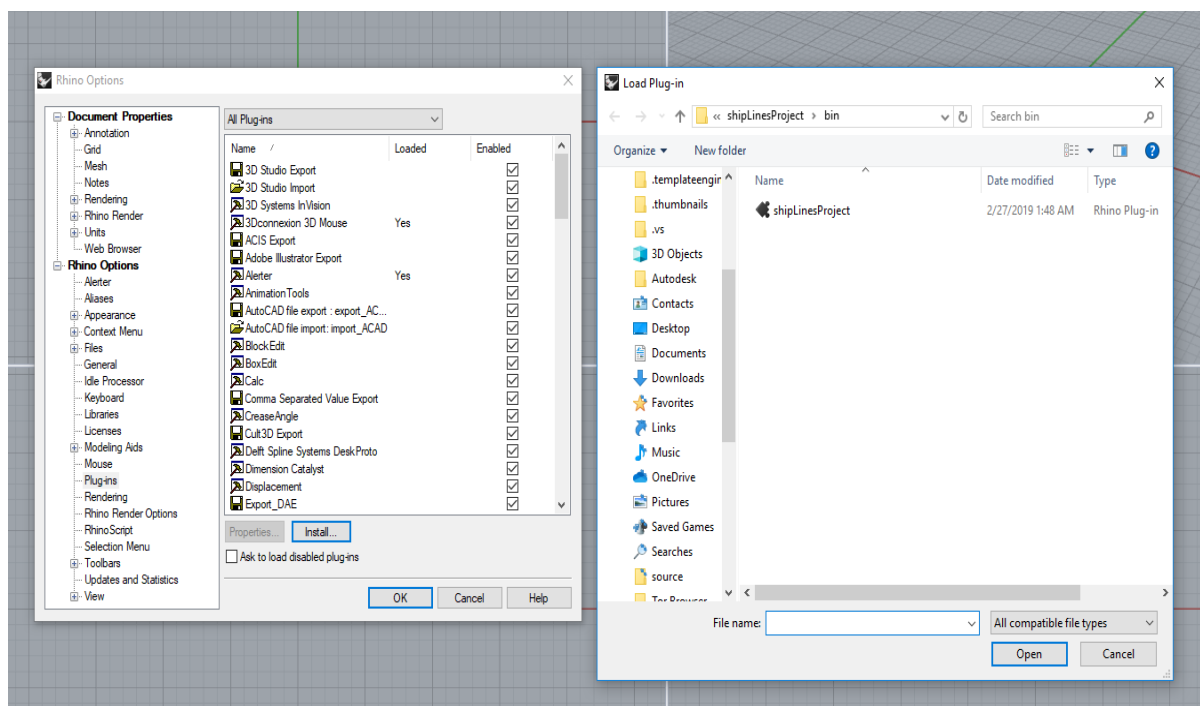
4

Υλοποίηση

4.1 Προγραμματιστικά εργαλεία και εγκατάσταση

Η υλοποίηση έγινε με C# σε περιβάλλον Windows 10 με το εργαλείο Visual Studio 2015 που περιέχει ενσωματωμένα εργαλεία τόσο για την C# όσο και για την XAML που χρησιμοποιήθηκε για την δημιουργία του παραθύρου. Το λογισμικό που απαιτούνται για την εγκατάσταση και λειτουργία της εφαρμογής είναι το Rhinoceros 5.

Σε αυτό το σημείο θα γίνει αναλυτική περιγραφή της διαδικασίας εγκατάστασης του Plugin. Αρχικά ο χρήστης θα πρέπει να πληκτρολογήσει στο Command του Rhinoceros την εντολή PlugInManager. Στη συνέχεια θα ανοίξει ένα παράθυρο που θα παρουσιάζει όλα τα Plugins που είναι εγκατεστημένα μαζί με άλλα αρχεία του προγράμματος, ο χρήστης θα επιλέξει Install και θα αναζητήσει το αρχείο τύπου Rhino Plug-in **shipLinesProject**. Αφού το επιλέξει και πατήσει Open το Plugin έχει πλέον εγκατασταθεί και μπορεί να χρησιμοποιηθεί πληκτρολογώντας στο Command την εντολή Shiplines.



Σχήμα 4.1: Περιγραφή εγκατάστασης Plugin

5

Παράδειγμα λειτουργίας

Το σημαντικότερο σκέλος της διπλωματικής εργασίας είναι η παρουσίαση ενός παραδείγματος σχεδίασης των ναυπηγικών γραμμών ενός πλοίου χρησιμοποιώντας το πρόγραμμα Shiplines.

5.1 Δεδομένα υπο σχεδίαση πλοίου

Το πλοίο του οποίου θα παράξουμε το πλέγμα των ναυπηγικών γραμμών έχει τα εξής χαρακτηριστικά:

- Μήκος μεταξύ καθέτων $L_{BP} = 187 \text{ m}$
- Πλάτος $B = 31.3 \text{ m}$
- Κοίλο $D = 17.215 \text{ m}$
- Παράλληλο τμήμα $L_{AFT} = 65.45 \text{ m}$ και $L_{FOR} = 140.25 \text{ m}$

Η πρύμνη του υπο σχεδίαση πλοίου είναι τύπου καθρέπτη (Transom) και οι πλευρές του στη μέση τομή είναι κάθετες. Για τη δημιουργία του μοντέλου χρησιμοποιήθηκαν αρχεία εισόδου για τους νομείς καθώς και για τις εξής καμπύλες Boundary:

- Stern και Stern
- FOS (Flat of Bottom)
- FOB (Flat of Side)
- Transom

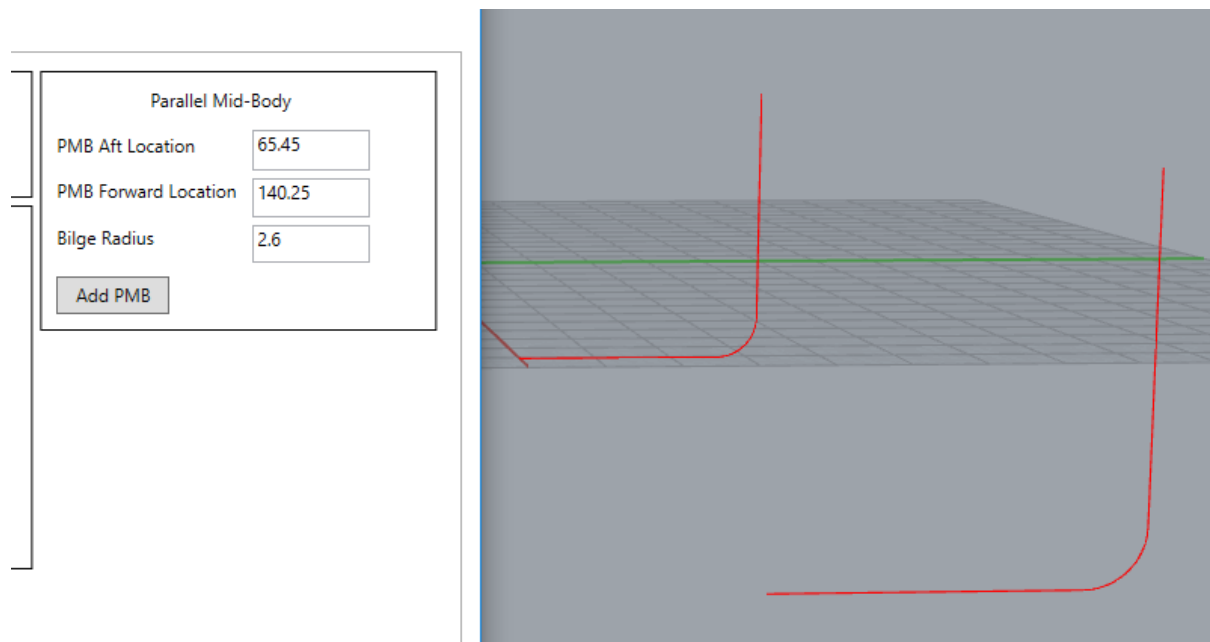
5.2 Αναλυτική διαδικασία δημιουργίας ναυπηγικών καμπυλών

Αρχικά εισάγουμε τις διαστάσεις του πλοίου στο General Tab και επιλέγουμε Accept, στο ίδιο Tab μπορούμε να επιβεβαιώσουμε ότι οι διαστάσεις του υπο σχεδίαση πλοίου είναι οι σωστές.

Current Project	
LBP	187
Beam	31.3
Depth	17.215
PMB Aft Position	
PMB Forward Position	

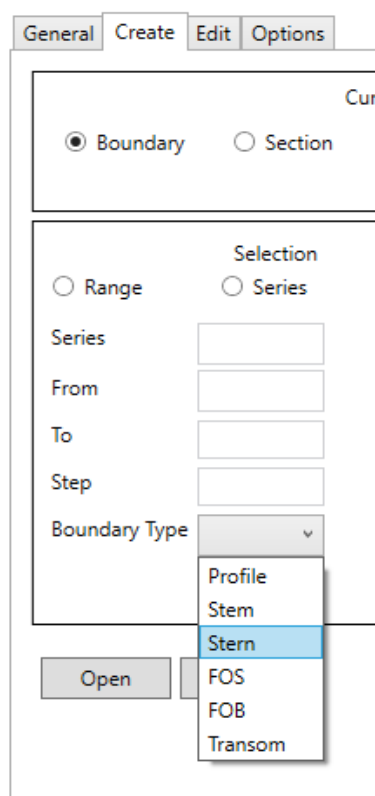
Σχήμα 5.1: Εισαγωγή κυρίων διαστάσεων του υπο σχεδίαση πλοίου

Στη συνέχεια δημιουργούμε το παράλληλο τμήμα του πλοίου από το Create Tab. Εισάγουμε τα ακραία όρια του παράλληλου τμήματος στα κατάλληλα κελιά, καθώς και την ακτίνα καμπυλότητας (Bilge Radius) και αφού εισάγουμε όλα τα δεδομένα επιλέγουμε το AddPMB. Παρατηρούμε σε αυτό το σημείο, ότι στο δέντρο εμφανίζονται τα ονόματα των καμπυλών που προσθέσαμε, ενώ στα Viewports πλέον υπάρχουν δύο όμοιες καμπύλες σε επίπεδα νομέων με τα χαρακτηριστικά που θέλουμε και στις θέσεις που οριοθετούν το παράλληλο τμήμα.



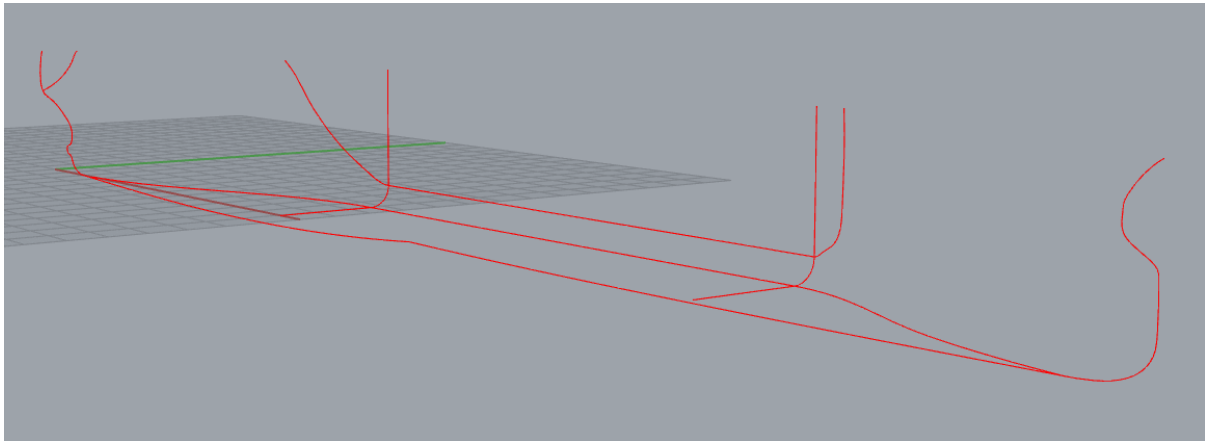
Σχήμα 5.2: Εισαγωγή του παράλληλου τμήματος

Το επόμενο βήμα είναι η εισαγωγή και των υπόλοιπων καμπυλών Boundary από τα αρχεία εισαγωγής που έχουμε κατασκευάσει. Αρχικά στο CreateTab επιλέγουμε το Boundary CheckBox, έπειτα παρατηρούμε ότι το Boundary Type ComboBox ενεργοποιείται ώστε να επιλέξουμε ποια καμπύλη από τη λίστα θέλουμε να εισάγουμε.

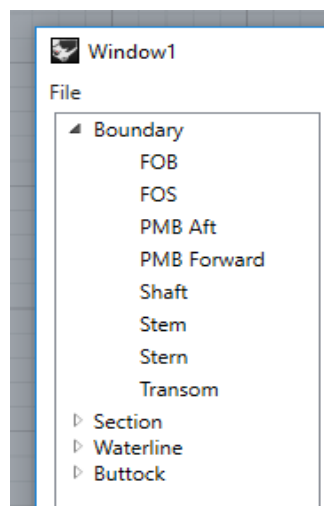


Σχήμα 5.3: Επιλογή της Boundary καμπύλης που θέλουμε να εισάγουμε

Αφού διαλέξουμε την ναυπηγική καμπύλη Boundary που θέλουμε να εισάγουμε επιλέγουμε το Button Open και ανοίγουμε το αρχείο που περιέχει τις συντεταγμένες των σημείων της καμπύλης που επιθυμούμε. Στο παρακάτω σχήμα φαίνεται το μοντέλο με όλες τις Boundary καμπύλες.



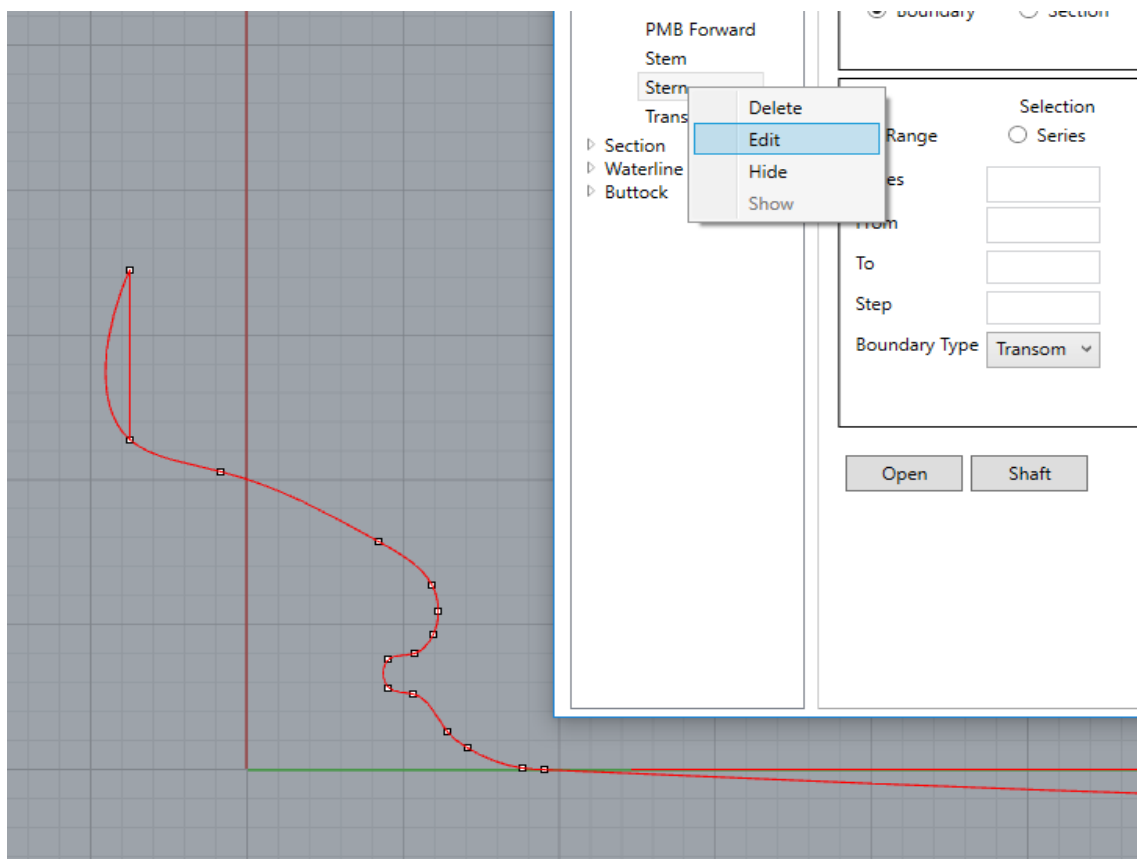
Σχήμα 5.4: Boundary καμπύλες υπο σχεδίαση πλοίου



Σχήμα 5.5: Οι καμπύλες Boundary που έχουμε εισάγει στο δέντρο

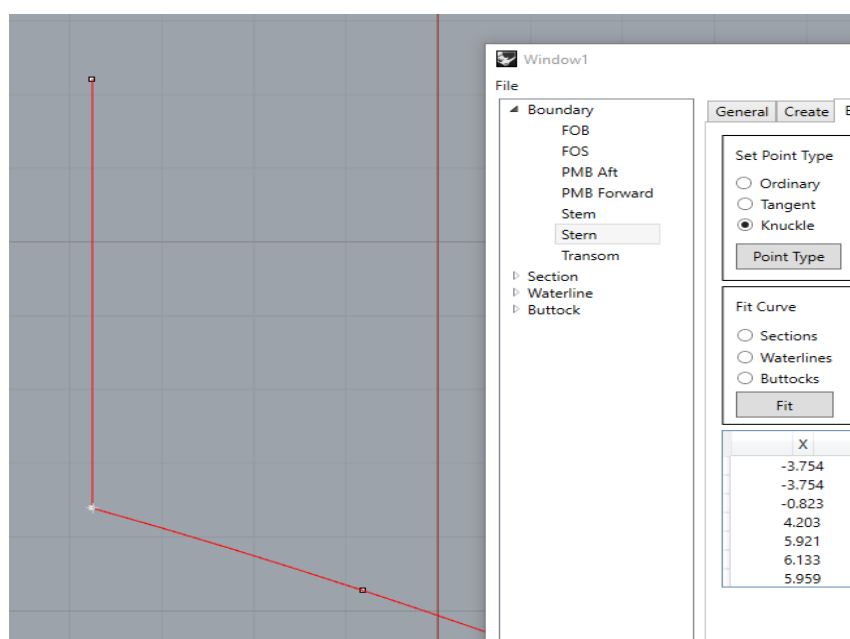
Πριν προχωρήσουμε στην εισαγωγή των νομέων απαιτείται η επεξεργασία των καμπυλών Boundary, έτσι ώστε να καθορισθεί ο τύπος όλων των σημείων και να διασφαλίσουμε ομαλή μεταβολή της καμπυλότητας.

Για να επεξεργαστούμε μια καμπύλη, την επιλέγουμε στο δέντρο και στη συνέχεια επιλέγουμε την εντολή Edit. Ανάλογα με τον τύπο της καμπύλης το πρόγραμμα μας κατευθύνει στο κατάλληλο Viewport του Rhinoceros και μας εμφανίζει όλα τα σημεία παρεμβολής της καμπύλης.



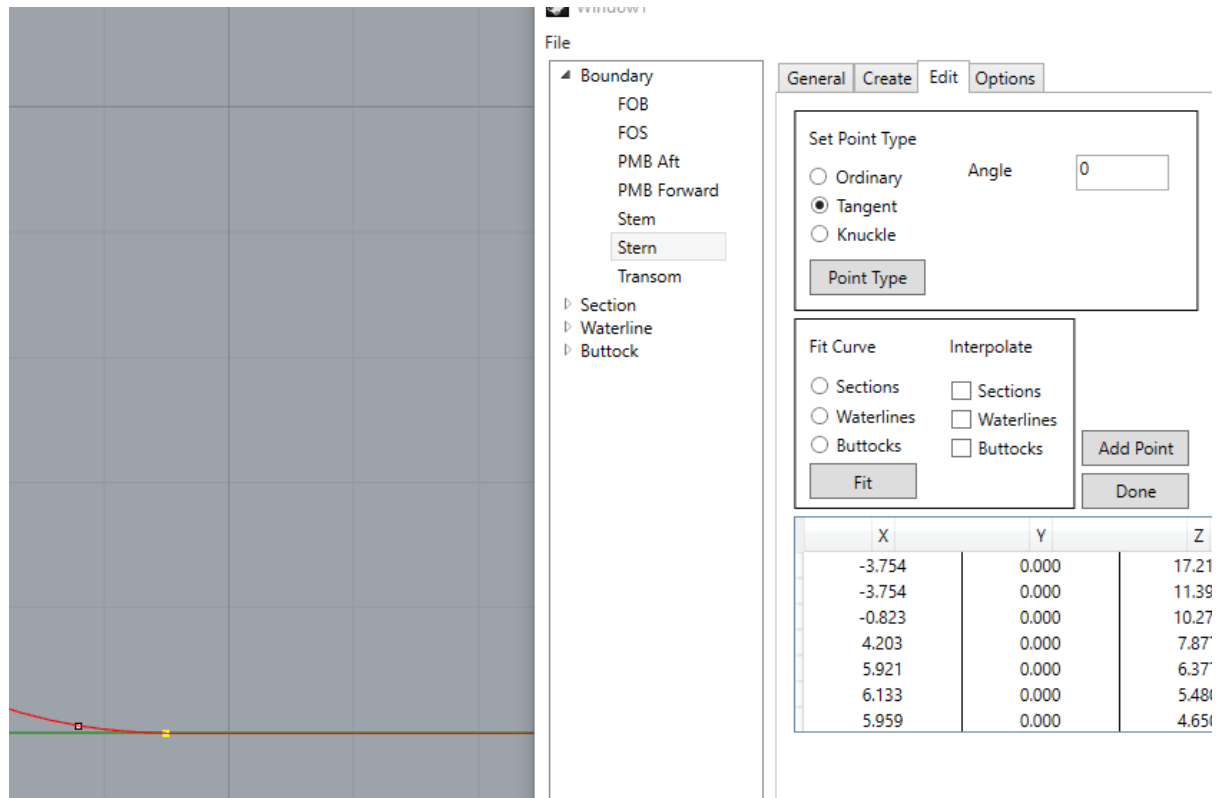
Σχήμα 5.6: Επεξεργασία του πρυμναίου τμήματος του Profile

Στο Tab Edit έχουμε τη δυνατότητα να αλλάξουμε τον τύπο των σημείων. Για την εισαγωγή ενός σημείου Knuckle επιλέγουμε το αντίστοιχο CheckBox, στη συνέχεια επιλέγουμε το Button Point Type και τέλος το σημείο που θέλουμε να αλλάξουμε.



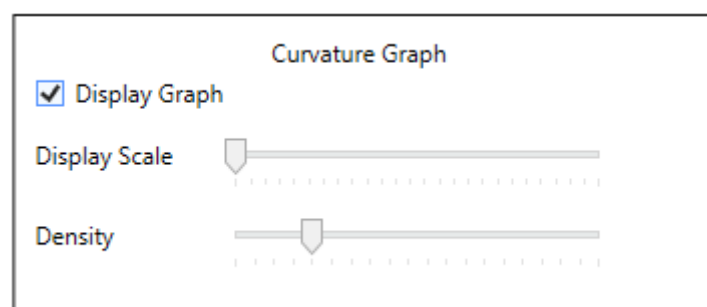
Σχήμα 5.7: Εισαγωγή σημείου Knuckle

Θα μπορούσαμε να εισάγουμε και τις κλίσεις των δύο καμπυλών στο σημείο αυτό, αλλά στην συγκεκριμένη εφαρμογή δεν είναι απαραίτητο. Για την εισαγωγή ενός σημείου τύπου Tangent ακολουθούμε αντίστοιχη διαδικασία με μόνη διαφορά την εισαγωγή και της κοινής κλίσης των δύο καμπυλών στο σημείο αυτό.



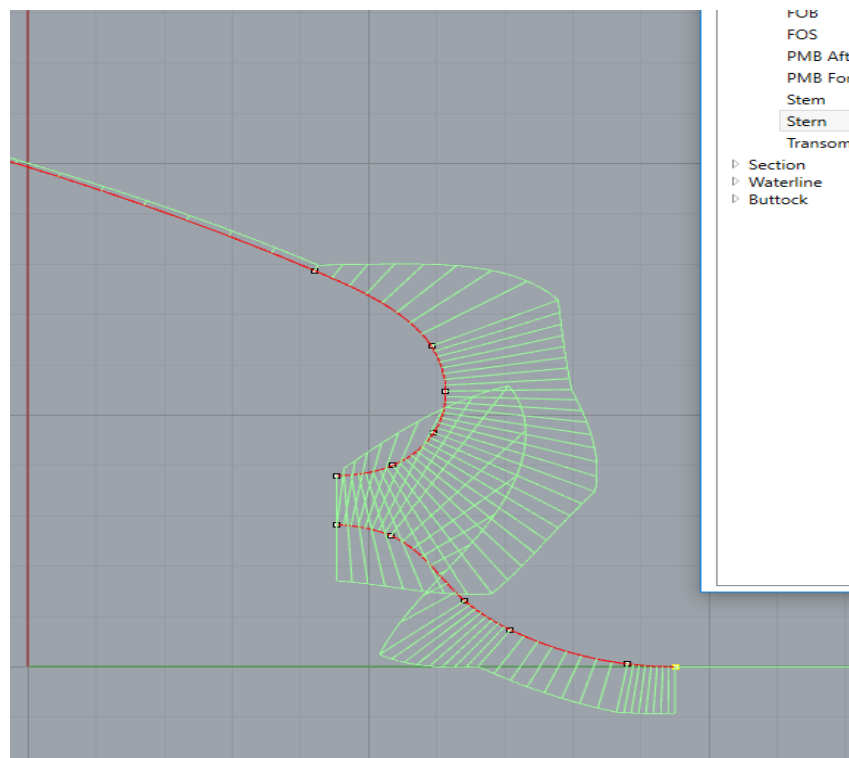
Σχήμα 5.8: Εισαγωγή σημείου Tangent

Για να εμφανίσουμε τα διανύσματα της καμπυλότητας επιλέγουμε το CheckBox Display Graph. Αν χρειαστεί να μεταβάλλουμε το μήκος ή την πυκνότητα των διανυσμάτων χρησιμοποιούμε τα αντίστοιχα εργαλεία.



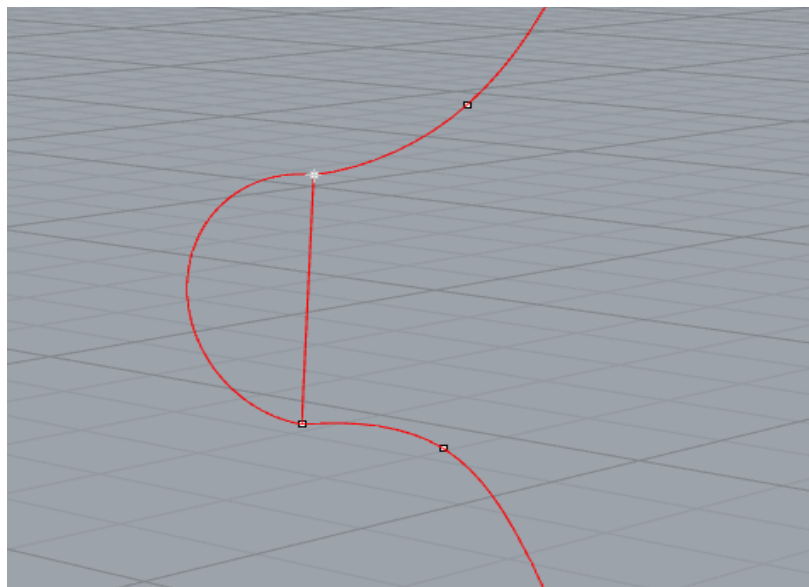
Σχήμα 5.9: Ενεργοποίηση διανυσμάτων καμπυλότητας

Όταν ο χρήστης επιτύχει το επιθυμητό αποτέλεσμα εξομάλυνσης επιλέγει το Button Done, προκειμένου να τελειώσει η επεξεργασία της συγκεκριμένης καμπύλης.



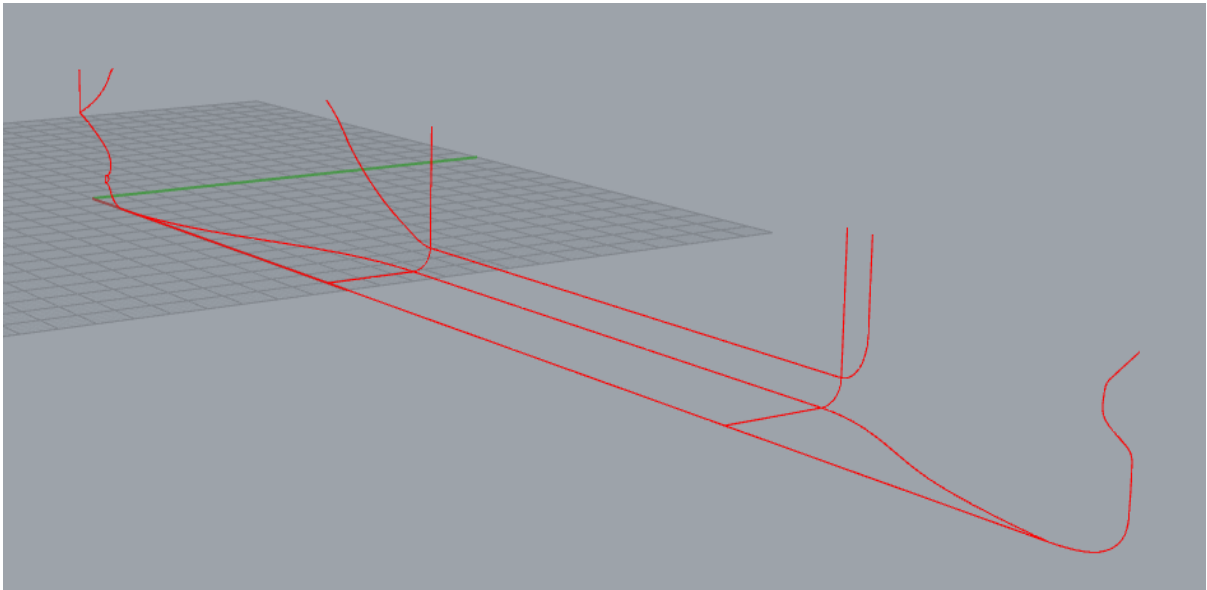
Σχήμα 5.10: Διανύσματα καμπυλότητας

Για την εισαγωγή του ημικύκλιου της χοάνης της έλικας, στο Create Tab πατάμε το Button Shaft, όσο ακόμα επεξεργαζόμαστε το προφίλ της πρύμνης και επιλέγουμε τα δύο σημεία του προφίλ που καθορίζουν τη θέση της χοάνης.



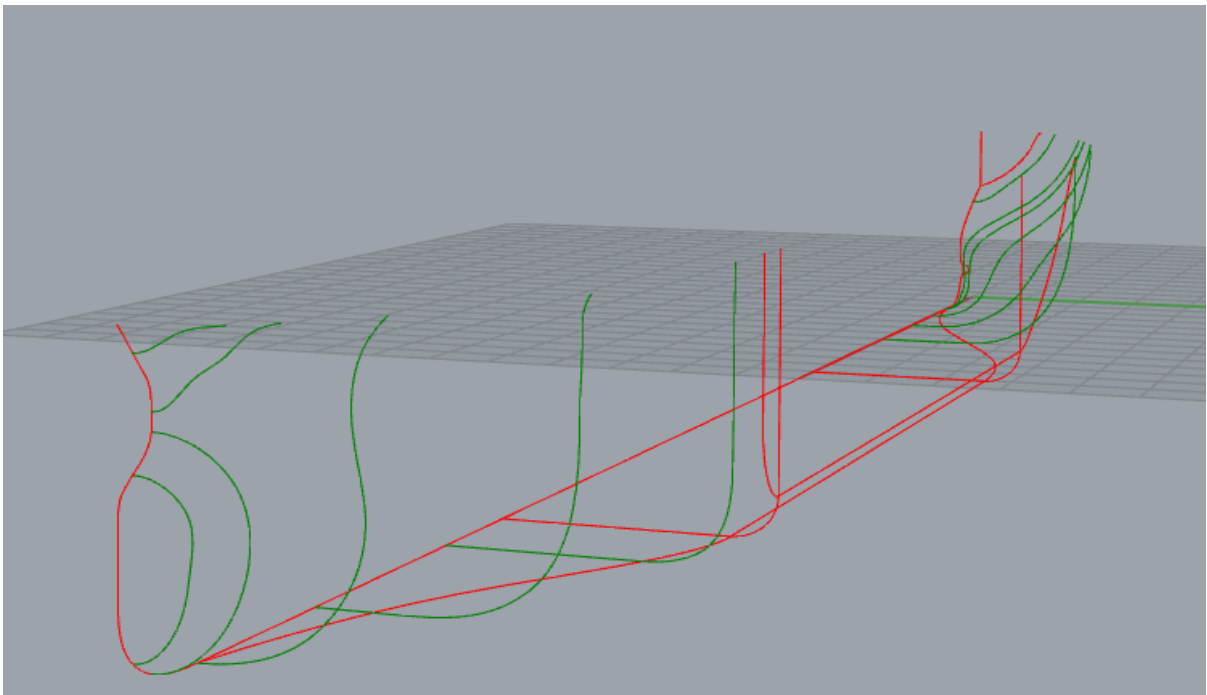
Σχήμα 5.11: Ημικύκλιο χοάνης της έλικας

Αφού ολοκληρωθεί η επεξεργασία και εξομάλυνση όλων των Boundary καμπυλών, μπορεί ο χρήστης να εισάγει τις καμπύλες των νομέων (Sections).



Σχήμα 5.12: Ολοκληρωμένες Boundary καμπύλες του πλοίου

Για την εισαγωγή των νομέων στο Create Tab επιλέγουμε το Section CheckBox και στη συνέχεια την εντολή Open. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να ανοίξουμε όλα τα αρχεία των νομέων που θέλουμε να προσθέσουμε.



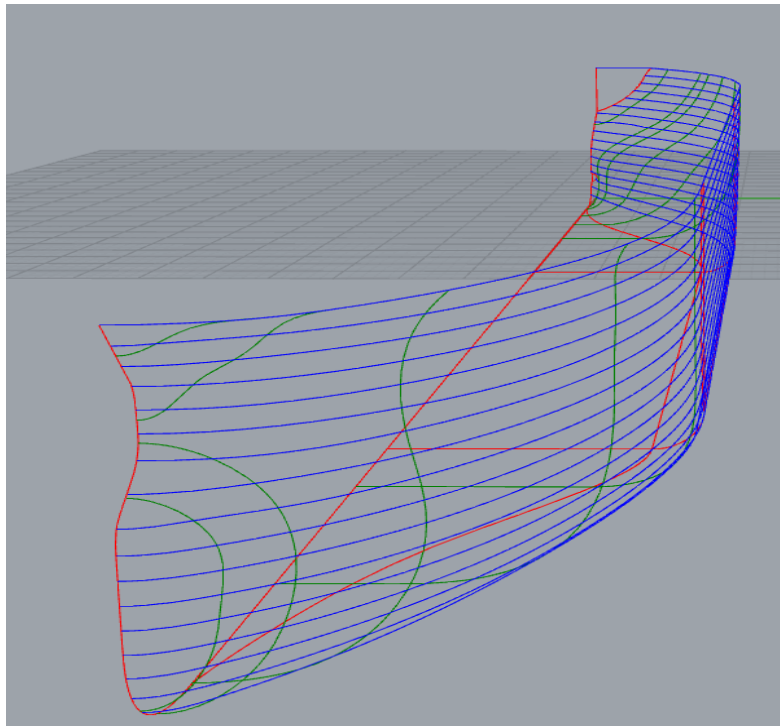
Σχήμα 5.13: Boundary και Section καμπύλες του πλοίου

Όπως μπορούμε να δούμε και από το παραπάνω σχήμα, οι νομείς που εισάγαμε προσαρμόζονται αυτόματα στις ήδη υπάρχουσες καμπύλες FOS και FOB. Το πρόγραμμα θα μετακινήσει τα κατάλληλα σημεία των νομέων ώστε να τέμνονται με τις καμπύλες Boundary και θα ορίσει και τις κλίσεις στα σημεία Tangent.

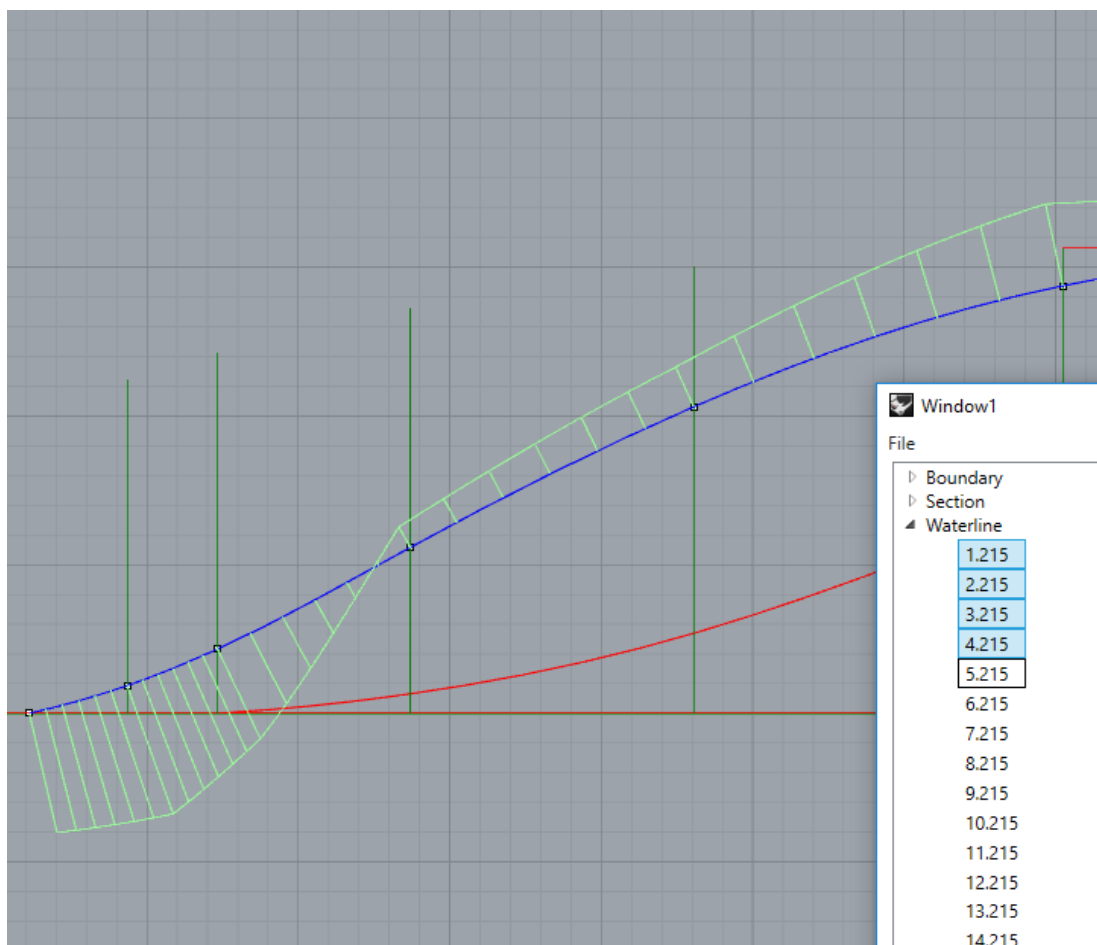
Αφού γίνει εξομάλυνση των νομέων μέσω της εντολής Edit και κατάλληλων μετακινήσεων σημείων, μπορούμε πλέον να περάσουμε τις καμπύλες Waterline. Αρχικά στο Create Tab επιλέγουμε το Waterline CheckBox, έπειτα θα πρέπει να επιλέξουμε ανάμεσα στη δημιουργία μιας μόνο καμπύλης σε κάποια συγκεκριμένη θέση ή την δημιουργία καμπυλών σε ένα διάστημα με κάπιο βήμα που θα ορίσει ο χρήστης. Στο παράδειγμά μας θα κατασκευάσουμε καμπύλες Waterline με βήμα 1 m, που θα καταλαμβάνουν όλο το κοίλο του πλοίου. Επιλέγουμε επομένως το Range CheckBox και εισάγουμε παρακάτω την αρχή και το τέλος του διαστήματος, καθώς και το βήμα. Όταν συμπληρώσουμε τα απαραίτητα κελιά επιλέγουμε τον τύπο των καμπυλών που θα θέλαμε το πρόγραμμα να αναζητήσει τομές με το επίπεδο στο οποίο ανήκει κάθε ζητούμενη καμπύλη Waterline και πατάμε το Create Button.

Σχήμα 5.14: Διαδικασία δημιουργίας Waterlines

Όπως και στην εισαγωγή των νομέων οι παραγόμενες Waterlines θα είναι προσαρμοσμένες έτσι ώστε να υπάρχουν τα σημεία τομής με τις κατάλληλες καμπύλες Boundary καθώς και οι κατάλληλες ιδιότητες των σημείων τομής. Η εξομάλυνση των καμπυλών Waterlines γίνεται με επιλογή κάθε φορά της καμπύλης που θέλουμε να εξομαλύνουμε από το δέντρο και εμφάνιση της καμπυλότητας. Μια άλλη χρήσιμη λειτουργία του προγράμματος είναι η απόκρυψη των καμπυλών που μπορεί να εμποδίζουν την επεξεργασία κάποιας άλλης. Αυτή η λειτουργία πραγματοποιείται από το δέντρο επιλέγοντας την εντολή Hide. Αντίστοιχα αν θέλουμε να εμφανίσουμε κάποια καμπύλη επιλέγουμε την εντολή Show.



Σχήμα 5.15: Καμπύλες Waterline του υπο σχεδίαση πλοίου



Σχήμα 5.16: Εξομάλυνση καμπύλης Waterline

Αφού εξομαλύνει ο χρήστης τις καμπύλες Waterline, θα πρέπει να ξαναδημιουργήσει τις καμπύλες Section προκειμένου να τέμνονται με τις εξομαλυμένες καμπύλες Waterline. Από το Edit Tab επιλέγει ο χρήστης τις καμπύλες που θέλει να δημιουργήσει ξανά, καθώς και τις καμπύλες από τις οποίες θα αναζητήσει τα σημεία τομής, όπως φαίνεται παρακάτω:

Fit Curve	Interpolate
<input checked="" type="radio"/> Sections	<input type="checkbox"/> Sections
<input type="radio"/> Waterlines	<input checked="" type="checkbox"/> Waterlines
<input type="radio"/> Buttocks	<input type="checkbox"/> Buttocks
<input type="button" value="Fit"/>	

Σχήμα 5.17: Fit καμπυλών Sections

Στη συνέχεια ο χρήστης επιλέγει την εντολή Fit, ώστε το πρόγραμμα να ξαναπεράσει τους νομείς Sections. Η ίδια διαδικασία πραγματοποιείται μέχρι μια ικανοποιητική σύγκλιση των σημείων τομής των καμπυλών.

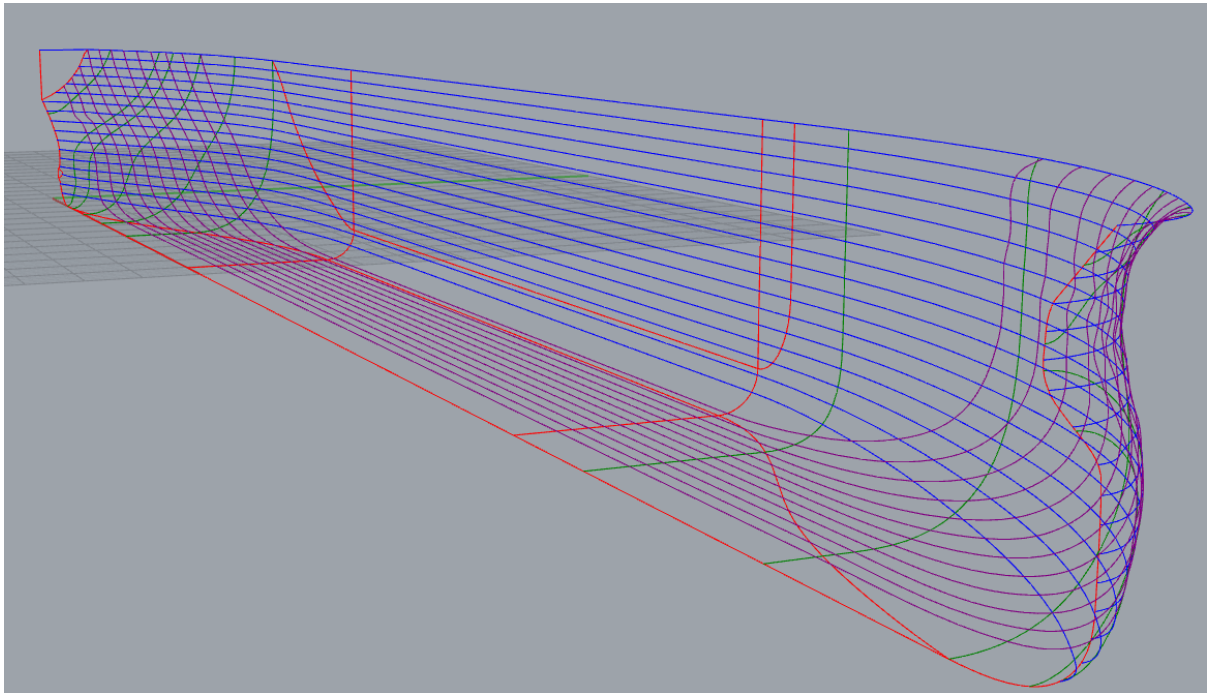
Όταν οι καμπύλες έχουν εξομαλυνθεί σε ικανοποιητικό δημιουργούμε και τις καμπύλες Buttock, με τρόπο όμοιο με αυτόν που δημιουργήσαμε τις καμπύλες Waterline, με τη διαφορά ότι θα αναζητήσουμε σημεία τομής και στις καμπύλες Section και στις καμπύλες Waterline.

Curve Type			
<input type="radio"/> Boundary	<input type="radio"/> Section	<input type="radio"/> Waterline	<input checked="" type="radio"/> Buttock

Selection		Interpolation	
<input checked="" type="radio"/> Range	<input type="radio"/> Series	<input checked="" type="checkbox"/> Section	<input checked="" type="checkbox"/> Waterline
Series	<input type="text" value=""/>	<input type="checkbox"/> Buttock	
From	<input type="text" value="2.65"/>		
To	<input type="text" value="13.65"/>		
Step	<input type="text" value="1"/>		
Boundary Type	<input type="button" value="v"/>		
		<input type="button" value="Create"/>	

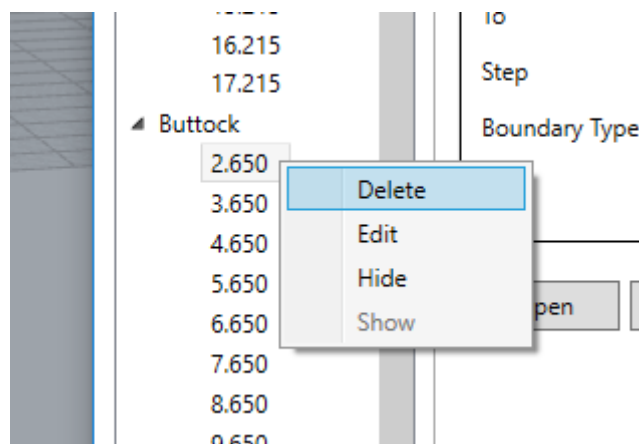
Σχήμα 5.18: Δημιουργία καμπυλών Buttock

Το τελικό αποτέλεσμα φαίνεται στο παρακάτω σχήμα:



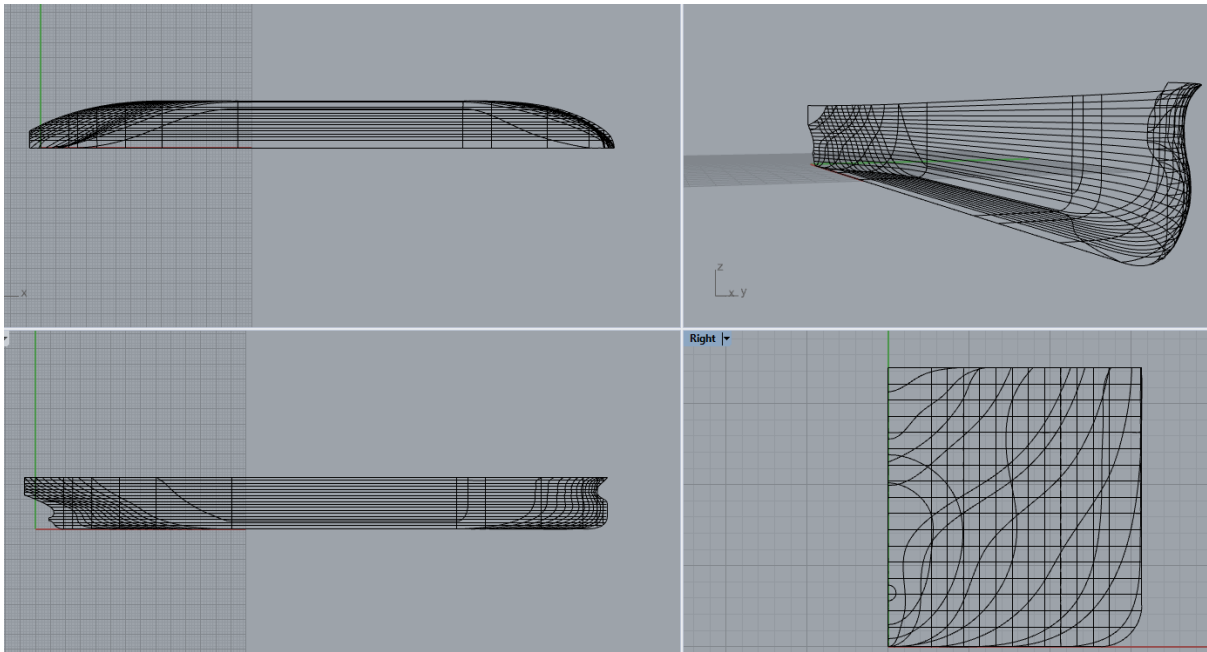
Σχήμα 5.19: Σχέδιο πλοίου με όλους του δυνατούς τύπος καμπυλών

Το δέντρο παρέχει επίσης στον χρήστη τη δυνατότητα διαγραφής καμπυλών που έχει δημιουργήσει, επιλέγοντας της εντολή Delete.



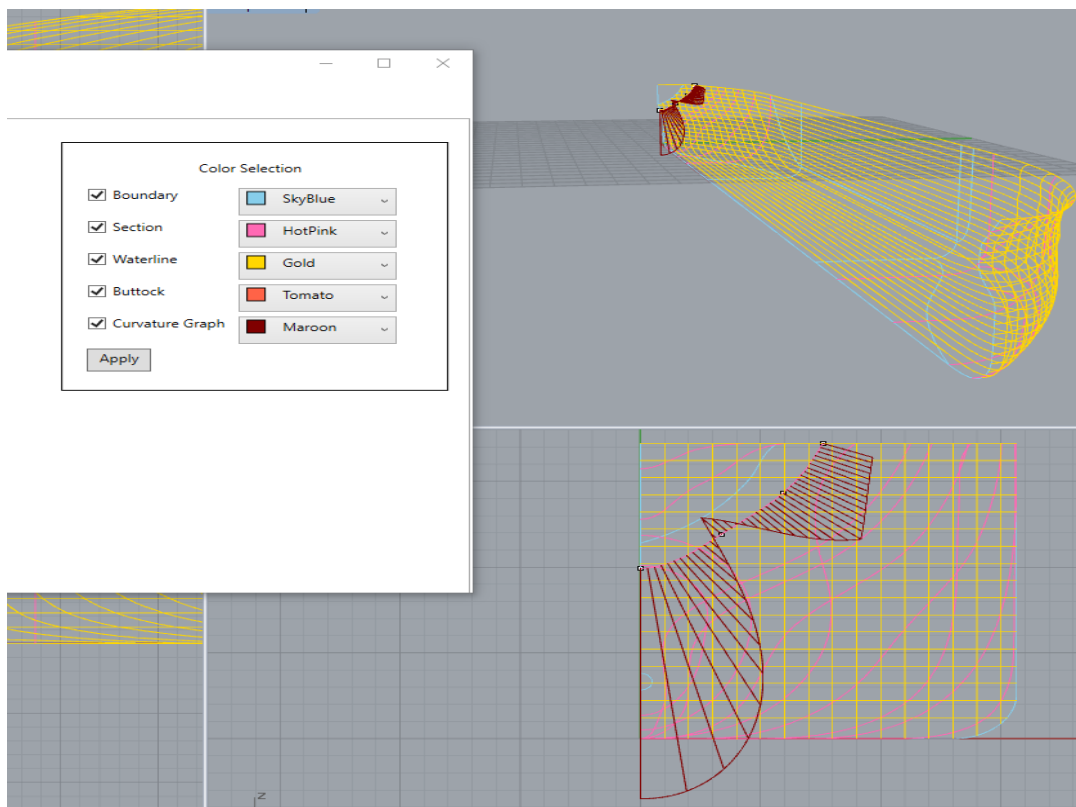
Σχήμα 5.20: Διαγραφή καμπύλης

Η αποθήκευση του αρχείου γίνεται από το Dropdown – Menu επιλέγοντας την εντολή Save, ενώ αντίστοιχα η φόρτωση κάποιου αρχείου γίνεται με την εντολή Load. Αν ο χρήστης θέλει να εξάγει το αρχείο σε μορφή που να μπορεί να το επεξεργαστεί με τα εργαλεία που του παρέχει το Rhinoceros επιλέγει την εντολή Export Lines. Παρακάτω φαίνεται το μοντέλο που δημιουργήσαμε σε αρχείο .3dm που μπορεί να επεξεργαστεί το Rhino.



Σχήμα 5.21: Αρχείο Export καμπυλών

Μια επιπλέον λειτουργία του προγράμματος προκειμένου η χρήση του να είναι λίγο πιο ευχάριστη από τον χρήστη είναι η αλλαγή του χρώματος των καμπυλών από το Options Tab. Ο χρήστης μπορεί να αλλάξει το χρώμα που θα έχει κάθε κατηγορία καμπυλών καθώς και τα διανύσματα καμπυλότητας.



Σχήμα 5.22: Αλλαγή χρώματος καμπυλών

6

Επίλογος

Έχοντας ολοκληρώσει ένα παράδειγμα σχεδίασης των βασικών ναυπηγικών γραμμών ενός συμβατικού πλοίου με τη χρήση του Shiplines, μπορούμε να συνοψίσουμε το περιεχόμενο και τα αποτελέσματα της υλοποίησης της παρούσας διπλωματικής εργασίας.

6.1 Σύνοψη και συμπεράσματα

Η εργασία καταπιάστηκε με το πρόβλημα της δημιουργίας ενός νέου λογισμικού για τη σχεδίαση των ναυπηγικών γραμμών συμβατικών πλοίων, το οποίο θα αποτελεί Plugin του λογισμικού Rhinoceros. Αυτό επιτεύχθηκε με την κατασκευή σου παραθύρου WPF καθώς και όλων των μεθόδων και κλάσεων που χρειάζονται για την λειτουργία του.

Μετά την δημιουργία καθώς και τη λειτουργία του Shiplines μπορεί κανείς να συμπεράνει ότι:

- Το Plugin έχει πολύ ομαλή και μικρή καμπύλη εκμάθησης και αποτελεί πολύ καλό μέσο για την εκπαίδευση νέων χρηστών πριν την γνωριμία τους με άλλα πιο δύσκολα λογισμικά.
- Η διαδικασία δημιουργίας των καμπυλών είναι πάρα πολύ γρήγορη.
- Παρέχει στον χρήστη τη δυνατότητα επεξεργασίας του μοντέλου της γάτρας με τα εργαλεία που παρέχει το Rhinoceros.
- Επιπλέον είναι επεκτάσιμο.

6.2 Μελλοντικές επεκτάσεις

Φυσικά η εφαρμογή και η υλοποίησή της είναι μια πρώτη απόπειρα δημιουργίας Plugin στο Rhinoceros που να παρέχει τη δυνατότητα αυτοματοποιημένης παραγωγής ναυπηγικών καμπυλών, συνεπώς υπάρχει περιθώριο για αρκετές και διαφορετικές μελλοντικές επεκτάσεις και βελτιώσεις.

Μερικές ενδιαφέρουσες προτάσεις για την περαιτέρω βελτίωσή του είναι:

- Η εισαγωγή όλων των απαραίτητων κλάσεων, μεθόδων και εργαλείων του παραθύρου προκειμένου να δημιουργηθούν και οι επιφάνειες του πλοίου, βάση των οποίων θα μπορούν να γίνουν πιο ακριβή υδροστατικοί και υδροδυναμικού υπολογισμοί.
- Η περαιτέρω βελτίωση του παρών κώδικα με καλύτερη εφαρμογή του αρχιτεκτονικού προτύπου MVVM, βελτιστοποίηση των κλάσεων και μεθόδων και όλων των δομών που απαρτίζουν το παράθυρο.
- Η χρήση διαφορετικού framework για το παράθυρο που πιθανόν να απαιτεί και τη μεταφορά όλου του κώδικα σε διαφορετική γλώσσα (π.χ. C++). Άλλα frameworks εξελίσσονται με ταχύτερους ρυθμούς και παρέχουν περισσότερες λειτουργίες, καθώς και μεγαλύτερη ευκολία στη σχεδίαση του παραθύρου με αποτέλεσμα να επιταχύνεται η διαδικασία. Ένα τέτοιο παράδειγμα framework είναι το Qt, που κυριαρχεί στην αγορά συγκριτικά με τα υπόλοιπα.

7

Βιβλιογραφία

1. Δρ. Μηχ. Κ.Β. Κώστας, 3D Σχεδίαση & Υπολογισμοί με το Rhino 3D.
2. Π.Δ. Κακλής, Καθηγητής ΕΜΠ, Συστήματα CAD/CAM για την Σχεδίαση και Κατασκευή Πλοίων.
3. Νικόλαος Μπιλάλης, Εμμανουήλ Μαραβελάκης, Συστήματα CAD/CAM & Τρισδιάστατη Μοντελοποίηση.
4. Αποστόλου Δ. Παπανικολάου, Καθηγητή ΕΜΠ, Μελέτη Πλοίου Μεθοδολογίες Προμελέτης Τεύχος 1.
5. Αποστόλου Δ. Παπανικολάου, Καθηγητή ΕΜΠ, Μελέτη Πλοίου Μεθοδολογίες Προμελέτης Τεύχος 2.
6. Διαφάνειες μαθήματος Ναυπηγικού Σχεδίου, Σχολής Ναυπηγών Μηχανολόγων Μηχανικών.