Δημιουργήστε το δικό σας παιχνίδι εξερεύνησης

Τελική Εργασία στον Προγραμματισμό Υπολογιστών με C++ ακαδ. Έτους 2026-27



Εικόνα από το demo της εργασίας. Το demo μπορείτε να το κατεβάσετε εδώ: https://auebgr-my.sharepoint.com/:u:/g/personal/gepap_aueb_gr/IQC2GkGxTWyLRLxNennWD22xARHMnPKqf7jb-zp5XW0MYDs?e=vlDK6r. Το παιχνίδι που σας δίνεται ως δείγμα έχει αρκετά πολύπλοκη υλοποίηση και δεν απαιτείται να υλοποιήσετε κάτι σε τέτοιο βαθμό. Δίνεται απλά για να πάρετε ιδέες για το δικό σας παιχνίδι.

Σκοπός της Εργασίας

Χρησιμοποιήστε τη βιβλιοθήκη Simple Game Graphics (https://github.com/cgaueb/sgg) για να δημιουργήσετε το δικό σας point-and-click adventure game.

Τί Είναι ένα Παιχνίδι Περιπέτειας

Σε ένα τέτοιο παιχνίδι, προσπαθούμε να λύσουμε γρίφους, να συλλέξουμε και συνδυάσουμε αντικείμενα έτσι ώστε να οδηγήσουμε την πλοκή ενός σεναρίου προς το τέλος του παιχνιδιού. Ένα παιχνίδι περιπέτειας μπορεί να έχει από ένα πολύ απλό σενάριο μέχρι κάποια ιστορία με μεγάλο βάθος, εναλλακτικές εκβάσεις και χτίσιμο χαρακτήρων. Οι κόσμοι που εκτυλίσσονται μέσα οι ιστορίες συνήθως είναι αρκετά εκτενείς αλλά στην εργασία σας, προσπαθήστε να κάνετε κάτι χαρακτηριστικό μεν, αλλά και ταυτόχρονα περιορισμένο σε έκταση ώστε να είναι εφικτή η υλοποίησή του στον δεδομένο χρόνο.

Τυπικά, σε ένα παιχνίδι περιπέτειας (τρίτου προσώπου) ο κόσμος είναι κατακερματισμένος σε «τοποθεσίες» που χονδρικά αντιστοιχούν σε διαφορετικές «οθόνες» του παιχνιδιού, δηλαδή περιεχόμενο που χωράει να δειχτεί στο παράθυρο που παίζουμε. Η μετάβαση από τοποθεσία σε

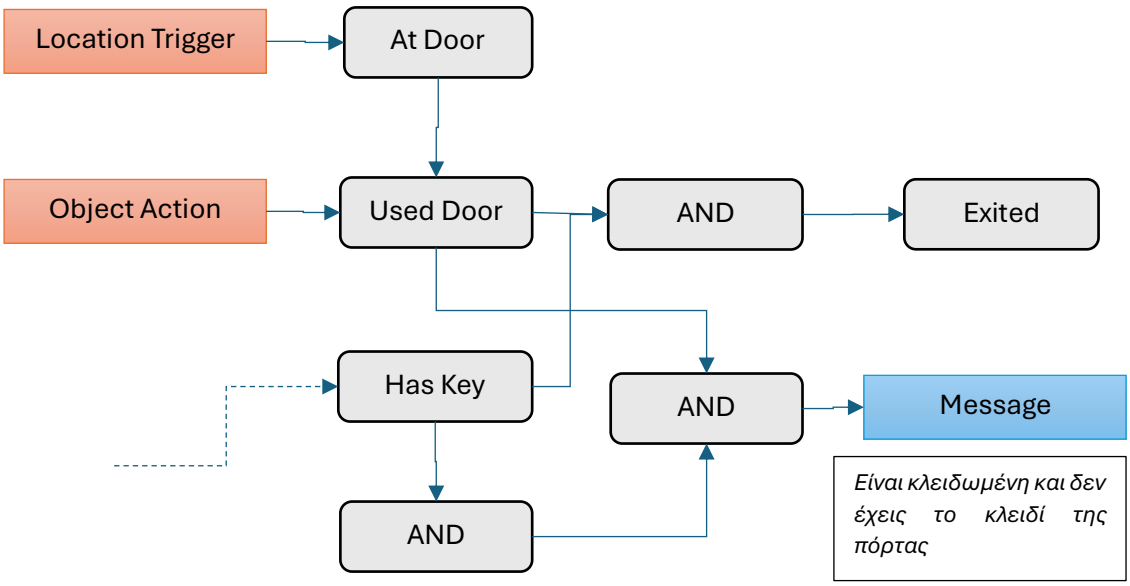
τοποθεσία μπορεί να είναι είτε διακριτή είτε συνεχής. Στη διακριτή περίπτωση, ενεργοποιώντας μια «έξοδο» από την τοποθεσία που βρισκόμαστε, κλείνει η τρέχουσα τοποθεσία και στη θέση της εμφανίζεται η νοηματικά ή χωρικά συνδεδεμένη μαζί της.

Η διάδραση σε ένα τέτοιο είδος παιχνιδιού κλασικά στηρίζεται στη χρήση μιας καταδεικτικής συσκευής (pointing device) όπως το ποντίκι, για την αναζήτηση και επιλογή ή ενεργοποίηση αντικειμένων και λειτουργιών στην κάθε τοποθεσία. Σε ορισμένες περιπτώσεις, αντικείμενα είναι δυνατό να τα πάρουμε, αφαιρώντας τα από το περιβάλλον και τοποθετώντας τα σε έναν ειδικό χώρο αποθήκευσης τον οποίο «κουβαλάμε» μαζί μας (Inventory). Αυτό μας επιτρέπει να συνδυάσουμε αντικείμενα που βρίσκονται σε διαφορετικές τοποθεσίες ή να συνθέσουμε νέα αντικείμενα.

Από πλευράς γραφικών, συνήθως ένα τέτοιο παιχνίδι (τουλάχιστον παραδοσιακά) ήταν μια κλασική υλοποίηση 2D τοποθεσιών που τις βλέπουμε από το πλάι, αλλά υπάρχουν και παιχνίδια με ισομετρική προβολή ή προβολή από πάνω. Συνήθως, ο ήρωας της περιπέτειας είναι εμφανής και παρουσιάζεται μέσα στην τοποθεσία, αλλά αν σας δυσκολεύει, μπορείτε να θεωρήσετε ότι αυτός/αυτή που εξερευνά το χώρο είστε εσείς.

Πλοκή και Καταστάσεις

Η πλοκή σε ένα παιχνίδι περιπέτειας είναι στην ουσία μια αλληλουχία ενεργοποίησης καταστάσεων υπό προϋποθέσεις. Για παράδειγμα, έστω ότι σκοπός μιας τοποθεσίας είναι να βγούμε από μια κλειδωμένη πόρτα. Ας ονομάσουμε την κατάσταση επιτυχούς εξόδου “Exited”. Για να μεταβούμε σε αυτή την κατάσταση πρέπει να συμβούν 2 πράγματα: α) να έχουμε φτάσει κοντά στην πόρτα (κατάσταση “At Door”) και β) να ενεργοποιήσουμε την πόρτα (π.χ. με click πάνω στο εικονίδιο του πόμολου) – κατάσταση “Used Door”. Αν όμως η πόρτα είναι κλειδωμένη, πρέπει πρώτα να έχουμε πάρει το κλειδί (που ας υποθέσουμε ότι είναι σε ένα άλλο σημείο της τοποθεσίας) και άρα να έχουμε περιέλθει πρώτα στην κατάσταση “Has Key”. Πολλές από τις καταστάσεις μπορούν να είναι ενεργοποιημένες ταυτόχρονα και ουσιαστικά αποτελούν συνθήκες/προϋποθέσεις για άλλες. Σχηματικά το παραπάνω παράδειγμα θα ήταν κάτι τέτοιο:



Στο σχήμα αυτό βλέπουμε και κάποια επιπρόσθετα πράγματα. Ροές καταστάσεων μπορούν να συντεθούν με λογικούς τελεστές ώστε να συνδυαστούν. Π.χ. Έχω το κλειδί ΚΑΙ χρησιμοποιώ την πόρτα. Επίσης έτσι μπορώ να δημιουργήσω και διαφορετικά σενάρια αντίδρασης. Π.χ. εδώ αν ΔΕΝ έχουμε το κλειδί (NOT) και επιχειρήσουμε να ανοίξουμε την πόρτα, μπορούμε να ενεργοποιήσουμε μια απάντηση προς το χρήστη «Είναι κλειδωμένη και δεν έχεις το κλειδί της πόρτας».

Είναι χρήσιμο, όσο απλό και αν είναι το παιχνίδι που έχετε σκεφτεί, να καταγράψετε τις αναμενόμενες καταστάσεις ώστε να είστε σίγουροι ότι δεν έχετε ξεχάσει κάτι και ότι δεν υπάρχουν απρόβλεπτοι συνδυασμοί που ενδέχεται να εμποδίσουν την πρόοδο της υπόθεσης. Μπορείτε αν θέλετε να υλοποιήσετε μια μηχανή καταστάσεων για τον έλεγχο των λειτουργιών ώστε να μπορέσετε να δημιουργήσετε πολυπλοκότερες σχέσεις μέσα στο παιχνίδι σας (προαιρετικό – προχωρημένο, βλ. παρακάτω). Στο demo που σας έχουμε ανεβάσει και την επίδειξη δυνατοτήτων, μπορείτε να δείτε το διάγραμμα καταστάσεων ζωντανά πατώντας το πλήκτρο 9.

Προδιαγραφές

Το παιχνίδι σας πρέπει να υπακούει στις ακόλουθες προδιαγραφές.

Χρήση της βιβλιοθήκης SGG. Το χτίσιμο του παιχνιδιού με βάση αυτή τη βιβλιοθήκη για όλη τη διαχείριση παραθύρου, εισόδου και σχεδιασμού γραφικών είναι υποχρεωτικό. Μπορείτε να συνδέσετε κι άλλες βιβλιοθήκες για να εκτελέσετε λειτουργίες που δεν επικαλύπτονται με τις υλοποιημένες από την SGG. Για παράδειγμα, μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη json.hpp για διάβασμα JSON δομημένων αρχείων, την SDL_Image για διάβασμα bitmaps, αλλά δε μπορείτε να χρησιμοποιήσετε τη βασική SDL2/3 ή την OpenGL.

Ελάχιστη συμβατική υλοποίηση. Το παιχνίδι σας θα πρέπει να απαρτίζεται από τουλάχιστον 3 διακριτές τοποθεσίες. Κάθε τοποθεσία θα πρέπει να απαιτεί τη συλλογή ή ακολουθιακή ενεργοποίηση 2 αντικειμένων ή λειτουργιών, αντίστοιχα χωρίς τα οποία δε θα μπορεί να ολοκληρωθεί η πλοκή.

Χρήση Πολυμορφισμού. Σε ένα παιχνίδι, έχουμε πολλές ευκαιρίες να σχεδιάσουμε σωστά τον κώδικά μας με χρήση πολυμορφισμού: η βασική και αφηρημένη κλάση GameObject, static assets, animated assets, events, triggers, states, effects, scenes, κλπ. Όλα μπορούν να συνδεθούν σε ιεραρχίες κλάσεων και να επαναχρησιμοποιήσουν πεδία και μεθόδους. Η κεντρική «μηχανή» του παιχνιδιού σας, μπορεί να καλεί πολυμορφικά μεθόδους χωρίς να ασχολείται με τον ακριβή τους τύπο. Η χρήση κληρονομικότητας και πολυμορφισμού είναι υποχρεωτική.

Χρήση δομών δεδομένων. Σε ένα παιχνίδι, είναι σχεδόν πάντα απαραίτητο να διατηρούμε αντικείμενα σε σειριακές ή άλλου τύπου δομές. Για παράδειγμα, μια συλλογή μπορεί να κρατάει τα αντικείμενα μίας τοποθεσίας και μια άλλη συλλογή τις τοποθεσίες μας. Αν χρησιμοποιούμε συμβάντα ή μηνύματα από και προς αντικείμενα, αυτά συσσωρεύονται προς εκτέλεση σε ουρές μηνυμάτων. Αν θέλουμε να κρατάμε ευρετήρια αντιστοίχισης δεικτών σε αντικείμενα και ονομάτων, χρειαζόμαστε maps, κ.ο.κ. Στην εργασία αυτή θα πρέπει να χρησιμοποιήσετε τουλάχιστον μια δομή της STL.

Χρήση τεχνητής νοημοσύνης. Ρητά απαγορεύεται η χρήση τεχνητής νοημοσύνης για τον προγραμματισμό της εφαρμογής πέρα από στοιχειώδεις εργασίες όπως code auto-completion (π.χ. με το Co-Pilot) μέσα από το περιβάλλον προγραμματισμού. Θα πρέπει να γνωρίζετε ακριβώς

τι κάνει ο κώδικάς σας και να είστε σε θέση να παρέμβετε σε αυτόν, αν σας ζητηθεί (βλ. αξιολόγηση). Επιτρέπεται η χρήση της ΤΝ για περιφερειακές εργασίες, όπως η σύνθεση μουσικού χαλιού, η παραγωγή φωνητικών διαλόγων ή η παραγωγή εικαστικών. Ειδικά για το τελευταίο, μπορείτε να δείτε διάφορες πηγές, όπως αυτή:

https://www.youtube.com/watch?v=Gb6zy40WojA

Εικαστικά και Ήχος. Δεν υπάρχει περιορισμός στη χρήση εικόνων, εικονιδίων. Μπορείτε να σχεδιάσετε τα δικά σας, να «δανειστείτε» assets από γνωστά παιχνίδια, να παράγετε assets με τεχνητή νοημοσύνη ή να κατεβάσετε πακέτα με sprites από το διαδίκτυο. Η αισθητική του παιχνιδιού δεν αποτελεί βαθμολογικό κριτήριο. Απλά σας δίνει ένα (μεγάλο) δημιουργικό περιθώριο που με κατάλληλο προγραμματισμό μπορεί να κάνει το παιχνίδι σας πολύ εντυπωσιακό.

Αναφορές

Αν θέλετε να ανατρέξετε σε παλιά αλλά και σχετικά πρόσφατα παιχνίδια για να εμπνευστείτε ή να καταλάβετε το ύφος του ζητούμενου παιχνιδιού, ακολουθεί μια ενδεικτική λίστα με λίγα από τα αρκετά πετυχημένα παιχνίδια του είδους.

Machinarium	https://www.youtube.com/watch?v=4xb3IImki-4
The Secret of Monkey Island	https://www.youtube.com/watch?v=U6Pzmv7qZ5o
Monkey Island 2	https://www.youtube.com/watch?v=iB-bCN1g0rI
Broken sword	https://www.youtube.com/watch?v=ZjkNnvdTcBI
The Inner World	https://www.youtube.com/watch?v=yRRgnoqlE6o
Day of the Tentacle	https://www.youtube.com/watch?v=he_DDsOnqI0
Indiana Jones and the Fate of Atlantis	https://www.youtube.com/watch?v=M2v2mj0cmg0
The Loom	https://www.youtube.com/watch?v=7o2eUJLEWA8
Legend of Kyrandia – Book 2	https://www.youtube.com/watch?v=Dl5tGVS2PL8

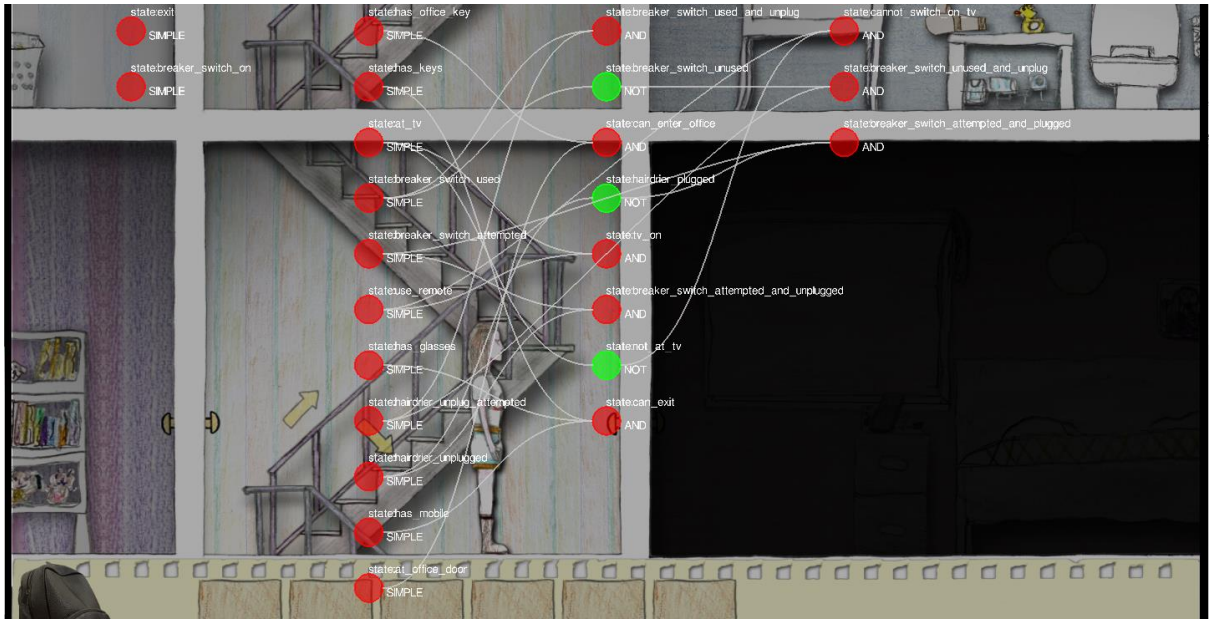
Για Προχωρημένους

Σε ένα πολύπλοκο παιχνίδι ή όταν θέλουμε να ενεργοποιήσουμε καταστάσεις ή να παίξουμε κάποια ακολουθία από animations, ήχους κλπ. Συχνά καταφεύγουμε στη χρήση ενός (κεντρικού) συστήματος από γεγονότα (events). Τυπικά, όλα τα αντικείμενα του παιχνιδιού έχουν τη δυνατότητα να γεννήσουν ένα event και να το αποθέσουν προς εκτέλεση στην «ουρά μηνυμάτων» (βασικά λίστα είναι…) που τη φιλοξενεί και ελέγχει η κεντρική κλάση του παιχνιδιού. Τα γεγονότα συχνά συνοδεύονται από 2 χρόνους: την καθυστέρηση ενεργοποίησής τους και τη διάρκειά τους. Η μηχανή του παιχνιδιού σε κάθε κύκλο εκτέλεσης (update function) διατρέχει τη λίστα των

γεγονότων και για όσα έχει έρθει ο χρόνος ενεργοποίησής τους τα θέτει σε ενεργή κατάσταση. Εσωτερικά τα ενεργοποιημένα events (καλώντας η μηχανή τη δική τους update) εξετάζουν αν έχουν λήξει και όσο λειτουργούν, εκτελούν την ενέργεια για την οποία κατασκευάστηκαν. Η μηχανή είναι υπεύθυνη να «καθαρίσει» όσα events έχουν τερματίσει. Ένα event τυπικά εξειδικεύεται ως κλάση για να εκτελεί συγκεκριμένες λειτουργίες, όπως AnimationEvent, FaderEvent, LocationTransitionEvent, SoundEvent, StateEnableEvent, κλπ. Όπως εύκολα είναι αντιληπτό και από τα ονόματα των παραπάνω κλάσεων, κατά τη δημιουργία τους και πριν σταλούν στην ουρά γεγονότων, ενδέχεται να τους περάσουμε κάποιο δείκτη ή αναφορά σε άλλα αντικείμενα του παιχνιδιού για να μπορέσουν τα γεγονότα να αλληλεπιδράσουν μαζί τους. Π.χ.: ένα FaderEvent, ενδέχεται να αλλάζει τη διαφάνεια κάποιου στοιχείου του παιχνιδιού το οποίο πρέπει ρητά να περαστεί κατά την κατασκευή του event.

Ένα δεύτερο στοιχείο που είναι σημαντικό για τα γεγονότα ως κλάσεις είναι ότι πάντα πρέπει να διαθέτουν τουλάχιστον ένα πεδίο με στιγμιότυπο μιας κλάσης «μετρητή» χρόνου (Timer), ώστε να μπορεί κάθε event να παρακολουθεί ανεξάρτητα το χρόνο που αυτό είναι ενεργό και να μπορεί να τερματίσει όταν υπερβεί το χρονικό όριο. Αρκετές φορές η λειτουργία του εξαρτάται επίσης από το χρόνο αυτό (π.χ. το FaderEvent παραπάνω ή κάποιο event Που παίζει κυκλικά κάποιο animation). Τα events έχουν και αρκετά μεγάλη χρησιμότητα όταν αλληλεπιδρούμε με triggers και καταστάσεις του παιχνιδιού. Η ενεργοποίηση μιας κατάστασης στο παιχνίδι μας συχνά οδηγεί στην ανάγκη να ξεκινήσουν διάφορα γεγονότα. Για παράδειγμα, όταν ανοίξουμε μια πόρτα θα πρέπει να αλλάξουμε μια τοποθεσία ή να μετακινήσουμε τον ήρωά μας σε άλλο χώρο. Ταυτόχρονα μπορεί να θέλουμε να παίξουμε κάποιον ήχο και να ενεργοποιήσουμε ένα sprite animation που δείχνει την πόρτα να ανοίγει. Ένα γεγονός επίσης μπορεί (με την ολοκλήρωσή του) να θέτει κάποια κατάσταση ως ενεργή η ανενεργή.

Όπως αναφέρθηκε και παραπάνω, ο έλεγχος της ροής του παιχνιδιού μπορεί να γίνεται μέσω ενός γράφου καταστάσεων και συνδυαστικών λογικών πυλών. Η μοντελοποίηση της πλοκής με έναν τέτοιο τρόπο βοηθάει στο να αποσυσχετιστεί το επίπεδο της διάδρασης από το επίπεδο της παρουσίασης και το επίπεδο της λειτουργικότητας σημαντικά και να μπορούμε να αλλάζουμε και παραμετροποιούμε τα αντικείμενα του παιχνιδιού σε πιο υψηλό επίπεδο. Αυτό μας δίνει επιπλέον τη δυνατότητα να δημιουργήσουμε πιο σύνθετες ροές ελέγχου και πιο ενδιαφέροντα σενάρια.



Γράφος καταστάσεων του demo.

SGG βιβλιοθήκη

Η βιβλιοθήκη της εργασίας βρίσκεται στο εξής repository: https://github.com/cgaueb/sgg. Μπορείτε να βρείτε αναλυτικές οδηγίες εγκατάστασης της βιβλιοθήκης εδώ https://cgaueb.github.io/sgg/pages.html. Μπορείτε να βρείτε επίσης video στο eclass του μαθήματος, στην κατηγορία σύνδεσμοι, σχετικά με την εγκατάσταση της βιβλιοθήκης **2023 - Χρήση της βιβλιοθήκης SGG - Μέρος Α.**

Ομάδες

Η εκπόνηση της εργασίας γίνεται σε ομάδες μέχρι 2 ατόμων. Σε εξαιρετικές περιπτώσεις όπου μια ομάδα 3 μελών επιθυμεί να κατεβάσει μια ενιαία εργασία, η έκταση, το θέμα και τα χαρακτηριστικά της θα πρέπει να πάρει την έγκριση του διδάσκοντος και σε κάθε περίπτωση θα πρέπει να είναι αναλογικά πολυπλοκότερη η υλοποίησή της σε σχέση με τις ελάχιστες συμβατικές προδιαγραφές. Τονίζεται ότι επιπρόσθετος φόρτος που μπορεί να απαιτηθεί για τη δημιουργία καλύτερων εικαστικών ή ήχου δεν είναι υπολογίσιμος ως προς τη δυνατότητα επέκτασης της ομάδας σε 3 μέλη, καθώς στο μάθημα εξετάζεται η εργασία από πλευράς κώδικα, όχι δημιουργικού σκέλους.

Bonus

Η εργασία δύναται να λάβει επιπρόσθετη βαθμολογία (+0.5) εφόσον διαπιστωθεί σημαντική προσπάθεια για ορθό, επεκτάσιμο και αποτελεσματικό σχεδιασμό κώδικα, υλοποίηση επιπρόσθετων χαρακτηριστικών και λειτουργιών ή/και ενσωμάτωση μηχανισμών που αναφέρονται στην ενότητα «για προχωρημένους». Τυπικά μια ομάδα 3 ατόμων θα πρέπει να ενσωματώσει αυτές τις βελτιώσεις ως κύριες προδιαγραφές της εργασίας (χωρίς να πάρει το bonus).

Παράδοση Εργασιών

Η εργασία σας θα πρέπει να ανέβει στο eclass από ένα από τα μέλη της ομάδας σαν ένα zip αρχείο που θα πρέπει να περιλαμβάνει:

- Τον κώδικα της εργασίας. Προσοχή, από τη βιβλιοθήκη SGG να έχετε μόνο τα 2 απαραίτητα header files για τη μεταγλώττιση του δικού σας προγράμματος (graphics.h, scancodes.h), όχι όλο τον κώδικα της βιβλιοθήκης!
- Τα assets που χρησιμοποιεί η εφαρμογή σας στους κατάλληλους φακέλους έτσι ώστε το εκτελέσιμο που χτίζεται να μπορεί να τα βρει κατά την εκτέλεσή του. Προσοχή: αν χρησιμοποιείτε πολλά ή/και μεγάλα αρχεία εικόνας ή ήχου και αυξάνει πολύ το μέγεθος του zip σας (>4MB), προτιμήστε να ανεβάσετε χωριστά το φάκελο των assets σε κάποιον εξωτερικό σύνδεσμο (που να μη λήγει – όχι π.χ. WeTransfer) και στο zip σας βάλτε ένα txt αρχείο που να περιγράφει α) τη διεύθυνση του εξωτερικού συνδέσμου, β) τον σχετικό κατάλογο ως προς το εκτελέσιμο που περιμένει η εφαρμογή να βρει τα δεδομένα αυτά. Π.χ. αν το εκτελέσιμο βρίσκεται στο φάκελο D:\game\bin\ και ψάχνει τα αρχεία δεδομένων στον κατάλογο D:\game\bin\assets, βάλτε στο txt αρχείο τον σχετικό κατάλογο .\assets.

- Τα απαραίτητα αρχεία για το χτίσιμο του κώδικα της εργασίας, π.χ. το solution (.sln) και Project file (.vcxproj) στους κατάλληλους υποφακέλους, αν χρειάζεται, ή κάποιο makefile ή κάποιο build script.

Προσοχή: Επιπλέον, από φέτος είναι υποχρεωτικό να δημιουργήσετε ένα σύντομο video περίπου 5 λεπτών όπου δείχνετε και εξηγείτε προφορικά (ή γραπτά με λεζάντες αν δε μπορείτε διαφορετικά):

α) στιγμές από τη λειτουργία του παιχνιδιού σας (όχι κατ' ανάγκη πλήρες play through) και

β) Τα βασικά και σημαντικά στοιχεία του κώδικά σας (βασική ιεραρχία κλάσεων, σημαντικά κομμάτια που θέλετε να επιδείξετε / τονίσετε / καμαρώσετε γι αυτά).

Το video μπορεί να είναι στα Ελληνικά ή στα Αγγλικά. Ανεβάστε το στο YouTube ως ιδιωτικό video και γράψτε το URL στα σχόλια της εργασίας στο eclass. Για τη δημιουργία του μπορείτε να χρησιμοποιήσετε οποιοδήποτε πρόγραμμα καταγραφής οθόνης, όπως το δημοφιλές OBS. Δεν είναι απαραίτητο να προβείτε σε επεξεργασία και επιμέλεια του video, καθώς δεν απαιτούνται τέτοιες δεξιότητες.

Η δημιουργία του video είναι υποχρεωτικό μέρος της αξιολόγησης.

Μην ανεβάσετε:

- Τα αρχεία των βιβλιοθηκών δυναμικής σύνδεσης (DLLs. SOs).
- Τα .lib αρχεία της SGG. Τα έχουμε κι εμείς!
- Τα προσωρινά αρχεία που δημιουργούνται κατά το χτίσιμο της εφαρμογής και τα οποία ενδέχεται (ειδικά για την περίπτωση του visual studio) να είναι αρκετά μεγάλα. Τέτοια είναι τα *.pdb, *.tmp, *.obj, *.ilk. Προσοχή: κάποια από αυτά είναι κρυφά, όπως ο κατάλογος .vs.

Στο όνομα του αρχείου zip πρέπει να περιλαμβάνονται οι αριθμοί μητρώου **όλων των μελών** της ομάδας.

Η καταληκτική ημερομηνία και ώρα παράδοσης της εργασίας είναι 17/01/2026, 23:55. Δε θα δοθεί καμία παράταση, καθώς η εξέταση της εργασίας θα ξεκινήσει αμέσως μετά σε χρονοθυρίδες μέσα στο διάστημα της εξεταστικής Ιανουαρίου – Φεβρουαρίου, τις οποίες θα δηλώσετε (θα σταλεί σχετικό μήνυμα).

Εξέταση και Βαθμολόγηση Εργασιών

Η εξέταση (και βαθμολόγηση) των εργασιών θα γίνει ατομικά στα μέλη των ομάδων, μέσω MS Teams σε διαφορετική για τον καθένα συνεδρία. Θα καταρτιστεί κατάλογος με τα μέλη που συμμετείχαν στις υποβληθείσες εργασίες και από την εβδομάδα που θα ακολουθήσει την καταληκτική ημερομηνία παράδοσης της εργασίας, θα σας καλέσουν οι βοηθοί του μαθήματος για εξέταση.

Κατά την εξέταση των εργασιών, διάρκειας περίπου 10 λεπτών, θα σας ζητηθεί να εξηγήσετε συγκεκριμένα μέρη του κώδικά σας και να παρέμβετε σε αυτόν για να παραλλάξετε τη λειτουργία του.

Η βαθμολόγηση των εργασιών, με άριστα το 4, θα γίνει σύμφωνα με τα ακόλουθα κριτήρια:

Κριτήριο	**Επίπτωση**	**Σχόλια**
Μη χρήση της βιβλιοθήκης SGG	Απόρριψη εργασίας	Η αποκλειστική χρήση της βιβλιοθήκης είναι υποχρεωτική για τη διαχείριση παραθύρου, γραφικών και εισόδου. Επιτρέπονται μόνο βοηθητικές βιβλιοθήκες (π.χ. `json.hpp`, `SDL_Image`), όχι άλλες βασικές όπως απλή SDL ή OpenGL.
Παράδοση εργασίας που δεν εμπίπτει θεματολογικά στην εκφώνηση	Απόρριψη εργασίας	Πρέπει να είναι point-and-click adventure game με τουλάχιστον 3 τοποθεσίες και αντίστοιχη λογική γρίφων/αντικειμένων (αν δεν είστε σίγουροι, ρωτήστε τον διδάσκοντα).
Μη χρήση κληρονομικότητας	Έως -1,0	Απαιτείται κατ' ελάχιστο σωστή ιεραρχία κλάσεων με βασική αφηρημένη κλάση (π.χ. `GameObject`) και παράγωγες κλάσεις για αντικείμενα, τοποθεσίες, γεγονότα κ.λπ.
Μη πολυμορφική κλήση κοινών μεθόδων	Έως -1,0	Η κεντρική μηχανή πρέπει να καλεί πολυμορφικά μεθόδους (π.χ. `update()`, `draw()`, ή άλλα events τα οποία ανήκουν σε κάποια STL συλλογή και αφορούν πολλαπλά αντικείμενα) μέσω δεικτών/αναφορών βασικής κλάσης χωρίς να εξετάζει τον ακριβή τύπο.
Μη χρήση δομών δεδομένων	Έως -0,5	Υποχρεωτική χρήση τουλάχιστον μίας δομής της STL (π.χ. `std::vector`, `std::list` για ουρές γεγονότων/αντικείμενα, ή `std::map`/`std::unordered_map` για ευρετήρια).
Μη ικανοποιητικός σχεδιασμός εφαρμογής	Έως -0,5	Αφορά την αρχιτεκτονική του κώδικα π.χ. (και όχι μόνο) καθαρότητα, αποφυγή «spaghetti» κώδικα, σωστός διαχωρισμός αρμοδιοτήτων, αποφυγή περιττών καθολικών μεταβλητών και συναρτήσεων, σωστός διαχωρισμός .cpp/.h αρχείων.

Μη λειτουργική διεπαφή με το χρήστη / έλεγχος ροής παιχνιδιού	Έως -0,5	Σωστή απόκριση σε clicks/ποντίκι, διαχείριση καταστάσεων (state machine/inventory), αλλαγή σκηνών και ομαλή ροή χωρίς bugs που «μπλοκάρουν» τον παίκτη.
Υλοποίηση επιθυμητών (μη υποχρεωτικών) χαρακτηριστικών	Έως +0,5	Ενσωμάτωση προχωρημένων μηχανισμών (π.χ. κεντρικό event system, γράφος λογικών πυλών/καταστάσεων, timers). Σημείωση: Για ομάδες 3 ατόμων αυτά αποτελούν υποχρέωση και δεν δίνουν bonus.
Μη παράδοση video	Έως -0,5	Υποχρεωτικό επεξηγηματικό video διάρκειας περίπου 5 λεπτών με σύντομο gameplay και επίδειξη βασικών σημείων κώδικα.
Αδυναμία απάντησης σε ερωτήσεις επί του κώδικα ή εκτέλεσης παρεμβάσεων σε αυτόν.	Έως -4,0	Κατά την ατομική προφορική εξέταση (~10 λεπτά στο MS Teams), κάθε μέλος πρέπει να μπορεί να εξηγήσει και να τροποποιήσει άμεσα τον κώδικα που παραδόθηκε.

Σημειώνεται ότι, με βάση τα παραπάνω, μια άρτια εργασία 2 ατόμων που έχει υλοποιημένα επιπρόσθετα χαρακτηριστικά από τα υποχρεωτικά, δύναται να υπερβεί σε βαθμό το 4.