Ως έκδοση v1.0 ονομάστε όλα τα τελικά τεχνικά κείμενο που θα παραδώσετε με την τελική παράδοση του project σας. Αν το τελευταίο τεχνικό κείμενο για κάτι ήταν η έκδοση v0.2 για παράδειγμα και σε αυτή δεν έγινε καμία αλλαγή, στο v1.0 απλά να αναφέρετε «είναι η έκδοση v0.2 χωρίς καμία αλλαγή».

Όλες τις εκδόσεις v1.0, για ότι έχετε παραδώσει έως τώρα, είτε έχει αλλάξει στην τελική έκδοση (π.χ. το Team-plan-v0.1 θα έχει οπωσδήποτε προσθήκες), είτε αν δεν έχει αλλάξει την τελευταία έκδοση, όπως περιγράφεται αναλυτικά στο τμήμα «Περιγραφές Τεχνικών Κειμένων».

Κυρίως σε αυτή τη φάση θα πρέπει να ασχοληθείτε με την τελική έκδοση του κώδικα Project-code-v1.0, η οποία θα πρέπει να περιλαμβάνει ικανοποιητική λειτουργικότητα τουλάχιστον για τα use cases του έργου σας τα οποία αναλύσατε.

**Team-plan(αφορά την οργάνωση της ομάδας σας για το projectτης Τ.Λ.)**

Στην τελική έκδοση (v1.0), θα πρέπει να εμπλουτίσετε το κείμενο με: α) την κατανομή προσπάθειας της ομάδας σας όπως περιγράφεται στο τμήμα «Βαθμολογία Project», β) το τελικό Gantt chart με ότι αλλαγές έγιναν, γ) την ανάθεση έργου στα μέλη της ομάδας σας σύμφωνα με το τελικό Gantt chart, και δ) συμπεράσματα από τον τρόπο εργασίας ως ομάδα (δηλαδή αν η μέθοδος που επιλέξατε δούλεψε, ή όχι, και τι θα αλλάζατε τώρα αν μπορούσατε να γυρίσετε το χρόνο πίσω).

Scene builder για τα fxml κι βιβλιοθήκες.

**Project-description(αφορά την περιγραφή του έργου)**

Στην τελική έκδοση (v1.0), όποιες οθόνες έχουν υλοποιηθεί θα πρέπει να ενσωματωθούν, δείχνοντας το αρχικό παράδειγμα (mock-up) και την τελική οθόνη που υλοποιήθηκε. Μπορεί στην τελική έκδοση να υπάρχει και το πλήρες ιστορικό των οθονών.

**Project-plan(αφορά τον χρονοπρογραμματισμό του έργου)**

**Risk-assessment(αφορά την εκτίμηση κινδύνου του έργου)**

**Team-risk-assessment(αφορά την εκτίμηση κινδύνου του projectκαι όχι του έργου)**

Εάν υποβάλετε team-risk-assessment τότε, αν κατά την πορεία του project κρίνετε ότι η αρχική σας εκτίμηση είχε κάπου λάθος, μπορείτε να επανέλθετε με νέα έκδοση, αλλά δεν είναι ζητούμενο. Όμως ζητούμενο είναι, αν υποβάλετε team-risk-assessment, να υποβάλλετε και την τελική έκδοση (v1.0) στην οποία να εξηγήσετε αν κάποιοι κίνδυνοι όντως έγιναν γεγονότα και πώς τους αντιμετωπίσατε.

Βιβλιοθήκη Javafx

Έσκασε το github και έσβησε κώδικα

**Use-cases**

**Robustness-diagrams**

**Sequence-diagrams**

**(Domain-model/) Class-diagram**

**Project-code (αφορά τον κώδικα του έργου)**

α) Σε κάποια git ο κώδικας είναι μοιρασμένος σε διάφορους χώρους (repos, folders) με οργάνωση που έχει νόημα ίσως για την ομάδα (δύσκολο να ελεγχθεί ο κώδικας από εμάς που τον εξετάζουμε στη λογική κάθε κλάση χωριστά. Καλύτερα ο κώδικας να είναι κάπου ενιαία και αν θέλετε και άλλες μορφές οργάνωσης να μην ακυρώνουν αυτή.

β) Σε πολλά git δεν υπάρχει καμία περιγραφή πώς ο κώδικας μπορεί να εκτελεστεί, τι άλλο απαιτείται για να εκτελεστεί και τι πρέπει να κάνει ο χρήστης όταν εκτελεστεί (π.χ. πρέπει να εισάγει τα εξής δεδομένα, πρέπει να κάνει κάτι άλλο;). Παρακαλώ προσθέστε μια σχετική περιγραφή.

Login εξήγηση, βιβλιοθήκες ποιες και πως στο intellij, κλάσεις από sequence, Scene builder