

בשבועות האחרונים אני כותב קוד לתצוגה של סיגנלים אנאלוגיים, כדוגמת סאונד, על המסך. העיקרון היה לצייר את החלק הממשי, המרוכב, את האמליפטודה, ואת הפאזה של הסיגנאל שנכנס לתוכנית. את הפאזה צריך לצייר פחות תדר ספציפי אותו מכניס המשתמש, כך אפשר לראות מה התדר של הסיגנאל שנכנס.

אחרי שעשיתי קצת פרוטו-משחקים בחיים שלי, החלטתי שבמקום לצייר למיקומים שלמים של פיקסלים על המסך, אצייר לחלקי פיקסל, על מנת שבעתיד אוכל לצייר בדיוק הגבוה מפיקסל בודד וכך להיות יכול לעשות אלגוריתמים של Anti aliasing בציר, שחוסר קיומם בתכנית בה אני עובד לפעמים פוגע במסקנות, דוגמאות בפרטי. אז איך עובדים עם חלקי פיקסל? פשוט, עובדים בfloating points (להלן, פלואטס), פלואטס הם תמיד חברים של כולם וקל לעבוד איתם. פעם פעם אף אחד לא עבד איתם כי הם עלו יותר ממספרים רגילים מבחינת ביצועים, אבל מאז כולם עובדים עם פלואטס בחומרה והם עולים פחות או יותר אותו דבר.

אז איפה יש לנו פלואטס בתכנית? א. הנתונים, החלק הממשי, המרוכב והאמפליטודה הם פלואטס, כמו כן הפאזה ב. התדר הספציפי שמכניס המשתמש הוא פלואט, וכמו כן הפאזות המתקבלות מהכפלת התדר במספר נקודת הדגימה. ג. המיקום על המסך של הדברים אותם אני רוצה לצייר (שמומרים לשלמים כשאני אשכרה בא לגשת לתמונה אותה אני מצייר)

אז למה אני כותב את זה בכלל? כי זה הורס לי את החיים. אוקיי, זה לא ממש, ויש קצת merit במאבק הזה, אבל זה עדיין די מעצבן, ומזמן לא כתבתי כלום ובעיקר לא דברים טכניים אז החלטתי לכתוב אותם כאן.

דוגמה ראשונה:

אחד הדברים שאני אוהב לעשות כשאני מצייר למסך, ובעיקר כשאני מצייר למסך בצורת גרף, הוא שאני מנסה להשתמש במשתנים שלי כמה שאני יכול. לדוגמה, אם אני אצייר נקודה במיקום עם x כלשהו, ובפעם הבאה אני אצייר נקודה במיקום עם $x + \text{deltax}$ ואם אני אעשה את זה אלף פעמים, אז אפשר פשוט לקבוע x התחלתי, וכל פעם לחבר deltax ל x -הזה. עולה השאלה, כמה נקודות אני יכול לצייר במסך? האדם השפוי יגיד רוחב המסך לחלק ל deltax -והאדם הזה כמובן ישגה לגמרי אם נשתמש בשיטה שהגדרתי למעלה בפלואטס.

בשביל להסביר את זה, אנו צריכים להבין שמספרי floating points הם בדיוק כאלה, הנקודה העשירונית שלהם צפה. כלומר, כל מספר מוגדר באמצעות מספר קבוע של ספרות דיוק (במחשב זה 23 ביטים, בדוגמאות להלן 2 ספרות עשרוניות) ומספר נוסף המגדיר באיזה חזקה של בסיס המספר הספרות האלה צריכות להיכפל. לדוגמה, המספר $3^{10} * 6.3$ הוא מספר פלואטינג פוינט המייצג את המספר 6300, האם למספר 6325 יהיה ייצוג שונה? רמז, לא, כי אין לנו מספיק ספרות דיוק כדי להבדיל ביניהם. אזי מסתבר שהמספר $6.3 * 10^3$ מייצג את תחום

המספרים [6250, 6350], והגדרה כלשהי תגדיר את זה התחום הפתוח, הסגור או החצי פתוח של המספרים האלה.

אוקיי, אז אם אני מתחיל ב-x מסוים ואז מגדיל אותו כל פעם ב-1.1, אז בהתחלה לא תהיה לי בעיית דיוק, כל נקודה תהיה במיקום 1.1 כפול מספר הנקודה. בשלב מסוים, הגענו ל- $x=11$, ואז כשאנחנו רוצים לחבר 1.1 אנחנו מקבלים 12, שזה לא כזה נורא כי זה עדיין 1.1*12, אבל כבר ב-16 השיטה שלנו תביא לנו $x=16$ ו- $1.1*16=17.1$. לכן אם הרחוב שלנו הוא $\Delta x * 16$, אנחנו יכולים להכניס נקודה אחת נוספת, ואם יש לנו הרבה יותר, אז אנחנו נתבלבל הרבה יותר. למה זה בעיה? כי אז אני מגיע לחור ענק בסוף המסך שלנו בגודל עשרות פיקסלים שאף אחד לא מבין מאיפה הוא מגיע.

גרוע מכך, אם ארצה שליבות שונות יציירו חלקים שונים של המסך, אני יכול להגיד שאחת תצייר את מיליון הנקודות הראשונות, והשנייה את מיליון הנקודות השניות. איפה ארצה שהליבה השנייה תתחיל לצייר? ב- $\Delta x * 1000000$. הליבה הראשונה תצייר מיליון נקודות, יהיה חור ענק באמצע בגלל בעיית הדיוק, ואז הליבה השנייה תתחיל לצייר. מצוין. פתרון לכך יהיה להוסיף ל-x מיליון פעם את Δx ואז להבין איפה מסתיימות המיליון נקודות הראשונות, ואז להגדיר לליבה השנייה להתחיל לצייר משם.

דוגמה שנייה:

התקדמנו, ואנחנו רוצים לחלק את הגרף שלנו לחלקים של מיליון נקודות, וכל ליבה בתורה תתפוס מיליון נקודות ותצייר אותן למסך. אנחנו יודעים כבר שה- Δx בין הקטעים הוא לא $\Delta x * 1000000$ אלא מיליון חיבורים של Δx -ל. אז יופי, כל ליבה תתחיל לצייר ב- $\Delta x * i$ - והכל יהיה טוב ויפה, ושוב, עושים את זה ונוצר חור בין הציורים, כי $\Delta x + \Delta x = 2 * \Delta x$ ו- $\Delta x + \Delta x = 10 * \Delta x$ ממש לא שווה ל- $\Delta x * 11$.

אז זה אותו הסיפור עוד פעם, אי אפשר להכפיל מספרים ולהגיע לאותו מיקום, אבל איך פותרים את זה אחת ולתמיד? פשוט צריך לזרוק את שיטת תוספת ה- Δx לכל נקודה. מעתה, כל נקודה מצוירת ב- Δx כפול מספר הנקודה, עולה קצת יותר, אבל אני לא ארג', מוכן לשלם על זה. אז עכשיו הכל רגוע ותשקוט הארץ 40 שנה.

ובימים ההם אין מלך על ישראל

אז למיקום על המסך הכל בסדר, פתרנו את זה, לא צריך לריב יותר, ולגבי הערכים של הנקודות אנחנו לא מחברים יותר מדי דברים אז זה לא משנה, אבל נשאר לנו עוד ערך אחד, וזה התדר החיצוני והפאזות הנגזרות ממנו. עדן, תפסיק לעשות לנו סיפורים, פשוט תכפיל את התדר במספרים ותעזוב אותנו בשקט. אבל אי אפשר לעשות את זה, כי בניגוד לדוגמאות מקודם, פאזות צריכות תמיד להיות בתחום $[-\pi, \pi]$ מה יקרה אם אני אכפיל את התדר, בסדר גודל של 0.1 ב-10 מיליון? אני אקבל מספר ממש גדול, ששייצג תחום ברוחב של 0.1 בערך. אם נחזור לדוגמה מקודם, נגיד והתדר שלי הוא 0.1 ואני רוצה לקבל את הפאזה בנקודה האלף, אז אני מכפיל את

התדר באלף ומקבל $10^{10} * 10^1$ שמייצג תחום ברוחב של 10. עולה השאלה, איזו נקודה התחום הזה מייצג בתחום $[-\pi, \pi]$? נכון, אף אחת מהן או את כולן. אז מה צריך לעשות? אז מה שאני עושה זה כל הזמן מוסיף 0.1 ומשאיר את הערך שיוצא לי בין π -ל- π ככה אני לא מאבד דיוק.

ומה קורה אם אני מצייר כמה קטעים בנפרד על המסך, אז לכל אחד מהם אני צריך להגדיר את הפאזה ממנה הוא צריך להתחיל, אחרת אי אפשר להבטיח שלא תהיה קפיצת פאזה במעברים בין קטע לקטע, או יותר מזה, אפשר להבטיח שתהיה.

ועכשיו: להזיז את המסך

אז בוא נגיד ואני לא מתחיל לצייר מנקודת המידע ההתחלתית, אלא אני יכול להגדיר איזו נקודה תהיה הנקודה הראשונה, ככה אני יכול להסתכל על קטעים שונים בקובץ. אבל בזמן שאני זז ממקום למקום, אני לא רוצה שתשתנה הפאזה של התדר שלי. אז מה שצריך לעשות זה להכפיל את הנקודה ההתחלתית החדשה בתדר, ולהתחיל את הציור מהפאזה הזאת. אוקיי, ואתם בטח מתחילים להבין שכשאני אומר להכפיל אני מתכוון לחבר מלא פעמים עד שאני מגיע לאן שאני רוצה, כי אי אפשר להכפיל כאן. אוקיי, אז אם אני במקום מתקדם בקובץ, לדוגמה הנקודה ה-27 מיליון, אז אני מחבר 27 מיליון פעמים את הדבר הזה ואני מקבל את מה שאני רוצה, אבל באיזה מחיר? 27 מיליון פעולות פלואטס, פלוס פעולות מודולו שמשאירות את הפאזה בין π -ל- π זה לוקח בערך 30 מ"ש בכל פריים במקרה הטוב ביותר, בפועל זה לקח פי 4 בערך, אולי אני צריך לטפל במשהו שם. אז מה אפשר לעשות? לשמור את נקודת הפאזה ההתחלתית בכל פריים ולהתחיל ממנה. ספציפית אני שומר רק את הפאזה ממנה מתחיל קטע מיליון הנקודות שלי, אבל זה לא כזה משנה.

אם אני זז במעלה הקובץ, אין לי בעיה, אני מוסיף כל פעם שאני עובר קטע מיליון פעמים וסוגר את זה בפריים אחד. ומה אם אני זז במורד הקובץ? אפשר להבין שניסיתי להוסיף את מינוס התדר שלי, אבל מה, אפשר להבין שזה לא ממש עובד, כי אם מוסיפים מיליון נקודות ואז מחסרים מיליון נקודות לא מגיעים למספר המקורי. אפשר תמיד לחזור לחשב את 27 מיליון הנקודות האלה שוב, אבל זה ממש יקר, ותנועה בסביבת הקובץ צריכה להיות משהו שזורם. כמו כן, שינוי תדר צריך להיעשות בקלות יחסית. קיצור בינתיים אני לא יודע מה אני אעשה, אבל בטח אפשר לפתור את זה איכשהו.

שלמה אמר לי שכדאי להגיד בסוף שבעצם כל מה שהמאמר הזה אומר הוא שאין אסוציאטיביות $((a+b)+c=a+(b+c))$ ודיסטריבוטיביות $(a(b+c)=ab+ac)$ בפלואטס, וזה נכון.

אני עדיין מנצח,

עדן סגל.