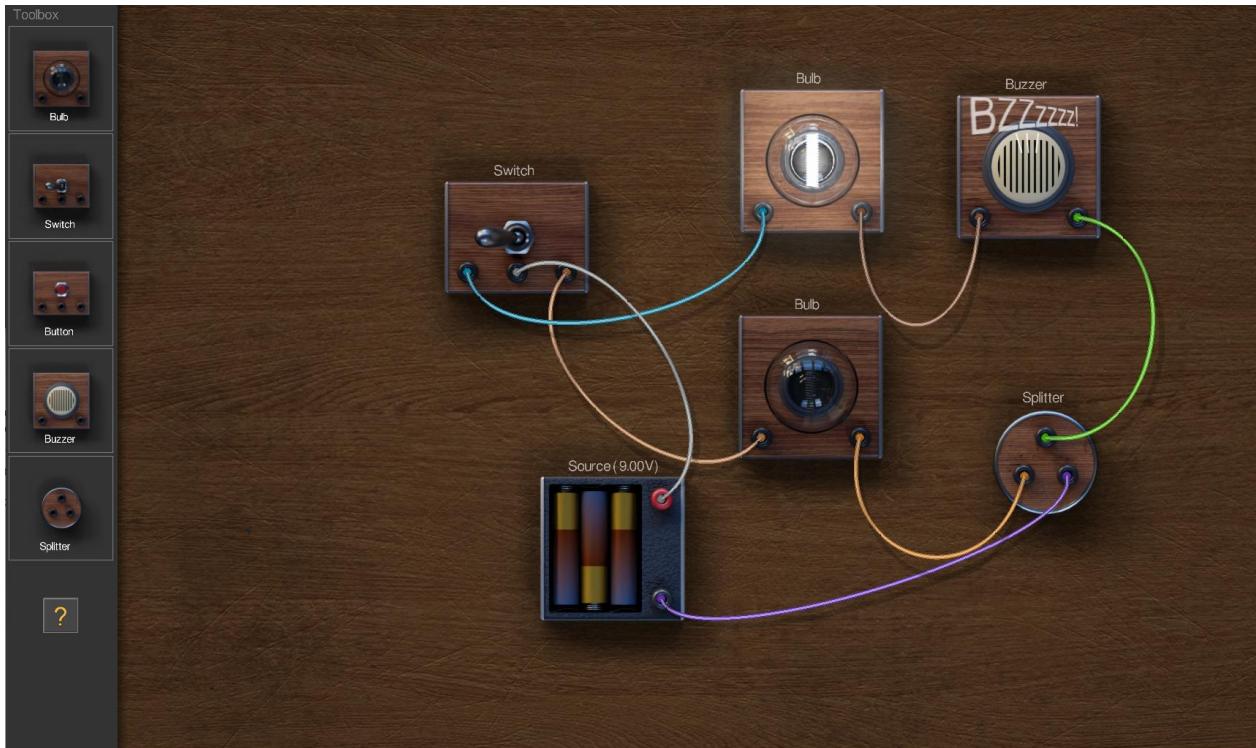


Εργασία στον Προγραμματισμό Υπολογιστών με C++

Ακαδ. Έτος 2025-26

Διαδραστική Εφαρμογή με Χρήση Δικτύου Κόμβων

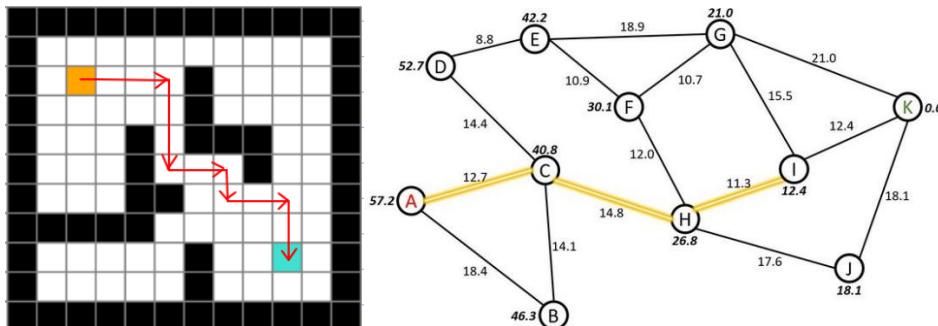


1. Σύντομη Περιγραφή

Ο στόχος της εργασίας είναι να δημιουργήσετε μια δική σας διαδραστική εφαρμογή η οποία βασίζεται σε λειτουργικότητα που εμπεριέχει την έννοια ενός δικτυώματος (γράφου) από κόμβους. Η εφαρμογή μπορεί να είναι είτε κάποιο παιχνίδι (υπαρκτό ή της δικής σας έμπνευσης) είτε κάποια απλή προσομοίωση. Η υλοποίησή σας είναι υποχρεωτικό να έχει γραφική έξοδο και κάποια αλληλεπίδραση με το χρήστη μέσω της βιβλιοθήκης [Simple Graphics Library](#) (SGG) που έχει φτιαχτεί για το μάθημα.

2. Πιθανές ιδέες

Δρομολόγηση σε χάρτη ή grid με εμπόδια – A-star algorithm (βαθμός δυσκολίας: μέτριος)



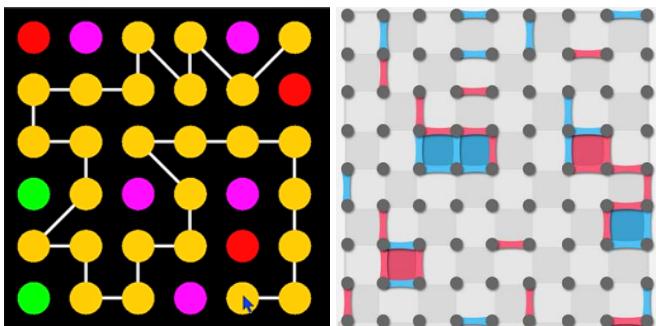
Προσομοίωση δικτύου υδάτων και φόρτου σε σωλήνες (βαθμός δυσκολίας: μέτριο)



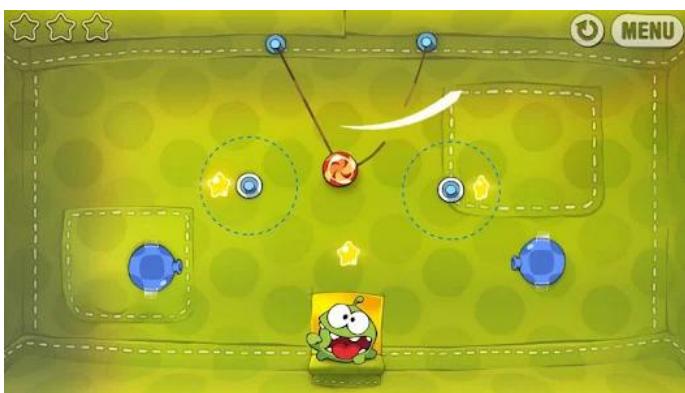
Ένα παιχνίδι τύπου “Pipe-mania” (βαθμός δυσκολίας: μέτριος)



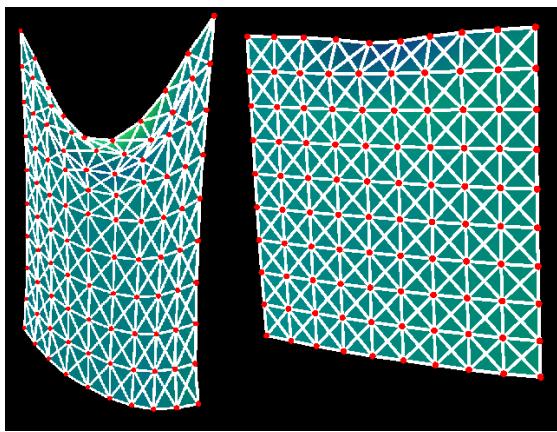
Ένα παιχνίδι τύπου “connect the dots” ή “dots and boxes” (βαθμός δυσκολίας: μέτριος)



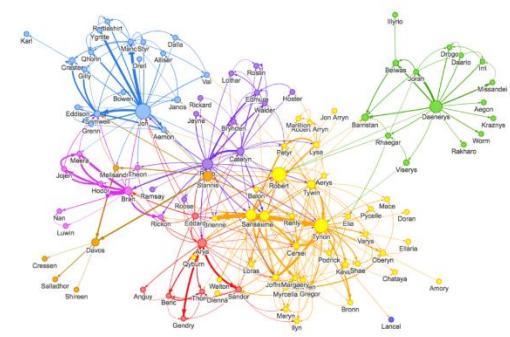
Ένα παιχνίδι τύπου “cut the rope”(βαθμός δυσκολίας: αυξημένος)



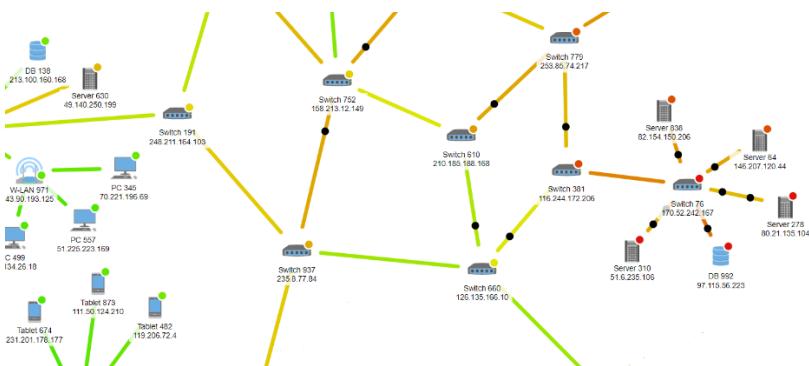
Διαδραστικό 2D Cloth simulation, ενδεχομένως με υποστήριξη για σκίσιμο (βαθμός δυσκολίας: αυξημένος).



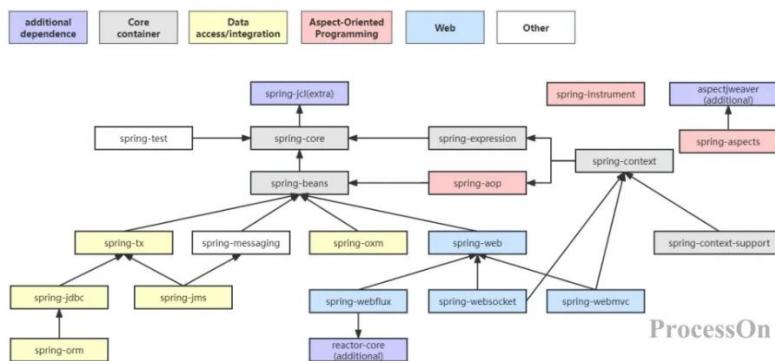
Social network visualization (βαθμός δυσκολίας: μέτριος)



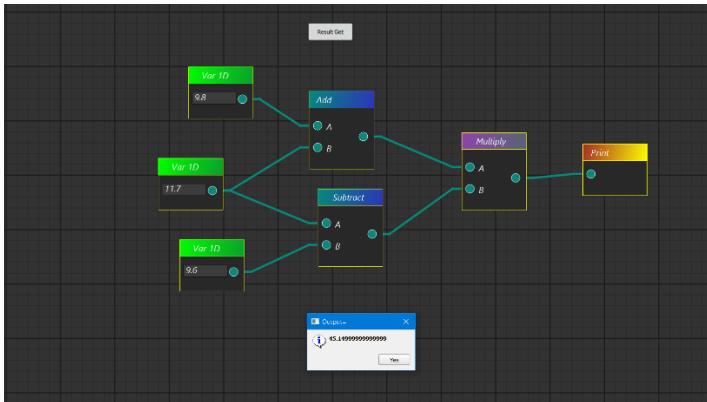
Προσομοίωση δρομολόγησης κίνησης δικτύου (βαθμός δυσκολίας: αυξημένος)



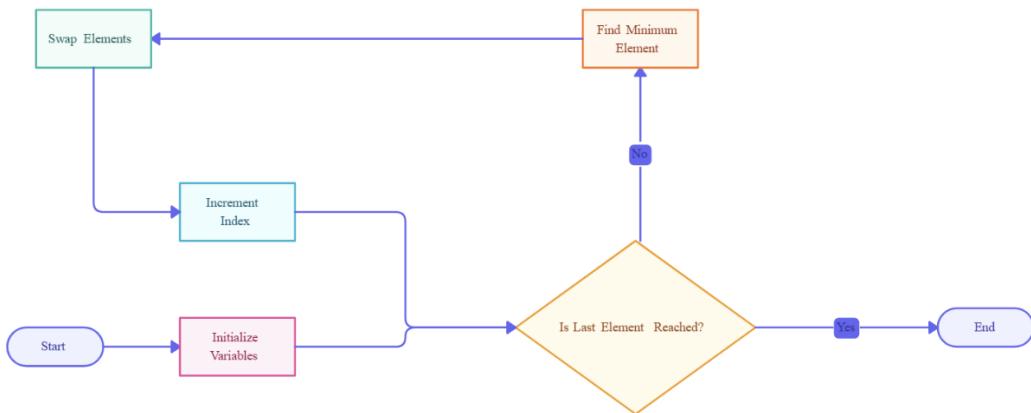
Οπτικοποίηση γράφου εξαρτήσεων (βαθμός δυσκολίας: μέτριος)



Εκπαιδευτική εφαρμογή visual coding



Εφαρμογή οπτικοποίησης / visual debugging ροής προγράμματος



Θυμηθείτε, η δημιουργικότητα ανταμείβεται. Σκεφτείτε δικές σας εφαρμογές!

3. Υλοποίηση

Κατά την υλοποίηση της εφαρμογής σας, καλείστε να συνδυάσετε γνώσεις που αποκομίσατε από τις διαλέξεις και να σκεφτείτε καλά την αρχιτεκτονική του κώδικά σας, προκειμένου να πετύχετε α) καλή επαναχρησιμοποίηση κώδικα, β) ενιαίο και πολυμορφικό τρόπο κλήσης μεθόδων, δ) αποδοτική εκμετάλλευση έτοιμων δομών της STL, γ) ταχύτητα.

Οι παρακάτω στόχοι είναι υποχρεωτικοί και βαθμολογούνται:

- **Χρήση της βιβλιοθήκης SGG.** Η ενσωμάτωση της βιβλιοθήκης SGG και χρήση των συναρτήσεων που παρέχονται είναι υποχρεωτική και αποκλειστική. Η εφαρμογή σας δε θα πρέπει να βασίζεται σε άλλη εξωτερική βιβλιοθήκη για τη διαχείριση του παραθύρου και των συμβάντων πληκτρολογίου και ποντικιού, τη σχεδίαση γραφικών και την αναπαραγωγή ήχου. Μπορείτε μόνο να χρησιμοποιήσετε άλλες βιβλιοθήκες για επιπρόσθετες και μη επικαλυπτόμενες με την SGG λειτουργίες, όπως επικοινωνία πάνω από δίκτυο, διάβασμα/εγγραφή XML/Json αρχείων, κλπ.
- **Χρήση δυναμικής μνήμης.** Στις διαδραστικές εφαρμογές και κυρίως σε παιχνίδια και προσομοιώσεις, πολλές οντότητες δημιουργούνται και «ζουν» για ένα περιορισμένο διάστημα κατά την εκτέλεση του κώδικα. Τέτοια στοιχεία πρέπει να δημιουργούνται δυναμικά στη μνήμη (με new) και να καταστρέφονται όταν δε χρειάζονται.
- **Κληρονομικότητα και πολυμορφισμός.** Τα διάφορα στοιχεία της εφαρμογής έχουν μια εσωτερική οντολογική ιεραρχική δομή. Για παράδειγμα, ενδεικτικά, οτιδήποτε εμφανίζεται επί της οθόνης

μπορεί να είναι ένα VisualAsset ή οτιδήποτε έχει συμπεριφορά «κόμβου» σε ένα δίκτυο/γράφο είναι απόγονος μιας class Node. Αν έχετε user interface, κι εκεί μπορείτε να εφαρμόσετε πολυμορφισμό, έχοντας για παράδειγμα μια γενική κλάση Widget που την εξειδικεύετε για να φτιάξετε λειτουργικά πιο συγκεκριμένα στοιχεία, όπως Button, Slider, κλπ.

- **Συλλογές.** Σε μια εφαρμογή σαν κι αυτή, αποθηκεύουμε και διαχειριζόμαστε μια πολλαπλότητα από αντικείμενα, είτε για τη λειτουργία του προγράμματος, είτε για τη σχεδίαση των γραφικών στην οθόνη. Καλείστε να χρησιμοποιήσετε τις καταλληλότερες για τη δουλειά που τις χρειάζεστε συλλογές της STL για τις ανάγκες αποθήκευσης, αναζήτησης και μαζικής εκτέλεσης μεθόδων. Προσοχή: για να δουλέψουν ορισμένες από τις παρεχόμενες συλλογές σωστά με δικές σας κλάσεις, θα πρέπει να προσδιορίσετε τους κατάλληλους τελεστές για την ταξινόμηση ή το hashing των αντικειμένων (βλ. διαφάνειες μαθήματος). Συστήνεται αυστηρά να μην υλοποιήσετε δικές σας συλλογές για πράγματα που ήδη σας παρέχει η STL (π.χ. μην υλοποιήσετε δικές σας λίστες).
- Θα πρέπει να έχετε στην εφαρμογή σας ένα και μοναδικό στιγμιότυπο μιας κλάσης **GlobalState** η οποία:
 - α) φυλάει δεδομένα για τη ροή και την κατάσταση του παιχνιδιού/εφαρμογής (π.χ. σε πιο level είμαστε, τα διαθέσιμα levels, στατιστικά, σκορ, τα raw δεδομένα μιας προσομοίωσης, κλπ.)
 - β) διαθέτει τις βασικές μεθόδους draw, init και update που καλούν τις αντίστοιχες έμμεσα ή άμεσα για οποιοδήποτε αντικείμενο λειτουργεί μέσα στη διαδραστική εφαρμογή.
 - δ) παρέχει ένα κεντρικό σημείο πρόσβασης για να εντοπιστούν στοιχεία και καθολικά δεδομένα της εφαρμογής (π.χ. getPlayer(), getGUI(), getWindowSize(), getCanvasSize(), κλπ).

4. Προαιρετικά Χαρακτηριστικά

Θα εκτιμηθεί θετικά ο σωστός σχεδιασμός και δόμηση του κώδικα, η σχολαστική δήλωση μεθόδων (π.χ σωστή χρήση αναφορών και const ορισμάτων ή μεθόδων), η εκμετάλλευση templated συναρτήσεων ή κλάσεων, όπου φαίνεται χρήσιμο. Υπενθυμίζεται ότι ορισμένα μονοπάτια κώδικα που εκτελούν ενδεχομένως βαριές διαδικασίες υπολογισμών μπορούν να εκτελεστούν σε ξεχωριστό(ά) thread(s)¹.

5. Συγκρούσεις

Αναπόφευκτα, όταν έχετε οντότητες μέσα σε ένα παιχνίδι, οπτικοποιούμενους κόμβους ενός γράφου ή αλληλεπίδραση με στοιχεία μιας διεπαφής, πρέπει να ελέγχετε συγκρούσεις (collisions) μεταξύ ορισμένων από τα στοιχεία αυτά ή την υπέρθεση ενός σημείου με μια περιοχή. Οι λόγοι που χρειάζεται αυτή η λειτουργικότητα είναι πολλοί:

- Έλεγχος highlighting/selection ενός στοιχείου από τον κέρσορα του χρήστη.
- Αποφυγή επικάλυψης οπτικών στοιχείων πάνω στον καμβά. Π.χ. να ανιχνεύουμε πότε (και πόσο) 2 κόμβοι ενός γράφου επικαλύπτονται ώστε να τους απωθούμε/τοποθετούμε σε διακριτές θέσεις.
- Ανίχνευση συγκρούσεων μεταξύ οντοτήτων ενός παιχνιδιού.

¹ ΜΗΝ το κάνετε για τη σχεδίαση ή οτιδήποτε έχει να κάνει με γραφικά και context παραθύρου, αυτά πρέπει να καλούνται από το κύριο thread της εφαρμογής. Διαχείριση αντικειμένων του παιχνιδιού, ανίχνευση συγκρούσεων, συμπεριφορά εχθρών και άλλες λειτουργίες μπορούν να γίνουν παράλληλα σε άλλα νήματα.

5.1. Έλεγχος περίληψης σημείου σε παραλληλόγραμμο

Ένας κοινός έλεγχος που είναι απαραίτητος κυρίως σε γραφικά στοιχεία μιας διεπαφής είναι το να μάθουμε εάν ένα σημείο (συνήθως ο κέρσορας) βρίσκεται μέσα σε ένα κουτί που παριστάνεται όρια του γραφικού στοιχείου. Έστω ότι το γραφικό στοιχείο έχει κέντρο (c_x, c_y) διαστάσεις (w, h) , τότε ένα σημείο με συντεταγμένες (x, y) βρίσκεται εντός του πρώτου όταν:

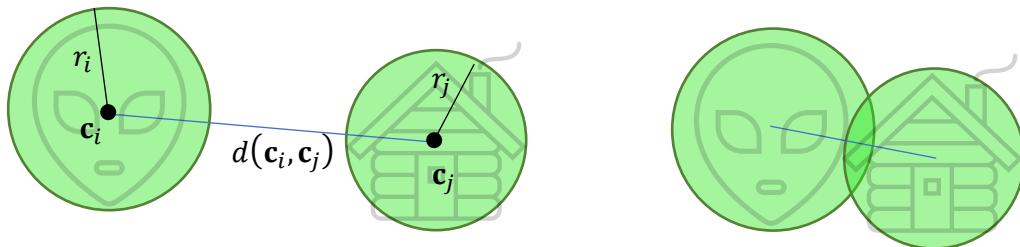
$$c_x - \frac{w}{2} \leq x \leq c_x + \frac{w}{2} \quad \text{AND} \quad c_y - \frac{h}{2} \leq y \leq c_y + \frac{h}{2}$$

5.2. Έλεγχος περίληψης σημείου σε κύκλο

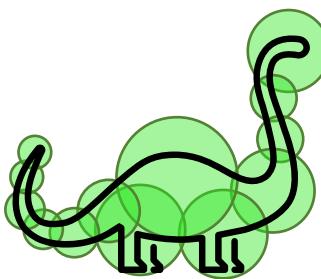
Ένα σημείο $\mathbf{p} = (x, y)$ βρίσκεται στην “εμβέλεια” ακτίνας r ενός άλλου σημείου $\mathbf{c} = (c_x, c_y)$, όταν: $d(\mathbf{p}, \mathbf{c}) \leq r$, όπου $d(\cdot, \cdot)$ η Ευκλείδεια απόσταση.

5.3. Έλεγχος επικάλυψης δίσκων

Αν έχω ένα αντικείμενο με κέντρο \mathbf{c}_i που μπορώ να προσεγγίσω με ένα αποτύπωμα σε σχήμα δίσκου με ακτίνα r_i για τις ανάγκες ενός ελέγχου επικάλυψης ή σύγκρουσης με άλλα αντικείμενα με αντίστοιχο κέντρο \mathbf{c}_j και ακτίνα r_j , τότε αυτά επικαλύπτονται στο επίπεδο αν και μόνο αν: $d(\mathbf{c}_i, \mathbf{c}_j) - r_i - r_j < 0$.



Αν έχω κάποια αρκετά σύνθετα σχήματα, μπορώ να ορίσω πολλαπλούς κύκλους σύγκρουσης για αυτά, έτσι ώστε μικραίνοντας τις ακτίνες αυτών, να περιγράψω καλύτερα τα όρια του αντικειμένου:



Ένα αντικείμενο που περιλαμβάνει πολλαπλές περιοχές (κύκλους εδώ) σύγκρουσης, συγκρούεται με κάποιο άλλο, αν έστω και μια περιοχή του βρει σύγκρουση με κάποια αντίστοιχη περιοχή σύγκρουσης του άλλου αντικειμένου.

6. Ομάδες

Οι εργασία παραδίδεται από ομάδες 1-2 ατόμων. Ο σχηματισμός ομάδων 3 ατόμων δεν ενθαρρύνεται. Σε περίπτωση που μια ομάδα επιλέξει να υλοποιήσειένα αρκετά πιο σύνθετο project (μετά από συνεννόηση με το διδάσκοντα), τότε μπορεί να επεκταθεί μέχρι τα 3 μέλη. Η πολυπλοκότητα της εφαρμογής θα πρέπει να συνεπάγεται και αντίστοιχη πολυπλοκότητα στο σχεδιασμό του κώδικα και την υλοποίηση. Δεν εμποδίζει

τίποτα προφανώς μικρότερες ομάδες να αναλάβουν και να παραδώσουν μια πιο σύνθετη εργασία (με την ανάλογη επιβράβευση).

Σε κάθε περίπτωση, όλα τα μέλη της ομάδας θα πρέπει να έχουν ασχοληθεί με κάποια λειτουργικότητα της εφαρμογής και να έχουν συντελέσει στη συγγραφή του κώδικα. Δηλαδή δεν επιτρέπεται να επιμεριστεί ο φόρτος μεταξύ των μελών ώστε κάποιος να αναλάβει μόνο τα εικαστικά ή τους ήχους.

7. Οπτικοακουστικό Υλικό

Οι εφαρμογές σας μπορούν να υλοποιηθούν και με πολύ βασικά γραφικά και τις σχεδιαστικές δυνατότητες της SGG, οπότε είτε κατά τα αρχικά στάδια της υλοποίησής σας (που κυρίως ελέγχετε λειτουργικότητα) είτε αν δεν έχετε δυνατότητα ή χρόνο να ασχοληθείτε με την «παρουσίαση» του περιεχομένου, μπορείτε να χρησιμοποιήσετε τα έτοιμα primitives σχεδίασης που σας παρέχει η SGG για να δείξετε απλές μορφές και σχήματα στην οθόνη. Δε παίρνετε αρνητική βαθμολογία για την αισθητική της εφαρμογής.

Αν πάλι θέλετε να βάλετε δικά σας στοιχεία ή έτοιμα bitmaps που κατεβάσατε από το διαδίκτυο, ο μορφότυπος PNG για τη φόρτωση και σχεδίαση εικόνων πάνω στα βασικά primitives είναι υπερ-αρκετός για τις ανάγκες σας, αφού υποστηρίζει και κανάλι διαφάνειας και όλα τα δημοφιλή και ελεύθερα προγράμματα επεξεργασίας και μετατροπής εικόνων το υποστηρίζουν. Τα αρχεία σας φέρτε τα στο κατάλληλο μέγεθος που να είναι συμβατό με τις ανάγκες της εφαρμογής. Για παράδειγμα, αν θέλετε να φορτώσετε ως background μια εικόνα που βρήκατε ανάλυσης 4400X3300 pixels, αυτή θα είναι πολύ μεγάλη, ξοδεύοντας ασκοπα α) μνήμη, β) χρόνο ανοίγματος (ειδικά σε debug mode), γ) χώρο στο δίσκο και στο zip που θα φτιάξετε στο τέλος (βλ. παράδοση εργασιών). Με κάποιο πρόγραμμα επεξεργασίας εικόνων, φέρτε τη σε διαστάσεις κατάλληλες για το παράθυρό σας. Π.χ. για ένα 1024X768 παράθυρο, στο παράδειγμά μας καλή θα είναι να μετατροπή της εικόνας σε 1067X800, δηλαδή να υπερκαλύπτει την αναμενόμενη ανάλυση παραθύρου, διατηρώντας το λόγιο πλάτους ύψους της αρχικής εικόνας. Σημείωση: αν έχετε φέρει τις εικόνες σας σε διαστάσεις που να είναι δυνάμεις του 2 (π.χ. 1024X512, 64X64), τότε η φόρτωσή τους από τη βιβλιοθήκη είναι γρηγορότερη, καθώς διαφορετικά πρέπει να τις μετατρέψει στις πλησιέστερες δυνάμεις του 2 η συνάρτηση φορτώματος.

8. Παράδοση Εργασιών

Οι 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 μεταγώπτιση 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 βρειτα 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.

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 χτίσμα 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 16/1/2026, 23:55. Δε θα δοθεί καμία παράταση, καθώς η εξέταση της εργασίας θα ξεκινήσει αμέσως μετά σε χρονο-θυρίδες μέσα στο διάστημα της εξεταστικής Ιανουαρίου – Φεβρουαρίου, τις οποίες θα δηλώσετε (θα σταλεί σχετικό μήνυμα).

9. Εξέταση και Βαθμολόγηση Εργασιών

Η εξέταση των εργασιών θα γίνει στα μέλη των ομάδων, μέσω MS Teams. Θα καταρτιστεί κατάλογος με τις υποβληθείσες εργασίες και την εβδομάδα που θα ακολουθήσει την καταληκτική ημερομηνία παράδοσης της εργασίας, θα σας καλέσουν οι βοηθοί του μαθήματος ανά ομάδα να παρουσιάσετε τη δουλειά σας. Κατά την εξέταση των εργασιών, τα μέλη των ομάδων θα εξεταστούν χωριστά πάνω στην εργασία και θα βαθμολογηθούν επίσης χωριστά.

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

Κριτήριο	Επίπτωση	Σχόλιο
Μη χρήση της βιβλιοθήκης SGG	Απόρριψη εργασίας	Η <u>αποκλειστική</u> χρήση της βιβλιοθήκης SGG είναι υποχρεωτική
Παράδοση εργασίας που δεν εμπίπτει θεματολογικά στην εκφώνηση	Απόρριψη εργασίας	Αν δεν είστε σίγουροι για το αν το θέμα που επιλέξατε είναι «arcade game» ή συναφές παιχνίδι, ρωτήστε.
Μη χρήση κληρονομικότητας	Έως -1,0	Βλ. ενότητα «Υλοποίηση»
Μη πολυμορφική κλήση κοινών μεθόδων	Έως -1,0	Βλ. ενότητα «Υλοποίηση»
Μη δήλωση και υλοποίηση κλάσης βάσης για στοιχεία με κοινή λειτουργικότητα και κλάσης κατάστασης GlobalState	Έως -0,5	Βλ. ενότητα «Υλοποίηση»
Μη ικανοποιητικός σχεδιασμός εφαρμογής	Έως -0,5	Βλ. ενότητα «Υλοποίηση»

Μη λειτουργική διεπαφή με το χρήστη / έλεγχος ροής παιχνιδιού	Έως -0,5	-
Υλοποίηση επιθυμητών (μη υποχρεωτικών) χαρακτηριστικών	Έως +0,5	Βλ. ενότητα «Υλοποίηση»

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.