

CMN6302: Major Project (BSc)

CMN6302.F3

Progress Presentation 1

Διαμαντής Μιχαήλ-Ευάγγελος



ANIMATION



AUDIO



BUSINESS



FILM



GAMES



WEB & MOBILE



Τίτλος & Περιγραφή

Τίτλος

The “Snapshot”: A complete, full-fledged and open source, video game, C++ data serialization solution and architecture.

Σύντομη Περιγραφή

Δημιουργία και σχεδιασμός μίας βιβλιοθήκης με το ελάχιστο δυνατό memory footprint για χρήση της σε βιντεοπαιχνίδια, με τη δυνατότητα διατήρησης και επαναφοράς των object references κατά τη διάρκεια του Serialization και του Deserialization.

Χρονοδιάγραμμα

Διαμαντής Μιχαήλ-Ευάγγελος				Μάρτιος (Εβδ.)				Απρίλιος (Εβδ.)				Μάιος (Εβδ.)				Ιούνιος (Εβδ.)				Ιούλιος (Εβδ.)			
13-13246				1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Συλλογή πληροφοριών																							
Μελετη βιβλιογραφίας																							
Εξέταση και ανάλυση εργαλείων																							
Σχεδιασμός βιβλιοθήκης																							
Υλοποίηση του σχεδιασμού																							
Εξέταση αποτελεσμάτων																							
Συγγραφή πτυχιακής εργασίας																							

Γραπτό μέρος

Κεφάλαιο 2: Θεωρητικό πλαίσιο

- 2.1 Εισαγωγή κεφαλαίου
- 2.2 Persistent Data στην Πληροφορική
- 2.3 Data Serialization
 - 2.3.1 Ορισμός του Serialization-Deserialization
 - 2.3.2 Τύποι Serialization
- 2.4 Memory management, addresses και serialization
 - 2.4.1 Logical και Physical addresses
 - 2.4.2 Memory Management και Virtual Memory σε ένα λειτουργικό σύστημα
 - 2.4.3 Τα memory addresses κατά το serialization
- 2.5 Επισκόπηση Serializers προς ανάλυση
- 2.6 Foreign Function Interfaces
 - 2.6.1 Ορισμός του Foreign Function Interface (FFI)
 - 2.6.2 Data Marshalling in FFIs
 - 2.6.3 Error Handling in FFIs
 - 2.6.4 Performance in FFIs
- 2.7 Library wrappers
 - 2.7.1 Managed και Unmanaged κώδικας
 - 2.7.2 Ορισμός των Library/DLL Wrappers στην Πληροφορική
 - 2.7.3 Managed και Unmanaged code wrappers
- 2.8 Συστήματα και λύσεις serializer στις μηχανές βιντεοπαιχνιδιών
 - 2.8.1 Unity
 - 2.8.2 Unreal Engine

Κεφάλαιο 3: Μεθοδολογία

- 3.1 Εισαγωγή κεφαλαίου
- 3.2 Ανάλυση απαιτήσεων της συνολικής λύσης
- 3.3 Ανάλυση επιλογής C/C++ ως βασική γλώσσα συγγραφής
- 3.4 Base serializer (protobuf ή MsgPack)
- 3.5 Χαρακτηριστικά λύσης
 - 3.5.1 Χαρακτηριστικά βιβλιοθήκης
 - 3.5.2 Χαρακτηριστικά FFI
 - 3.5.3 Χαρακτηριστικά Wrapper
- 3.6 Διατήρηση των references μεταξύ των instances μετά το serialization (SMRI)
- 3.7 Test environment στη Unity
- 3.8 Σχεδιασμός λύσης
 - 3.8.1 Σχεδιασμός βασικής βιβλιοθήκης σε C/C++
 - 3.8.2 Σχεδιασμός Foreign Function Interface
 - 3.8.3 Σχεδιασμός C# Wrapper
- 3.9 Σχεδιασμός test environment στη Unity
- 3.10.1 Σύγχρονη ή Ασύγχρονη προσέγγιση;
- 3.11 Βαθμός πραγματοποίησης στόχων διατριβής
- 3.12 Ανάδειξη σχεδιαστικών στοιχείων

Γραπτό μέρος – Το Do

- Περίληψη
- Abstract
- Ευχαριστίες
- Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή

- 1.1 Πρόλογος - Εισαγωγή
- 1.2 Στόχος και συμβολή
- 1.3 Τρόπος προσέγγισης
- 1.4 Δομή πτυχιακής εργασίας

Κεφάλαιο 4: Υλοποίηση

- 4.1 Εισαγωγή κεφαλαίου
- 4.2 Υλοποίηση API και DLL
- 4.3 Υλοποίηση FFI
- 4.4 Υλοποίηση C# wrapper
- 4.5 Στήσιμο test environment στη Unity
- 4.6 Ανάλυση τελικής λύσης
- 4.7 Προβλήματα κατά την υλοποίηση

Κεφάλαιο 5: Δοκιμές

- 5.1 Εισαγωγή κεφαλαίου
- 5.2 Δεδομένα προς serialization
 - 5.2.1 Data Complexity (Ίσως)
- 5.3 Προβλήματα κατά την πειραματική διαδικασία

Κεφάλαιο 6: Αποτελέσματα

- 6.1 Εισαγωγή κεφαλαίου
- 6.2 Ανάλυση παραχθέντος serialized αρχείου δοκιμών
- 6.3 Αξιολόγηση χρήσης βιβλιοθήκης
- 6.4 Σημειώσεις βελτιστοποίησης

Κεφάλαιο 7: Συμπεράσματα

- 7.1 Επισκόπηση της διατριβής
- 7.2 Ανάδειξη κύριων αποτελεσμάτων (Ίσως)
- 7.3 Ποσοστό πραγματοποίησης στόχων
- 7.4 Περιγραφή μελλοντικών βελτιώσεων και επεκτάσεων
- 7.5 Επίλογος

Βιβλιογραφία

Παραρτήματα

Video

Πρακτικό μέρος

C++ Library

- Data containerization
- Access API
- Error Handling and error codes
- Internal Utility Library
- I/O Handling
- Data Handling – Serialization
- Data Handling – Deserialization
- FFI

FFI

- C++ Side
- C# Side

C# Wrapper

- Library wrapping
- Data Marshalling
- Exposed API
- Error Code Handling

Architecture

- Proposed Usage
- World Loader
- Save/Load Manager
- ISnapshot
- ISnapshotModel

Σας ευχαριστώ πολύ
για τον χρόνο σας!