

ARCH 4342 ΠΛΗΡΟΦΟΡΙΚΗ:

Υπολογιστικές Προσεγγίσεις στις Δημιουργικές Τέχνες και Επιστήμες

Δημήτριος Μπίμης - ar24759 - 2ο εξάμηνο

Assignment 03:

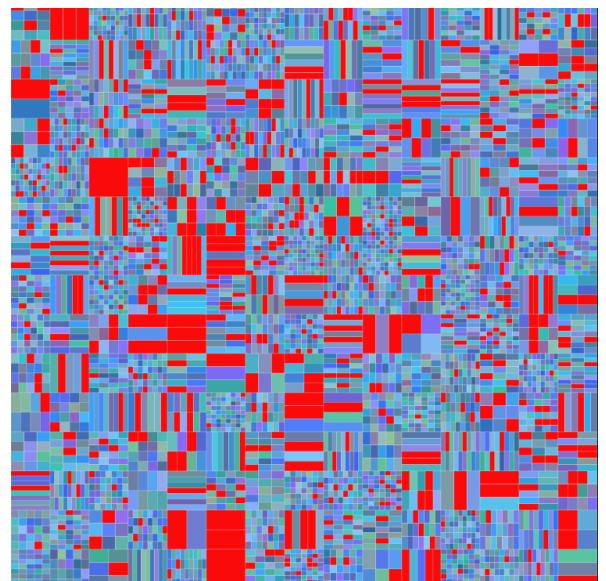
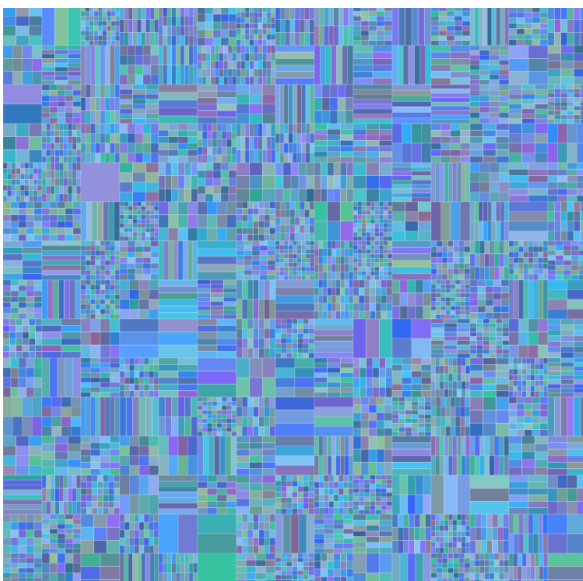
Πρόβλημα 1 / Arrays & Iteration:

<https://editor.p5js.org/mhtsos/sketches/ly6SSPqC>

Ψευδοκώδικας:

1. Δημιουργία array για τα υποκελιά (subcells)
2. Ορισμός αριθμού γραμμών και στηλών του βασικού κανάβου
3. Υπολογισμός πλάτους και ύψους κάθε κελιού
4. Για κάθε βασικό κελί: (διπλός βρόχος σε γραμμές και στήλες):
 - γέμισμα του κελιού με τυχαία απόχρωση πράσινου
 - επιλογή τυχαίου αριθμού υπογραμμών και υποστηλών (1 έως 9)
 - Υπολογισμός διαστάσεων κάθε υποκελιού
5. Για κάθε υποκελί εντός του βασικού κελιού: (διπλός βρόχος):
 - υπολογισμός διαστάσεων και συντεταγμένων
 - αποθήκευση κάθε υποκελιού(συντεταγμένες,χρώμα) στο array
6. Για κάθε υποκελί:
 - Αν είναι ενεργοποιημένο: κόκκινο χρώμα
 - Αλλιώς: αρχικό του χρώμα
7. Για κάθε υποκελί:
 - Αν το κλικ είναι μέσα στα όριά του: Αναστροφή της κατάστασης ενεργοποίησης (on/off)

Αποτέλεσμα του κώδικα:



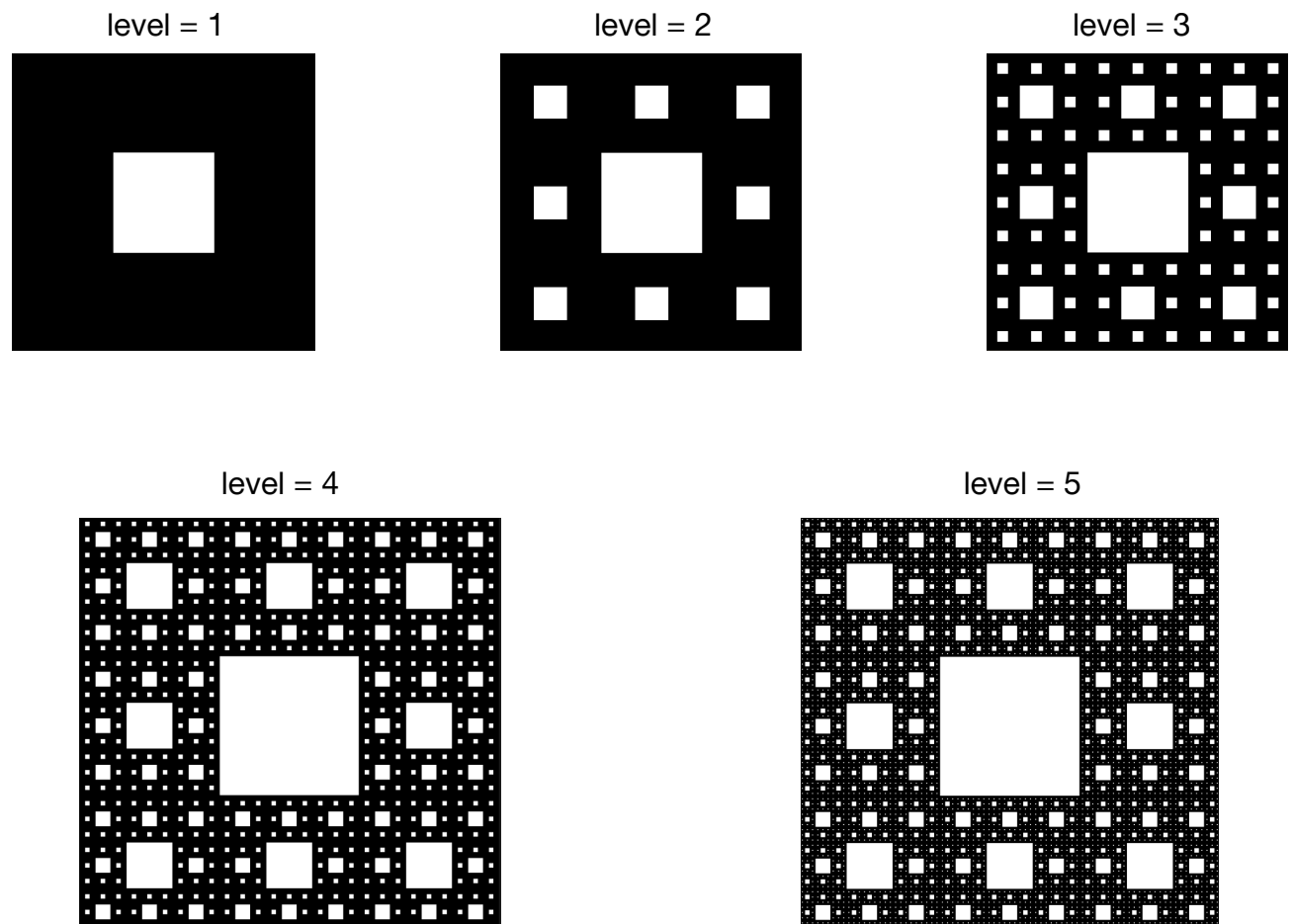
Πρόβλημα 2 / Recursion: Fractal City:

<https://editor.p5js.org/mhtsos/sketches/m8RluLtGd>

Ψευδοκώδικας:

1. Όρισε ένα array που ονομάζεται "squares" με ένα μόνο τετράγωνο που καλύπτει ολόκληρο τον καμβά ($x=0$, $y=0$, $size=600$)
2. Όρισε επίσης μια μεταβλητή "level" για να ελέγχει πόσα αναδρομικά επίπεδα θα δημιουργηθούν.
3. Δημιουργήστε ένα νέο array "next" για να αποθηκεύσεις το νέο σύνολο τετραγώνων.
4. Για κάθε τετράγωνο που βρίσκεται στο array "squares":
Χωρίστε το σε ένα πλέγμα 3×3 (9 μικρά τετράγωνα συνολικά)
5. Για κάθε μικρό τετράγωνο:
 - Εάν είναι το κεντρικό τετράγωνο:
 - Σχεδιάστε το με μαύρο χρώμα στον καμβά
 - Διαφορετικά:
 - Αποθηκεύστε το στο 'next' για το επόμενο επίπεδο
6. Αύξηση του μετρητή "level".
7. Εάν το τρέχον επίπεδο είναι ίσο με 5, σταματήστε τον βρόχο.

Αποτέλεσμα του κώδικα:



Computer Game:

<https://editor.p5js.org/mhtsos/sketches/VOXcMoCjz>

Στα πλαίσια της άσκησης μας ζητήθηκε να συνδυάσουμε τις έννοιες των arrays, βρόχων (loops), και conditionals (if, then, else, switch, case), για να δημιουργήσουμε μια διαδραστική εφαρμογή παιχνιδιού computer game (arcade or puzzle). Το παράδειγμα που δόθηκε ήταν το snake game με την ελευθερία να επεκταθούμε.

Βλέποντας τα χαρακτηριστικά του παιχνιδιού snake game μου ήρθε στο μυαλό ένα παλιό viral παιχνίδι, το Agar.io. Το Agar.io είναι ένα δωρεάν online παιχνίδι όπου οι παίκτες ελέγχουν ένα κελί, τρώνε μικρότερα κελιά για να μεγαλώσουν και αποφεύγουν να φαγωθούν από μεγαλύτερα. Κυκλοφόρησε τον Απρίλιο του 2015, απέκτησε γρήγορα δημοτικότητα λόγω του απλού αλλά εθιστικού gameplay του και έγινε viral hit.

Επομένως δημιούργησα ένα παιχνίδι βασισμένο σε κάποια από αυτά τα χαρακτηριστικά.

Ψευδοκώδικας:

1. Ορίστε ένα αντικείμενο παίκτη με θέση, ακτίνα και ταχύτητα.
2. Ορίστε ένα array για να κρατάτε το 'φαγητό'.
3. Ορίστε μια σταθερά foodCount με τιμή 100 για να καθορίσετε τον αριθμό του φαγητού στην οθόνη.
4. Ορίστε μια boolean gameOver για να ελέγξετε αν ο παίκτης έχει κερδίσει, αρχικά ορισμένη σε false.
5. Ορίστε μια μεταβλητή restartBtn για την αποθήκευση του στοιχείου του κουμπιού επανεκκίνησης.
6. Ορίστε το initGame function για να αρχικοποιήσετε την κατάσταση του παιχνιδιού.
7. Ορίστε τη θέση του παίκτη στο κέντρο του καμβά, την ακτίνα στο 20 και την ταχύτητα στο 2.
8. Καθαρίστε το 'food' array.
9. Δημιουργήστε νέα τροφή με τυχαία θέση και χρώμα και προσθέστε τη στο 'food' array.
10. Ορίστε το gameOver σε false.
11. Αν το gameOver είναι true:
12. Εμφανίστε το κείμενο "You Win!" με πράσινο χρώμα, κεντραρισμένο στην οθόνη.
13. Αν το κουμπί επανεκκίνησης δεν έχει δημιουργηθεί ακόμα:
 - Δημιουργήστε ένα κουμπί "Restart" και τοποθετήστε το κάτω από το μήνυμα.
 - Αναθέστε στο κουμπί να καλεί το restartGame όταν το πατήσετε.
 - Σταματήστε την περαιτέρω εκτέλεση του βρόχου του παιχνιδιού.
14. Δημιουργήστε ένα διάνυσμα που δείχνει από τον παίκτη προς το ποντίκι.
15. Κανονικοποιήστε αυτό το διάνυσμα και πολλαπλασιάστε το με την ταχύτητα του παίκτη.
16. Μετακινήστε τον παίκτη προς το ποντίκι χρησιμοποιώντας αυτό το διάνυσμα.



17. Κάθε 60 καρτέ:
- Εάν η ακτίνα του παίκτη είναι μεγαλύτερη από 10, συρρικνώστε την κατά 1
18. Εάν η απόσταση από τον παίκτη έως το τρόφιμο είναι μικρότερη από την ακτίνα του παίκτη:
- Αφαιρέστε το συγκεκριμένο τρόφιμο από τον πίνακα.
 - Αυξήστε την ακτίνα του παίκτη κατά 1.
 - Προσθέστε ένα νέο τυχαίο αντικείμενο τροφής για να διατηρήσετε τον αριθμό των τροφών.
19. Για κάθε στοιχείο τροφής στον πίνακα:
- Σχεδιάστε έναν μικρό χρωματιστό κύκλο στη θέση του.
20. Σχεδιάστε τον παίκτη ως κόκκινο κύκλο
21. Εάν η ακτίνα του παίκτη είναι μεγαλύτερη ή ίση με τη μισή μικρότερη διάσταση της οθόνης:
- Θέστε το `gameOver` σε `true` (ο παίκτης έχει κερδίσει).
22. Καλέστε την `initGame` για να επαναφέρετε την κατάσταση του παιχνιδιού και να ξεκινήσετε ξανά.

Εικόνες από το τελικό αποτέλεσμα:

