

Άσκηση 1 HY-150

16/02/2024 - 01/03/2024

Σε αυτή την άσκηση θα φτιάξετε ένα πρόγραμμα που θα παράγει απλές εικόνες με τη μορφή PPM. Δίνεται κώδικας με παραδείγματα για το πως μπορεί να παραχθεί μια τέτοια εικόνα.

Μπορείτε να διαβάσετε περισσότερα για τη μορφή εικόνας PPM εδώ:

<https://en.wikipedia.org/wiki/Netpbm>

Είναι υποχρεωτικό να γίνει η χρήση Exceptions για τον έλεγχο των εισόδων που δίνει ο χρήστης και να εμφανίζονται τα κατάλληλα μηνύματα.

A. Απλό μενού και I/O [25%]

Φτιάξτε ένα απλό μενού για να μπορεί ο χρήστης να αλληλεπιδρά με το πρόγραμμά σας.

Ο χρήστης θα πρέπει να μπορεί να επιλέξει το όνομα της εικόνας που θα δημιουργηθεί καθώς και τις διαστάσεις αυτής (ύψος, πλάτος). Σε περιπτώσεις που ο χρήστης δώσει λάθος τύπου είσοδο για τις διαστάσεις (αρνητικούς αριθμούς ή χαρακτήρες) θα πρέπει να τυπώνετε κατάλληλο μήνυμα λάθους και είτε να τερματίζετε το πρόγραμμά σας, είτε να ζητάτε από τον χρήστη να σας δώσει, ξανά, καινούρια είσοδο.

Προσοχή, σχεδιάστε το μενού με τέτοιο τρόπο ώστε να μπορείτε να προσθέσετε εύκολα περισσότερες επιλογές στα επόμενα ερωτήματα.

```
Image Generator - csdXXXX
```

```
Image name:
```

```
my_pattern_image
```

```
Image width:
```

```
-10
```

```
Invalid selection. Must be a positive number.
```

```
Image width:
```

```
awqa
```

```
Invalid selection. Must be a positive number.
```

```
Image width:
```

```
100
```

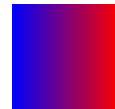
```
Image height:
100
...
```

B. Εικόνες ομαλής χρωματικής διαβάθμισης [75%]

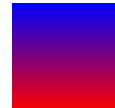
Εμπλουτίστε το μενού που φτιάξατε παραπάνω δίνοντας στο χρήστη την **επιλογή να διαλέξει ένα από τους 3 τύπους εικόνας** χρωματικής διαβάθμισης:

 $c1 = [0, 0, 1.0] \quad c2 = [1.0, 0, 0]$

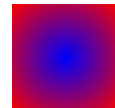
1. **Αριστερά - Δεξιά:** Ομαλή διαβάθμιση από το χρώμα 1 στο χρώμα 2, από αριστερά προς τα δεξιά.



2. **Πάνω - Κάτω:** Ομαλή διαβάθμιση από το χρώμα 1 στο χρώμα 2, από πάνω προς τα κάτω.



3. **Κέντρο - Άκρες:** Ομαλή διαβάθμιση από το χρώμα 1 στο χρώμα 2, από το κέντρο προς τις άκρες της εικόνας.



Εφόσον ο χρήστης επιλέξει τον τύπο της εικόνας θα χρειαστεί να επιλέξει **και 2 χρώματα**. Το κάθε χρώμα αποτελείται από 3 double αριθμούς (Red, Green, Blue - RGB) από 0.0 έως 1.0. Ο κάθε ένας από αυτούς τους αριθμούς δηλώνει το ποσοστό του εκάστοτε χρώματος

Η εικόνα σας θα πρέπει να γράφεται σε αρχείο με τη μορφή .ppm (στο παράδειγμα που σας δίνεται μπορείτε να δείτε πως γράφετε σε αρχεία .ppm).

Κατα τη διάρκεια δημιουργίας της εικόνας θα πρέπει το πρόγραμμα σας να ενημερώνει τον χρήστη για την εξέλιξη μέσω κατάλληλων μηνυμάτων :

```
Generating 94%
Generating 95%
Generating 96%
Generating 97%
Generating 98%
Generating 99%
Generating 100%
```

Έχετε υπόψη σας, ότι το πρόγραμμα σας θα **πρέπει να υποστηρίζει όλους τους δυνατούς συνδυασμούς ύψους/πλάτους** εικόνας. Επίσης, όπως και στο πρώτο ερώτημα, **πρέπει να διαχειριστείτε τυχόν λάθος είσοδο** από τον χρήστη με την εμφάνιση κατάλληλων μηνυμάτων λάθους.

Παρακάτω φαίνεται ένα παράδειγμα εκτέλεσης:

```
Image Generator - csdXXXX

Image name:
my_gradient_image
Image width:
1280
Image height:
720
Select Image type:
1. Gradient
2. Geometric Pattern
1
Select Gradient type:
1. Left - Right
2. Top - Bottom
3. Center Outwards
2
Enter color (r[0-1] g[0-1] b[0-1]):
0.5 0.7 1
Enter color (r[0-1] g[0-1] b[0-1]):
0.99 0.99 0.99
Generating 1%
Generating 2%
Generating 3%
Generating 4%

...

Generating 98%
Generating 99%
Generating 100%
Image was generated successfully!
```

Η εικόνα που παρήχθη είναι:



C. [BONUS] Παραγωγή γραφικής παράστασης [20%]

Προσθέστε στο μενού ακόμα μια επιλογή για την δημιουργία εικόνων με ημιτονοειδείς καμπύλες.

Η εικόνα που θα δημιουργείται θα πρέπει να αναπαριστά τη συνάρτηση $\eta\mu(x)$.

(χρησιμοποιήστε την συνάρτηση [double sin\(double x\)](#))

Η **μέγιστη τιμή της συνάρτησης (πλάτος συνάρτησης)** πρέπει να **αλλάζει αυτόματα ανάλογα με το ύψος της εικόνας** (π.χ. η συνάρτηση να πιάνει το 80% του ύψους της εικόνας), ενώ **το μήκος κύματος θα πρέπει να είναι σταθερό** ανεξαρτήτως του μήκους της εικόνας (δεν πρέπει να γίνεται stretching στον x άξονα για μεγαλύτερο μήκος εικόνας - δείτε το παράδειγμα παρακάτω).

Το πάχος της γραμμής της γραφικής παράστασης πρέπει να υπολογίζεται και αυτό αυτόματα, βάση του ύψους της εικόνας (π.χ. 5% του ύψους της εικόνας).

Εφόσον ο χρήστης επιλέξει τη δημιουργία μιας τέτοιας εικόνας θα πρέπει να **μπορεί να διαλέξει το χρώμα της γραμμής και του φόντου**.

Δίνεται ψευδοκώδικας για να σας βοηθήσει στην υλοποίηση:

για κάθε pixel p με θέση (x, y) :

Κανονικοποιημένο ύψος γραμμής

lwNorm = πάχος γραμμής/ύψος εικόνας

Κανονικοποιημένο x

xNorm = (x / (ύψος εικόνας - 1)) - 0.5

Κανονικοποιημένο y

yNorm = (y / (ύψος εικόνας - 1)) - 0.5

Απόσταση (Απόλυτη τιμή) από συνάρτηση

εάν $|yNorm - 0.4 * \eta\mu(xNorm/0.1)| < lwNorm$:

χρώμα_pixel = χρώμα_γραμμής

αλλιώς:

χρώμα_pixel = χρώμα_φόντου

Παρακάτω φαίνονται οι εικόνες που παρήχθησαν και δίπλα η είσοδος του χρήστη:

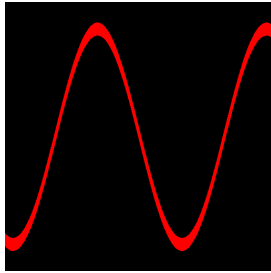


Image height: 512px
Image width: 512px
Line Color = [1.0, 0, 0]
Background Color = [0, 0, 0]

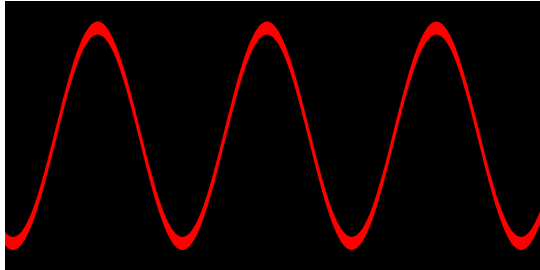


Image height: 512px
Image width: 1024px
Line Color = [1.0, 0, 0]
Background Color = [0, 0, 0]

Παρακάτω φαίνεται ένα παράδειγμα χρήσης του προγράμματος για τη δημιουργία μιας τέτοιας εικόνας:

```
Image Generator - csdXXXX

Image name:
my_pattern_image
Image width:
1024
Image height:
512
Select Image type:
1. Gradient
2. Geometric Pattern
2
Select Geometric Pattern:
1. Sine wave
1
Enter color (r[0-1] g[0-1] b[0-1]):
1 0 0
Enter color (r[0-1] g[0-1] b[0-1]):
0 0.47 1
Generating 1%
Generating 2%
Generating 3%
```

...

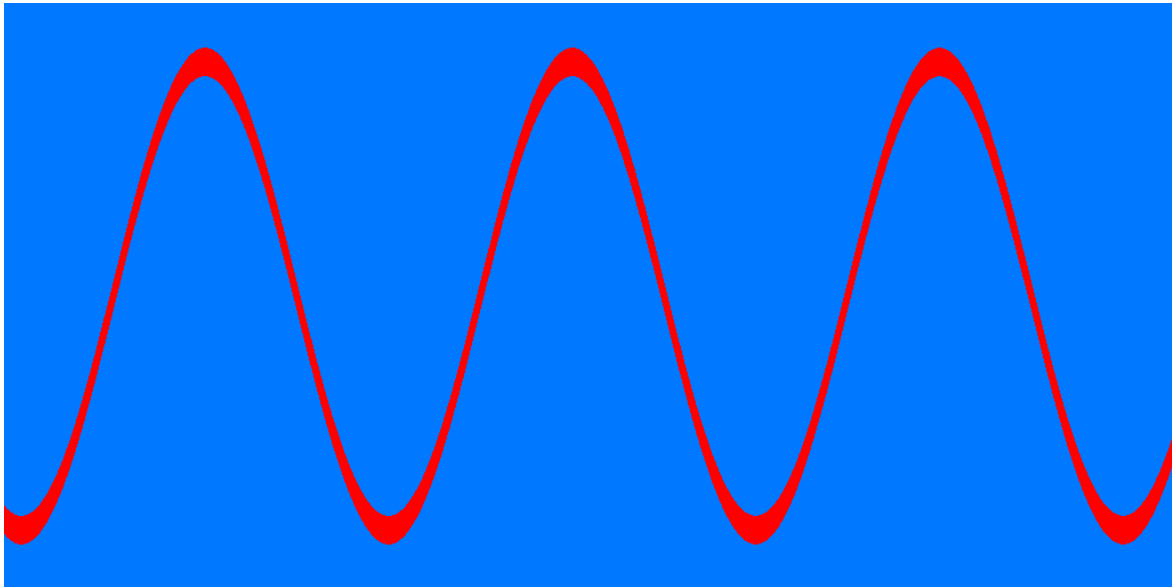
Generating 98%

Generating 99%

Generating 100%

Image was generated successfully!

Η εικόνα που παρήχθη είναι:

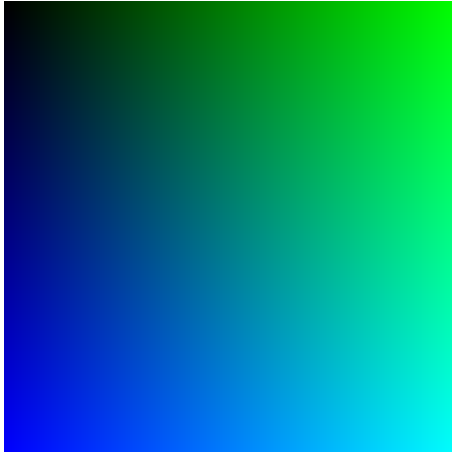


Υλοποίηση

Για την υλοποίηση θα χρειαστεί να κατεβάσετε τον φάκελο με το υλικό της άσκησης 1. Αυτός ο φάκελος περιέχει διάφορες βιβλιοθήκες με συναρτήσεις που θα σας βοηθήσουν, καθώς και ένα παράδειγμα με τη χρήση αυτών για τη δημιουργία μιας απλής εικόνας.

Example.cpp

Αυτό το αρχείο περιέχει κώδικα για τη δημιουργία της παρακάτω εικόνας:



Οι πρώτες γραμμές στον κώδικα:

```
#include "vec3.h"
#include "color.h"
```

χρειάζονται για να υπάρχει πρόσβαση σε βοηθητικές συναρτήσεις όπως η `write_color(...)`. Αυτές θα χρειαστεί να τις συμπεριλάβετε και στο αρχείο του κώδικά σας αν θέλετε να χρησιμοποιήσετε αυτές τις συναρτήσεις.

vec3.h

Αυτό το αρχείο περιέχει βοηθητικές συναρτήσεις για διανύσματα και πράξεις με αυτά. Ο τρόπος χρήσης των συναρτήσεων αυτών υπάρχει στο αρχείο `Example.cpp`.

color.h

Αυτό το αρχείο περιέχει βοηθητικές συναρτήσεις για τα χρώματα(`color`). Παραδείγματα χρήσης αυτών υπάρχουν στο αρχείο `Example.cpp`.

Assignment1.cpp

Εδώ θα χρειαστεί να γράψετε τον κώδικά σας.

Δομήστε σωστά τον κώδικά σας και σπάστε τις διάφορες λειτουργίες σε κατάλληλες συναρτήσεις.

Δοκιμή του Προγράμματος σας

Για να εμφανίσετε τις εικόνες PPM μπορείτε να:

- εγκαταστήσετε το plugin [PBM/PPM/PGM Viewer for Visual Studio Code](#) για το Visual Studio Code ή
- να ανατρέξετε εδώ <https://jumpshare.com/viewer/ppm> ή
- εδώ <https://0xc0de.fr/webppm/>

Υποβολή

- Η άσκηση είναι καθαρά ατομική.
- Τα παραδοτέα αρχεία θα ελεγχθούν για αντιγραφές. Τυχόν αντιγραφή ισοδυναμεί με μηδενισμό σας.

Παραδώστε όλο τον φάκελο με την υλοποίηση σας σε ένα .zip αρχείο με το όνομα Assignment1_csdXXXX.zip.