

RAPORT

DE IMPLEMENTARE A PROIECTULUI INDIVIDUAL

I. TEMA PROIECTULUI

Nume proiect: Stiva Mini
Autor: Cezar Petreanu
Data raport: 18.01.2023
Versiune: Finală

Tipul programului: aplicație desktop, software de artă grafică
Limbaj de programare: Java

Link-ul proiectului (GitHub): <https://github.com/CezarPetreanu/STIVA-MINI>

II. DESCRIEREA PROIECTULUI

Descriere generală

Stiva Mini este un program de creat și editat desene de tip pixel art special concepute să fie reprezentate sub forma unui obiect 3D prin tehnica „sprite stacking” (stivuirea graficilor 2D). Proiectul are scopul de a pune în aplicare tehnica și de a oferi dezvoltatorilor de jocuri o unealtă care îi va ajuta să obțină acest efect pentru graficile lor.

Concepte introduse / utilizate în proiect

Programare Orientată pe Obiect (POO): paradigm de programare „ce se concentrează pe proiectarea, implementarea și utilizarea ierarhiilor de clasă ce oferă polimorfism și încapsulare” (Stroustrup, Bjarne, 2013). Programarea Orientată pe Obiect introduce următoarele concepte:

- Clase – descriu un obiect definit de attribute care poate executa cod prin metodele lor
- Abstractizare – simplificarea unui concept, al clasei
- Încapsularea – ascunderea datelor și funcționalităților
- Polimorfismul – posibilitatea unui obiect de a fi implementat în mai multe moduri
- Moștenire – implementarea unor clase ce conține, pe lângă attributele și metodele proprii, proprietățile clasei moștenite

Serializarea: procesul de transformare a unui obiect într-un șir de octeți astfel încât să poată fi convertit înapoi într-o copie a obiectului (Oracle).

Interfața grafică: facilitează o interacțiune ușoară între utilizator și program cu ajutorul elementelor grafice: ferestre, iconițe, meniuri și cursor.

Pixel art: stil de desen digital. Imaginile sunt de regulă mici ca dimensiune, la nivelul pixelilor.

Sprite: imagini folosite în jocurile video 2D pentru a reprezenta un element, de exemplu un personaj sau un obiect. Sprite-urile pot fi formate dintr-o singură imagine sau cadru, fiind statice, sau din mai multe cadre care, derulate, animează elementul.

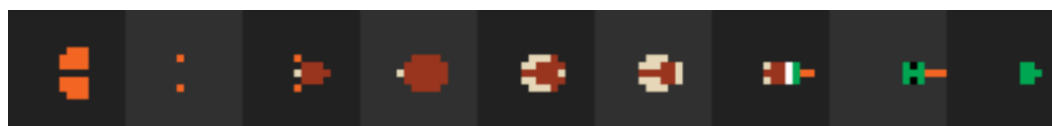
Strat (layer): planșe folosite în programele de creat și editat imagini pentru a separa porțiuni din desen. Straturile formează împreună imaginea integrală, dar utilizatorul poate modifica fiecare strat individual fără să le afecteze pe celelalte sau întreg desenul.

Sprite stacking: tehnică ce folosește mai multe sprite-uri 2D aranjate (stivuite) în așa fel încât să creeze un efect 3D. Pentru a obține acest efect, desenatorul trebuie să urmeze pașii:

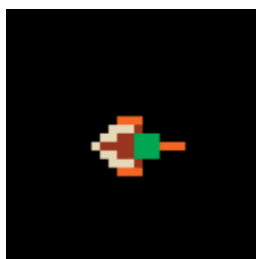
1. Să imagineze obiectul 3D dorit tăiat în mai multe secțiuni transversale, perpendiculare cu podeaua,
2. Să deseneze fiecare secțiune separat, în ordine,
3. Să afișeze secțiunile transversal în ordine. Fiecare secțiune este deplasată puțin în sus față de secțiunea precedentă.

Secțiunile transversale sunt desenate și introduse separat în proiectul jocului de artist, iar programatorul are sarcina de a programa secțiunile în așa fel încât să fie aranjate conform punctului 3.

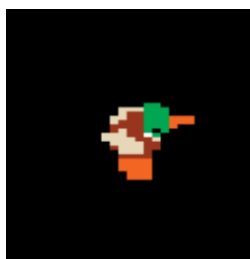
Demonstrație:



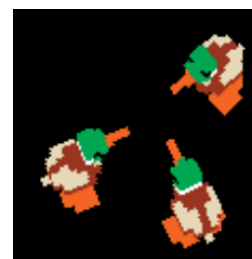
Secțiunile transversale ale unei rațe, desenate separat.



Secțiunile stivuite, fără spațiu între ele.

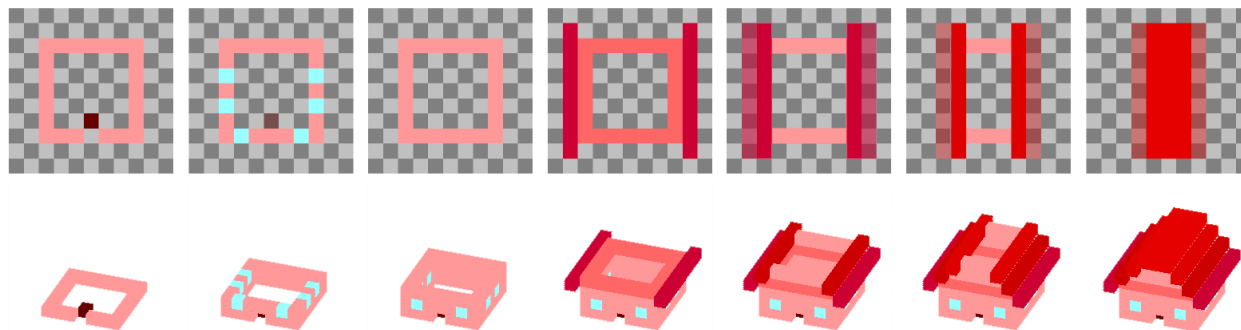


Fiecare secțiune este deplasată în sus față de cea precedentă.



Mai multe unghiuri.

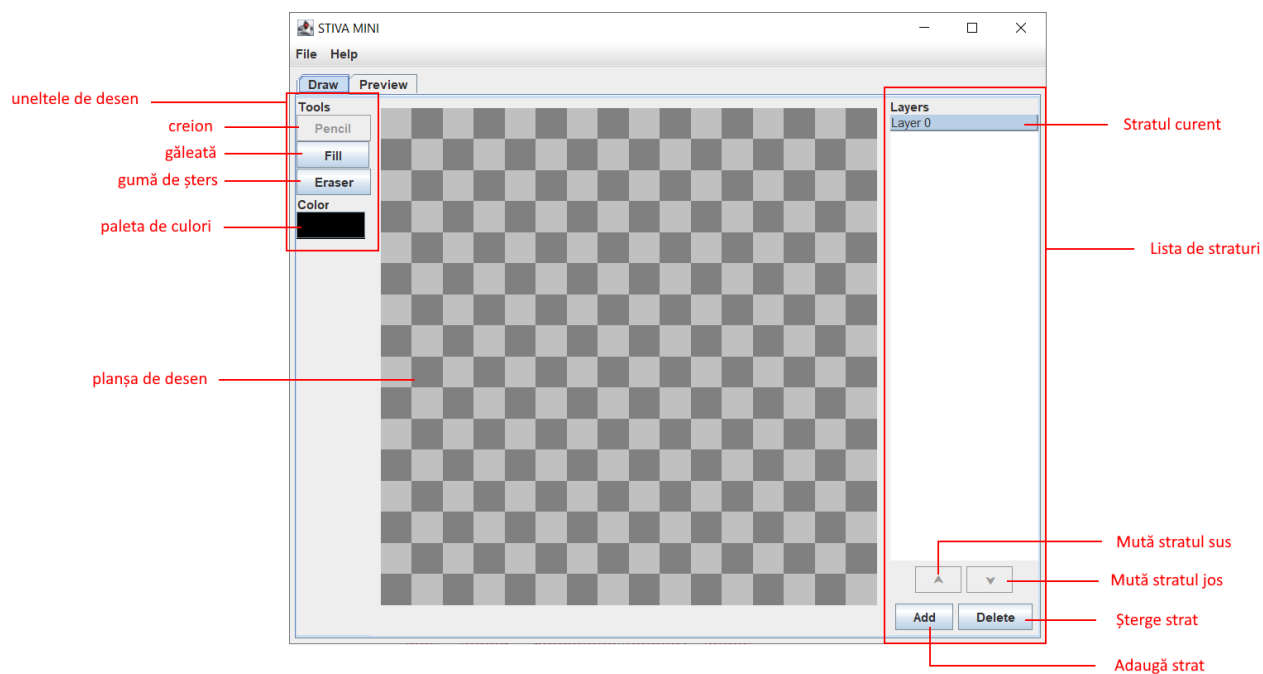
Un alt mod de a privi secționarea obiectului și straturile rezultate este prin imaginarea construirii unei clădiri, privind de sus.

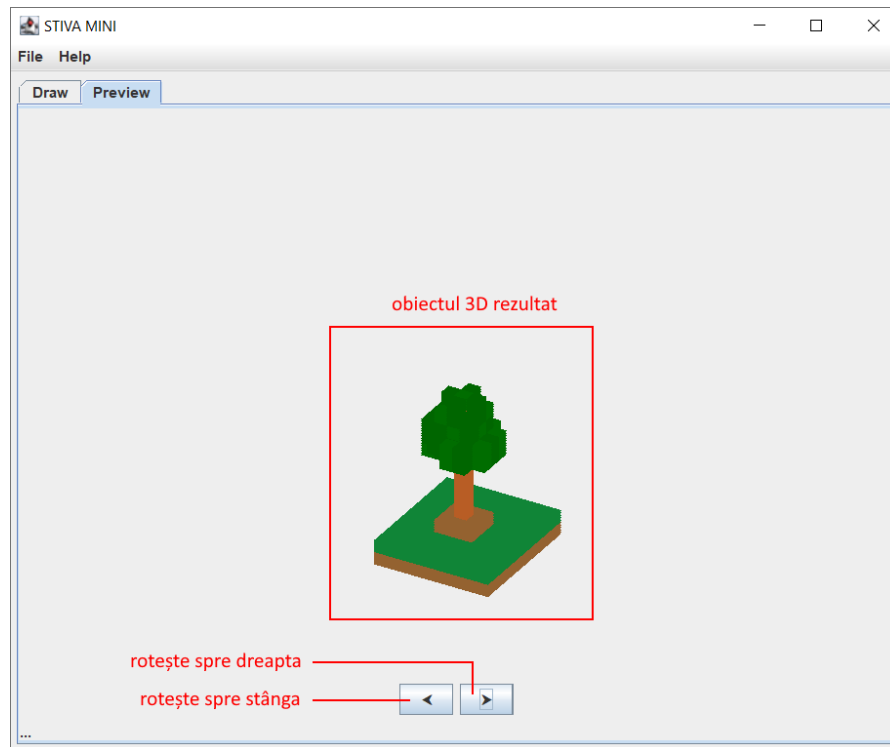


II. ARHITECTURA PROIECTULUI

Interfața grafică

Modul „Desen”





Clase

DrawArea – clasa interfeței grafice

În clasa DrawArea au loc toate funcționalitățile programului. Clasa construiește interfața grafică, ia input de la utilizator, desenează pe ecran și efectuează operațiile de salvare și deschidere a fișierelor și de exportare a imaginilor.

```
private Color color;           // culoarea curentă
private int currentLayer;      // indexul stratului curent
private int numberOfLayers;    // numărul de straturi
private JPanel contentPane;    // panoul ce conține elementele interfeței grafice
private tools tool;           // unealta curentă (PENCIL, FILL, ERASER)
private List <Color[][]> layer = new ArrayList<>(); // Lista ce conține toate
                                                    // straturile de desen
private JTable table;         // tabel ce conține numele straturilor
                               // folosit mai mult să arate utilizatorului cum sunt
                               // aranjate straturile
private int previewAngle;     // unghiul la care este prezentat modelul
private int previewPixelSize; // mărime constantă a pixelilor modelului
private String path;          // calea unde se află fișierul proiectului

private boolean newProject = true; // este un proiect nou?
private boolean modified = false;  // a fost modificat recent?

// folosite pentru a întreba utilizatorul dacă vor să salveze
// înainte să iasă din program sau să creeze/deschidă un proiect nou
```

MyCanvas – un obiect de tipul Canvas (din AWT), care conține informații în plus despre mărimea planșei efective și mărimea unei matrice de culori(unităților).

Funcționalități

Desenează pe strat:

Planșa de desen (obiectul DrawArea) conține o zonă de desenat ce poate înregistra acțiunile utilizatorului (click-stânga).

Atunci când utilizatorul apasă pe planșa de desen, programul va memora poziția mouse-ului în acel moment. Programul calculează apoi poziția aproximativă unde va avea loc modificarea în stratul curent, reprezentat ca o listă de matrice ce conțin culori (un set de 4 valori: roșu, verde, albastru, transparența), în funcție de poziția mouse-ului.

Modificarea în matrice depinde de unealta selectată. În cazul creionului, de exemplu, valoarea de pe poziția aproximată se va schimba în culoarea curentă, iar în cazul gumei de șters, valoarea va deveni nulă. După ce s-a făcut schimbarea în matrice, pe planșa de desen se va desena un pătrat pe poziția aproximativă calculată înainte (și modificată ca să se potrivească cu dimensiunile planșei).

Creează/Șterge strat:

Utilizatorul poate crea sau șterge straturi. Atunci când apasă pe butonul „Add layer”, planșa de desen (obiectul DrawArea) va adăuga în lista sa de straturi. Stratul nou adăugat va fi o matrice de culori goală (toate elementele ei sunt nule). La final, stratul nou se afișează pe ecran.

Atunci când apasă pe butonul „Delete layer”, programul se va asigura mai întâi dacă după ștergerea stratului vor mai rămâne alte straturi. Dacă vor mai rămâne cel puțin un strat, atunci se va șterge stratul curent, matricea de culori de pe poziția curentă din listă. Dacă, în schimb, va rămâne un singur strat, butonul se va dezactiva, pentru a evita ștergerea tuturor straturilor din listă. Programul va avea întotdeauna minim un strat. La final, se trece la stratul anterior (sau la stratul următor, dacă stratul șters e primul din listă) și se afișează.

Vizualizează modelul 3D format din straturile desenate:

Atunci când utilizatorul apasă pe butonul „Preview”, planșa de desen va crea o nouă fereastră (și un nou obiect de tipul ImagePreview) căruia lista de straturi.

Fereastra obiectului ImagePreview conține o zonă în care pot fi desenate grafici. Programul va parcurge lista de straturi și creează pentru fiecare un obiect de tip imagine. Se parcurge matricea curentă, iar în imagine se desenează un pătrat, culoarea și poziția acestuia fiind determinate de valoarea, respective poziția elementului curent din matrice. Imaginea va fi apoi desenată în fereastră, aplatizată și rotită. Fiecare imagine corespunzătoare matricii va fi desenată mai sus decât imaginea matricii anterioare.

Salvarea/Deschiderea proiectului desenului

Utilizatorul poate să salveze proiectul desenului, straturile desenate și ordonate, într-un

fișier special, și îl poate deschide înapoi în program ca să îl modifice. Funcționalitatea se realizează cu ajutorul serializării. Atributele planșei de desen sunt serializate într-un fișier de octeți, iar programul îi cere utilizatorului să aleagă locația unde va fi salvat fișierul. La deschidere, fișierul este deserializat și încărcat într-un obiect, ca pe urmă programul să încarce informațiile obiectului în el.

Unele obiecte, precum elementele de interfață grafică, nu pot fi serializabile. Printre aceste obiecte se numără și tabelul care conține ordinea straturilor. Din fericire, lista cu straturi este serializabilă, așa că tabelul se va goli și se va reumple cu ordinea straturilor conform listei.

Exportarea imaginii

După ce utilizatorul este mulțumit de obiectul rezultat, poate să salveze toate straturile într-o singură imagine. Programul creează și procesează o nouă imagine de lățimea planșei și de lungimea planșei înmulțită de atâtea ori câte straturi sunt. Apoi straturile sunt desenate una lângă alta pe imaginea goală. La final, programul va cere utilizatorului să aleagă locul unde va salva imagine. Imaginea va fi de tipul PNG (Portable Network Graphics), deoarece poate conține și culori transparente.

Indicatori de performanță realizați în proiect

S-a observat o problemă importantă: modul în care sunt desenate planșele și modelul nu sunt eficiente. Deoarece se creează un obiect 3D, ce prezintă lungime, lățime și înălțime, anumite operații au complexitatea de $O(n^3)$, unde n reprezintă mărimea matricei de culori sau mărimea imaginii.

Un indicator clar care atenționează performanța scăzută o reprezintă timpul de desinare pe ecran. Sunt momente în care desenul „sacadează”, observându-se clar cum fiecare pixel este desenat pe rând pe ecran. Vizualizarea modelului final are cel mai mult de suferit, întrucât desenează mai multe straturi decât este necesar ca să ofere un aspect mai solid.

III. TEHNOLOGII UTILIZATE

Limbaje

Java

Java este un limbaj de programare orientat pe obiect.

„Simplitatea este unul dintre obiectivele principale ale designului Java. Simplitatea și eliminarea multor „trăsături” de valoare dubioasă din strămoșii săi C și C++ mențin Java relativ mic și reduc sarcina programatorului în producerea de aplicații de încredere.” (James Gosling, Henry McGilton, 1996)”

Java.awt, Swing:

„Abstract Window Toolkit” (AWT) este un pachet ce conține clasele necesare pentru a implementa aplicații cu interfețe grafice. Swing este considerat o versiune îmbunătățită a pachetului AWT, oferind în principal mai multe elemente grafice

Funcții

Straturile sunt reprezentate ca tablouri bidimensionale ce stochează o culoare reprezentată de patru valori, RGBA: roșu (red), verde (green), albastru (blue) și transparența (alpha). Astfel, crearea desenului presupune operații de modificare a tablourilor bidimensionale.

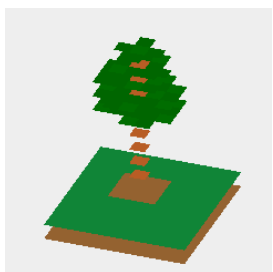
drawPixel(MyCanvas canvas, Graphics g, MouseEvent e) desenează un pătrat ce reprezintă un pixel pe planșa de desen. Pentru a face asta, prima dată ia poziția cursorului pe planșă și calculează poziția aproximativă a culorii pe matricea de culori. În funcție de unealta folosită, elementul din matrice va avea valoarea culorii curente (un pixel este desenat cu creionul) sau null (un pixel este șters cu guma de șters).

fillColor(int x, int y, Color seekColor) folosește un algoritm de umplere (flood fill) pentru a umple în mod recursive o zonă de o culoare din matrice cu o culoare nouă, fără să treacă prin alte zone de culori diferite. Începând dintr-o poziție dată în matrice, schimbă culoarea de pe poziția curentă, apoi se autoapelează pe pozițiile vecine dacă au aceeași culoare ca și culoarea precedent de pe poziția curentă.

canvasUpdate(MyCanvas canvas, Graphics g) este apelată de regulă când se schimbă stratul pe care se lucrează. Se golește planșa de desen, se parcurge matricea de culori și se desenează pe planșă pătrate/pixeli.

paintLayer(Canvas canvas, int layerNumber) se folosește la vizualizarea obiectului 3D rezultat. Creează o imagine 2D folosind matricea de culori a stratului current într-un mod asemănător metodei canvasUpdate. Imaginea rezultată poate fi manipulată (mărită, rotită, aplatizată) însă, fapt ce se observă în metoda drawPreview.

drawPreview(Canvas canvas) desenează pe planșa corespunzătoare vizualizării modelul final. Se parcurge lista straturilor, a matricelor de culori, și se generează o imagine cu ajutorul metodei paintLayer. Imaginile straturilor sunt desenate în ordinea precizată de tehnica sprite-stacking. Imaginea unui singur strat va fi desenată la rândul ei de mai multe ori, să ofere obiectului un aspect mai „solid”, să nu facă obiectul să arate ca un teanc de hârtii.



Fiecare strat este desenat o singură dată.



Același strat este desenat de mai multe ori.

Platforma

Proiectul a fost creat în programul Eclipse, un mediu de dezvoltare open-source specializat pentru scrierea aplicațiilor în limbajul Java.

Codul sursă

Codul sursă poate fi găsit pe GitHub: <https://github.com/CezarPetreanu/STIVA-MINI>

IV. CONCLUZII

Proiectul a reușit să se apropie foarte mult de obiectivele propuse în temă: rezultatul este un program funcțional, permite utilizatorilor să deseneze, să salveze și să deschidă proiecte de desen, să vizualizeze un obiect 3D format prin suprapunerea straturilor desenate și să exporteze o imagine ca să fie folosită în proiectele lor.

Cu toate acestea, din cauza dificultăților apărute în realizarea lui și problemelor legate de organizarea timpului, proiectul încă mai poate beneficia de anumite îmbunătățiri înainte să fie lansat publicului.

Provocări și dificultăți

Una dintre marile provocări ale proiectului a fost refacerea interfeței grafice. Versiunea beta avea o interfață minimală, conținând doar strictul necesar pentru desenarea și vizualizarea modelului. Versiunea finală aduce o nouă interfață mai completă și mai prietenoasă cu utilizatorul.

O altă provocare a fost implementarea sistemului de straturi, astfel încât să permită afișarea lor pe ecran în momentul selectării lor și efectuarea operațiilor de adăugare, ștergere și mutare.

Au fost probleme la implementarea funcționalității de salvare și încărcare a fișierelor proiectului de desen, în principal din cauză că anumite elemente de interfață grafică oferite de AWT și Swing nu pot fi serializate. Problema a fost rezolvată până la urmă, însă a durat o vreme să se găsească o soluție.

Produse și servicii similare

În principal, Stiva Mini este un editor de imagini, specializat pe grafici de dimensiuni mici (pixel art) dar care permite utilizatorilor, desenatorilor și dezvoltatorilor de jocuri video, să creeze și să vizualizeze modele făcute după tehnica „sprite stacking”.

Programele enumerate prezintă aspecte similare și diferite față de program:

- Majoritatea programelor de creat și editat imagini pot fi folosite pentru crearea graficilor pixel art și conțin mai multe unelte și funcționalități. Ele nu pot simula însă funcționalitatea programului Stiva Mini de a vizualiza modelul final. În mod normal, utilizatorul ar trebui să introducă straturile modelului în programul de creare jocuri, să implementeze afișarea lor corespunzătoare și abia atunci să vadă rezultatul final.

- Aseprite este un program specializat pentru pixel art, popular în rândul dezvoltatorilor de jocuri. Pe lângă funcționalitățile de bază ale programelor de editat imagini, Aseprite permite crearea și vizualizarea animațiilor cadru cu cadru și oferă unelte implementate special pentru lucrul cu pixeli. Programul nu este specializat pentru tehnica „sprite stacking” asemeni Stiva Mini, dar utilizatorii ar prefera să rămână pe metoda tradițională de desenat grafici pentru jocuri, ceea ce ar putea fi un obstacol major pentru Stiva Mini.
- SpriteStack este o unealtă online încă în dezvoltare specializată pentru tehnica „sprite stacking” și momentan cel mai mare competitor pentru Stiva Mini. Pe lângă funcționalitatea de a desena după tehnica menționată, SpriteStack poate să creeze animații (funcționalitate experimentală), să aplice umbre și lumini modelelor și să permită utilizatorului să importe modele 3D ca să fie procesate apoi în grafici 2D. Momentan, versiunea prototip poate fi cumpărată pe Steam și Itch, iar versiunea alfa, lansată în 2021 este valabilă numai persoanelor abonate pe Patreon. Conform dezvoltatorului, cunoscut online drept Rezoner, progresul a stagnat drastic în 2020. (spritestack.io)

Posibile direcții de dezvoltare ale proiectului

Proiectul poate beneficia de o nouă interfață grafică modernă, receptivă (responsive user interface).

De asemenea, se dorește implementarea unor funcționalități care ar fi fost de preferat să existe în versiunea finală, precum încărcarea unei imagini direct în program pentru a o modifica.

Nu în ultimul rând, se va încerca găsirea și implementarea unor metode mai rapide de a desena modelele, deoarece performanța scăzută a algoritmilor prezenți prezintă o problemă pentru experiența utilizatorului.

V. BIBLIOGRAFIE

Stroustrup, Bjarne; (Mai, 2013). „The C++ Programming Language (4th Edition)”

Gosling, James; McGilton, Henry (Mai, 1996). „The Java Language Environment”

spritestack.io