

LỜI NÓI ĐẦU

Đồ họa máy tính là một lĩnh vực của khoa học máy tính nghiên cứu về cơ sở toán học, các thuật toán cũng như các kỹ thuật để cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính có liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích, hình học họa hình, quang học,... và kỹ thuật máy tính, đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị xuất, nhập, các vi mạch đồ họa...).

Theo nghĩa rộng hơn, đồ họa máy tính là phương pháp và công nghệ dùng trong việc chuyển đổi qua lại giữa dữ liệu và hình ảnh trên màn hình bằng máy tính.

Để phục vụ cho học tập và tìm hiểu rõ hơn về bộ môn đồ họa máy, chúng em đã phân tích thiết kế và đã xây dựng hai ứng dụng sử dụng những kiến thức đã học từ đồ họa máy tính.

Trong quá trình thực hiện chúng em có tham khảo một số tài liệu tổng hợp từ internet, một số tài liệu giảng dạy của thầy cô trong trường và một số tài liệu nước ngoài nên một số thuật ngữ tin học không thể thay thế bằng tiếng Việt hoặc dịch ở mức độ tương đối.

Trong quá trình thực hiện chúng em có nhận được sự hỗ trợ, hướng dẫn tận tình của cô Huỳnh Thị Nhật Hằng. Chúng em xin chân thành cảm ơn và hy vọng sản phẩm của chúng em sẽ được sự ủng hộ nhiệt tình từ thầy, cô và các bạn sinh viên trường Đại Học Tiền Giang.

Trong phạm vi hạn hẹp của học phần đồ án chuyên ngành và thời gian ngắn, chúng em không thể thực hiện được đầy đủ các tính năng của một ứng dụng quản lý cửa hàng cần phải có và mặc dù đã rất cố gắng trong quá trình thực hiện, song không tránh khỏi những thiếu sót. Rất mong nhận được sự đóng góp từ quý thầy cô và các bạn sinh viên để sản phẩm của chúng em được hoàn chỉnh hơn.

Mọi ý kiến góp ý xin liên hệ trực tiếp đến sinh viên Hồ Đoàn Thanh Ngoãn và sinh viên Ngô Thành An thuộc lớp Công Nghệ Thông Tin 16, Khoa Công Nghệ Thông Tin – Trường Đại Học Tiền Giang.

Chân thành cảm ơn.

Sinh viên thực hiện
Hồ Đoàn Thanh Ngoãn

Ngô Thành An

MỤC LỤC

A. TỔNG QUAN

I. Giới thiệu về môn đồ họa máy tính?.....	1
II.Mục đích môn đồ họa máy tính và hướng ứng dụng của môn học	1
1.Mục đích	1
2.Hướng ứng dụng của môn học.....	1
III.Mục tiêu của đề án	1
IV.Phạm vi đề án.....	2

B. PHẦN NỘI DUNG

3

I.ĐỒ HỌA 2D

3

1.Tổng quan về Java 2D API.....	3
2.Các vấn đề cơ bản của Java 2D API	3
3.Chương trình rắn sẵn mỗi trong hệ tọa độ 2D	4
3.1.Cài đặt và demo	4
3.2.Code và giải thuật	7

II.ĐỒ HỌA 3D.....

10

1.Tổng quan về Java 3D API.....	10
2.Các vấn đề cơ bản của Java 3D API	10
3.Chương trình cục xúc xác 3D	11
3.1.Cài đặt và demo	11
3.2.Thuật toán và Code	12

C. PHẦN KẾT LUẬN.....

16

D. TÀI LIỆU THAM KHẢO

17

DANH MỤC ẢNH

Hình 1. Giao diện chạy chương trình rắn sẵn mồi.....	5
Hình 2. Giao diện khi người dùng bấm vào các phím điều khiển con rắn.....	3
Hình 3. Sau khi ăn rắn tự động tăng lên một đơn vị thân	6
Hình 4. Khi va chạm, hoặc cắn đuôi rắn sẽ chết và kết thúc game	6
Hình 5. Giao diện khi mở chạy chương trình xúc xắc 3D.....	11
Hình 6. Ấn vào các nút w,s,a,d để xoay xúc xắc	12

A. TỔNG QUAN

I. Giới thiệu về môn đồ họa máy tính ?

- Đồ họa máy tính là 1 ngành khoa học Tin học chuyên nghiên cứu về các phương pháp và kỹ thuật để có thể mô tả và thao tác trên các đối tượng của thế giới thực bằng máy tính.

- Về bản chất: đó là 1 quá trình xây dựng và phát triển các công cụ trên cả 2 lĩnh vực phần cứng và phần mềm hỗ trợ cho các lập trình viên thiết kế các chương trình có khả năng đồ họa cao.

- Với việc mô tả dữ liệu thông qua các hình ảnh và màu sắc đa dạng của nó, các chương trình đồ họa thường thu hút người sử dụng bởi tính thân thiện, dễ dùng,... kích thích khả năng sáng tạo và nâng cao năng suất làm việc. Do vậy, đồ họa máy tính được ứng dụng trong nhiều lĩnh vực: giáo dục, thương mại.

II. Mục đích môn đồ họa máy tính và hướng ứng dụng của môn học.

1. Mục đích.

- Sau khi hoàn thành học phần, sinh viên có khả năng sau:
 - + Về kiến thức: Sinh viên hiểu biết cơ bản về các khái niệm, nguyên lý, mô hình và các thuật toán cơ sở của đồ họa máy tính hai chiều và ba chiều.
 - + Về kỹ năng : Sinh viên có khả năng lập trình ứng dụng đồ họa hai chiều và ba chiều dùng các thư viện đồ họa Java 2D và 3D.
 - + Về thái độ: Có sự quan tâm, hứng thú đối với các chủ đề của học phần cũng như có ý thức vận dụng kiến thức của học phần vào thực tiễn nghề nghiệp.

2. Hướng ứng dụng của môn học.

- Hỗ trợ thiết kế
- Tạo giao diện người dùng
- Biểu diễn thông tin
- Giải trí
- Tự động hóa và điều khiển
- Mô phỏng
- Bản đồ
- Giáo dục
- Biểu diễn số liệu khoa học

III. Mục tiêu của đồ án.

- Mục tiêu nhằm xây dựng một chương trình đồ họa 2D và một chương trình đồ họa 3D dựa trên kiến thức của môn học Đồ Họa Máy Tính .

- Công nghệ xây dựng chương trình:

- + Eclipse IDE for Java Developers 2020-06.
- + Netbean 8.2 .
- + Môi trường windows 10.
- + Blender 3D.

IV. Phạm vi đồ án

- Quy mô:

Đồ án có quy mô được giới hạn trong phạm vi của môn Đồ Họa Máy Tính.

- Thời gian:

Thiết kế và xây dựng trong 7 tuần kể từ tuần thứ 7 đến tuần 15 của học kỳ 2 năm học 2019-2020.

+ Tìm hiểu tài liệu :

Từ ngày 01 tháng 05 năm 2020 tới ngày 10 tháng 05 năm 2020.

+ Xây dựng chương trình:

Từ ngày 10 tháng 05 năm 2020 tới ngày 17 tháng 07 năm 2020.

+ Viết báo cáo:

Từ ngày 17 tháng 07 năm 2020 tới ngày 23 tháng 07 năm 2020.

+ Thử nghiệm chương trình và báo cáo:

Từ ngày 23 tháng 04 năm 2020 tới ngày 25 tháng 07 năm 2020.

B. PHẦN NỘI DUNG

I. ĐỒ HỌA 2D

1. Tổng quan về Java 2D API

- Java 2D API là API để vẽ đồ họa 2 chiều sử dụng ngôn ngữ Java.
- Java 2D API cải tiến khả năng về graphics, text và image của Abstract Windowing Toolkit (AWT). AWT là bộ tool đầu tiên giành cho việc tạo User Interface (UI) và graphics trong Java. Vì mục đích tương thích, Java 2D là công cụ AWT.
- Java 2D là công nghệ mạnh mẽ. Nó có thể được sử dụng để tạo rich UI, game, animation, ứng dụng multimedia và hàng loạt các hiệu ứng đặc biệt khác.
- Java 2D™ API tăng cường về khả năng đồ họa, văn bản và ảnh của Abstract Windowing Toolkit (AWT), giúp phát triển về giao diện người sử dụng và ứng dụng về JAVA trong một số lĩnh vực mới. Cùng với khả năng mạnh về đồ họa, phông chữ và ảnh trong các API, thì Java 2D API còn hỗ trợ những thuộc tính và thành phần màu sắc đã được nâng cao, và thành công trong việc biểu diễn các bề mặt hình học tùy ý và một kiểu tô trát chuẩn cho máy in và thiết bị hiển thị. Java 2D API cũng cho phép tạo ra các thư viện đồ họa mở rộng, như các thư viện của CAD-CAM và các thư viện tạo hiệu ứng đặc biệt cho ảnh và đồ họa, cũng như việc tạo ra các bộ lọc đọc/viết file ảnh và đồ họa. Khi được sử dụng kết hợp với Java Media Framework Java Media APIs khác, thì Java 2D APIs có thể được sử dụng để tạo ra và hiển thị hiện thực ảo và các dạng đa phương tiện khác. Java Animation và Java Media Framework APIs dựa trên Java 2D API để hỗ trợ việc tô trát(rendering).

2. Các vấn đề cơ bản của Java 2D API

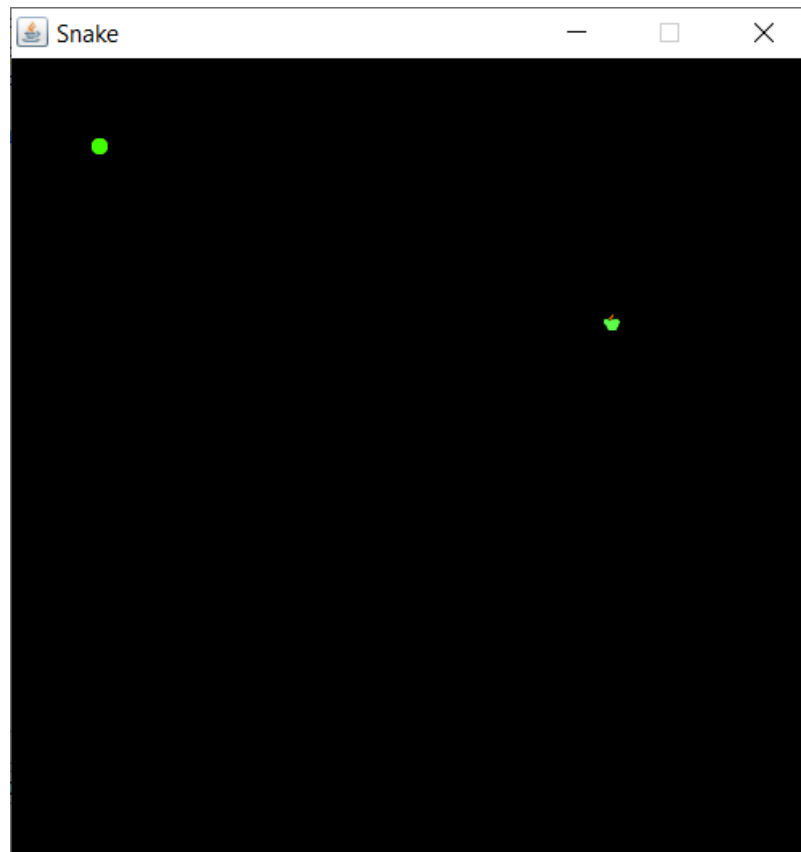
- Java 2D API cung cấp chức năng sau:
 - + Mô hình render đồng nhất cho các thiết bị hiển thị và máy in.
 - + Detect điểm trên hình vẽ, text và image.
 - + Mô hình kết hợp
 - + Support màu
 - + Kiểm soát chất lượng của render
- Enhanced Graphics, Text, and imaging: Nếu các phiên bản trước của AWT chỉ cung cấp các gói tô trát(rendering) đơn giản chỉ phù hợp cho việc rendering các trang

HTML đơn giản, mà không đáp ứng đủ cho các dạng đồ họa, văn bản và ảnh phức tạp. Thì Java 2D™ API cung cấp gói tô trát đầy đủ các tính năng bằng cách mở rộng AWT để hỗ trợ cho đồ họa và các thao tác rendering. Ví dụ như thông qua các lớp đồ họa cho phép vẽ hình chữ nhật, hình oval, và các đa giác. Đồ họa 2D tăng cường về khái niệm của phép tô trát hình học bằng cách cung cấp một cơ chế cho phép rendering ảo của bất kì bề mặt hình học nào. Tương tự như vậy với Java 2D API bạn có thể vẽ các dạng đường với bất kì độ rộng và dạng hình học nào.

- Rendering Model: Kiểu tô trát đồ họa đơn giản không có gì thay đổi khi thêm vào Java 2D™ APIs. Để tô trát đồ họa thì phải thiết lập các thông số về đồ họa và gọi phương thức của đối tượng Graphics. Trong Java 2D API, lớp Graphics2D mở rộng lớp Graphics để hỗ trợ thêm nhiều thuộc tính đồ họa và cung cấp thêm các phương thức cho quá trình tô trát. The Java 2D API tự động cân chỉnh những sai khác trong các thiết bị tô trát và cung cấp một kiểu tô trát thống nhất cho các dạng thiết bị khác nhau. Tại tầng ứng dụng, quá trình tô trát là giống nhau cho dù thiết bị cuối đó là màn hình hay máy in. Với Java 2 SDK, version 1.3, Java 2D API còn hỗ trợ cho các môi trường đa màn hình (multi-screen environments).

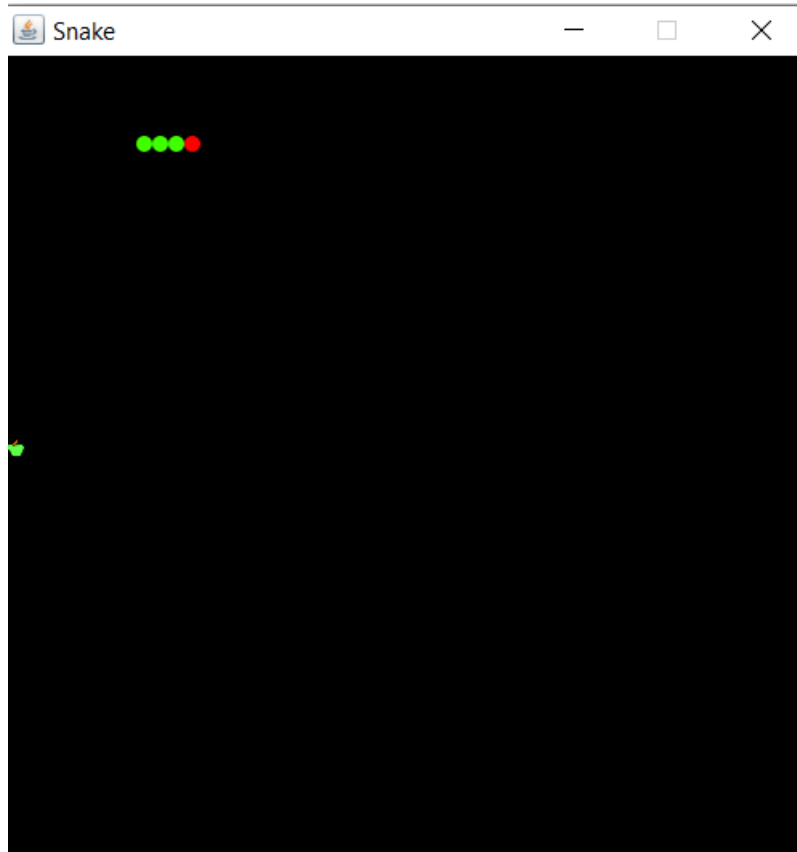
3. Chương trình rắn sẵn môi trong hệ tọa độ 2D

3.1. Cài đặt và demo



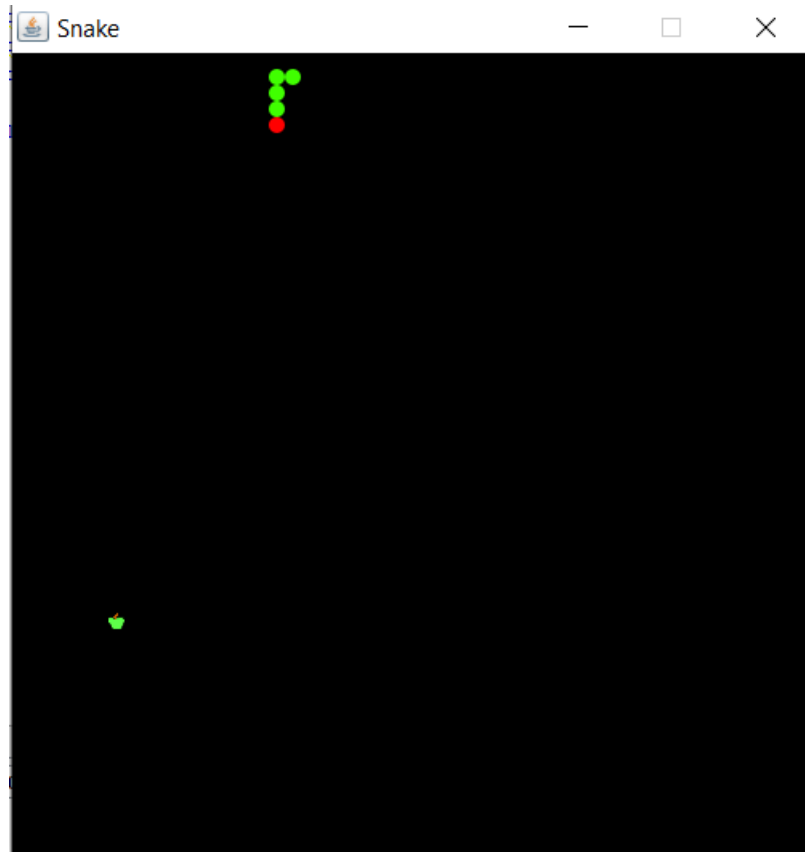
Hình 1. Giao diện chạy chương trình rắn sẵn môi.

- Khi chạy chương trình và người chơi bấm vào phím mũi tên bất kì để di chuyển con rắn đến vị trí trái táo.

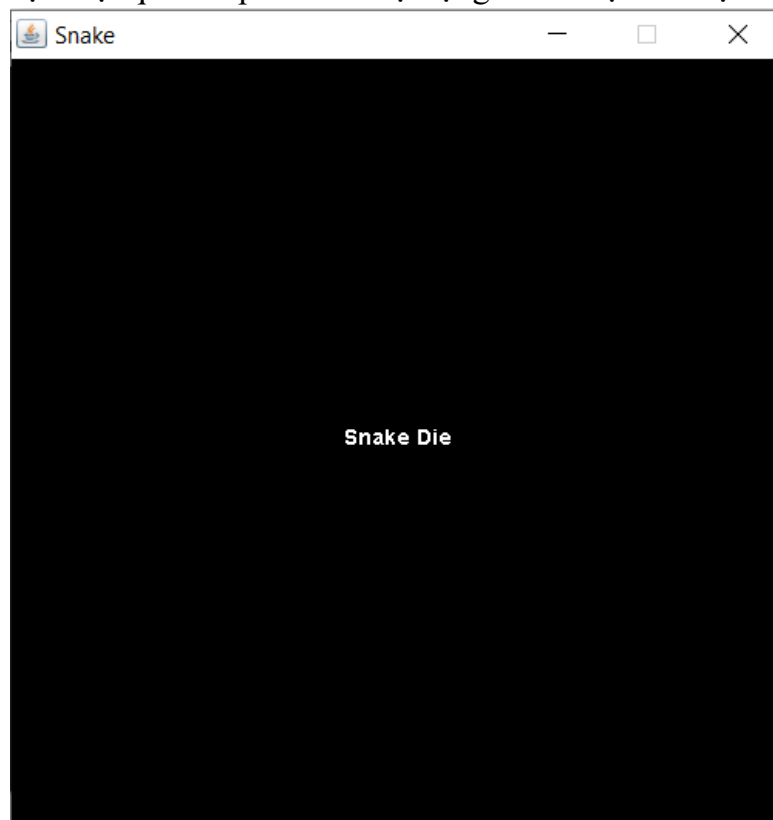


Hình 2. Giao diện khi người dùng bấm vào các phím điều khiển con rắn.

- Ban đầu con rắn sẽ được thiết lập với 1 chấm đầu và 3 chấm thân.



Hình 3. Sau khi ăn rắn tự động tăng lên một đơn vị thân
- Khi rắn ăn được một quả táo phần thân tự động thêm một đơn vị.



Hình 4. Khi va chạm, hoặc cắn đuôi rắn sẽ chết và kết thúc game.

- Chẳng may con rắn chạm vào khung chương trình hoặc chạm vào phần đuôi thì dừng chương trình và hiển thị Snake Die.

3.2. Code và giải thuật:

```
public class Board extends JPanel implements ActionListener {

    //Thiết lập kích thước các đối tượng có trong game
    private final int B_WIDTH = 500;//kích thước của cửa sổ
    private final int B_HEIGHT = 500;
    private final int DOT_SIZE = 10;//khoảng cách giữa các dot
    private final int ALL_DOTS = 900;//số lượng chấm của Rắn (900 = (300*300)/(10*10))
    private final int RAND_POS = 40;//khoảng cách giữa rắn và táo
    private final int DELAY = 100;//tốc độ rắn chạy giảm = nhanh

    private final int x[] = new int[ALL_DOTS];// lưu trữ tọa độ phần thân của con rắn
    private final int y[] = new int[ALL_DOTS];

    private int dots;//tạo biến chứa phần thân
    private int apple_x;//biến vị trí xuất hiện táo
    private int apple_y;

    //Tạo các biến để điều khiển: trên, dưới, phải ,trái
    private boolean trai = false;
    private boolean phai = false;
    private boolean tren = false;
    private boolean duoi = false;
    private boolean inGame = true;

    private Timer timer;// thời gian
    private Image than;// phần thân
    private Image tao;//táo
    private Image dau;//đầu
```

- Thiết lập những đối tượng cần sử dụng trong chương trình.

```
private void loadImages() {

    ImageIcon iid = new ImageIcon("src/resources/dot.png");
    than = iid.getImage();

    ImageIcon iia = new ImageIcon("src/resources/apple.png");
    tao = iia.getImage();

    ImageIcon iih = new ImageIcon("src/resources/head.png");
    dau = iih.getImage();
}
```

- Load hình ảnh vào trong chương trình gồm: đầu, thân và táo.

```
private void gameOver(Graphics g) {

    String msg = "Snake Die";
    Font small = new Font("Helvetica", Font.BOLD, 14);
    FontMetrics metr = getFontMetrics(small);

    g.setColor(Color.white);
    g.setFont(small);
    g.drawString(msg, (B_WIDTH - metr.stringWidth(msg)) / 2, B_HEIGHT / 2);
}
```

- Hàm hiện game Over.

```
private void checkApple() {
    if ((x[0] == apple_x) && (y[0] == apple_y)) {
        dots++;
        locateApple();
    }
}
```

- Nếu rắn ăn được quả táo, sẽ tăng số lượng chấm thân của con rắn. Gọi phương thức locationApple () để định vị vị trí ngẫu nhiên của quả apple mới.

```
private void move() {
    for (int z = dots; z > 0; z--) {
        x[z] = x[(z - 1)];
        y[z] = y[(z - 1)];
    }

    if (traai) {
        x[0] -= DOT_SIZE;
    }

    if (phai) {
        x[0] += DOT_SIZE;
    }

    if (tren) {
        y[0] -= DOT_SIZE;
    }

    if (duoi) {
        y[0] += DOT_SIZE;
    }
}
```

- Thuật toán điều khiển con rắn. Có thể điều khiển các khớp của con rắn cùng di chuyển đồng bộ, nối tiếp nhau.

```

private void checkCollision() {

    for (int z = dots; z > 0; z--) {

        if ((z > 4) && (x[0] == x[z]) && (y[0] == y[z])) {
            inGame = false;
        }

    }

    if (y[0] >= B_HEIGHT) {
        inGame = false;
    }

    if (y[0] < 0) {
        inGame = false;
    }

    if (x[0] >= B_WIDTH) {
        inGame = false;
    }

    if (x[0] < 0) {
        inGame = false;
    }

    if (!inGame) {
        timer.stop();
    }

}

```

- Thuật toán kiểm tra va chạm.

```

private void locateApple() { //lấy số ngẫu nhiên

    int r = (int) (Math.random() * RAND_POS);
    apple_x = ((r * DOT_SIZE));

    r = (int) (Math.random() * RAND_POS);
    apple_y = ((r * DOT_SIZE));

}

@Override
public void actionPerformed(ActionEvent e) {

    if (inGame) {

        checkApple();
        checkCollision();
        move();

    }

    repaint();

}

```

- Thuật toán định vị vị trí xuất hiện của quả táo.

```

private class TAdapter extends KeyAdapter {

    @Override
    public void keyPressed(KeyEvent e) {

        int key = e.getKeyCode();

        if ((key == KeyEvent.VK_LEFT) && (!phai)) {
            trai = true;
            tren = false;
            duoi = false;
        }

        if ((key == KeyEvent.VK_RIGHT) && (!trai)) {
            phai = true;
            tren = false;
            duoi = false;
        }

        if ((key == KeyEvent.VK_UP) && (!duoi)) {
            tren = true;
            phai = false;
            trai = false;
        }

        if ((key == KeyEvent.VK_DOWN) && (!tren)) {
            duoi = true;
            phai = false;
            trai = false;
        }
    }
}

```

- Thiết lập các phím chức năng.

II. ĐỒ HỌA 3D

1. Tổng quan về Java 3D API

- Java 3D API được thừa kế từ các lớp của java cho phép tạo các giao diện phức tạp như hệ thống âm thanh và hình ảnh 3 chiều. Người lập trình có thể sử dụng nó để xây dựng điều khiển các đối tượng hình học 3 chiều. Những đối tượng này cư trú trên một môi trường ảo sau đó mới được dựng hình lại. Các hàm API được thiết kế rất linh động cho phép tạo một trường ảo đó chính xác với rất nhiều dạng kích cỡ từ lớn đến bé.

- Dù có nhiều chức năng như vậy nhưng các hàm API vẫn có thể sử dụng rõ ràng. Các chi tiết được dựng hình tự động. Sử dụng lợi thế của các luồng trong java, việc dựng hình trong Java 3D cho phép làm song song, đồng thời cũng cho phép tối ưu hóa việc dựng hình. Chương trình Java 3D tạo các instances của các đối tượng và đưa chúng vào cơ sở dữ liệu đồ thị khung cảnh. Đồ thị khung cảnh là một sắp xếp các đối tượng Java 3D trên một cấu trúc cây cho phép chỉ ra nội dung của một thế giới ảo và chúng có thể được dựng hình như thế nào.

- Chương trình Java 3D có thể được viết để chạy độc lập hay như là một applet trên các trình duyệt hỗ trợ Java 3D hoặc cả 2.

2. Các vấn đề cơ bản của Java 3D API

- Java 3D API định nghĩa hơn 100 lớp trong gói javax.media.j3d những lớp này được coi như là những lớp cơ bản nhất của Java 3D.

- Có đến hàng trăm trường và phương thức trong các lớp của Java 3D API tuy nhiên một môi trường ảo đơn giản chỉ cần dùng một vài lớp. Chương này mô tả tập nhỏ nhất các đối tượng và tương tác giữa chúng để dựng nên một môi trường ảo.

- Chương này còn bao gồm ví dụ HelloJava3D hiển thị một khối lập phương quay. Ví dụ này được phát triển theo từng bước lần lượt thể hiện các phần xử lý trong lập trình Java 3D.

- Ngoài gói nhân Java 3D chúng ta còn sử dụng thêm một gói nữa là com.sun.j3d.utils thường gọi là các lớp tiện dụng, chúng bổ sung thêm những tính năng hiệu quả và rất mạnh cho nhân.

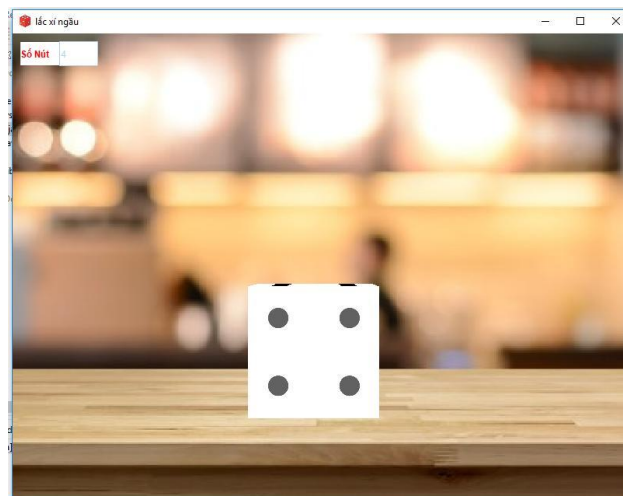
- Lớp utility gồm 4 mục: content loader(nạp nội dung), hỗ trợ xây dựng đồ thị khung cảnh, các lớp hình học, và các công cụ tiện ích.

- Tất nhiên là ngoài gói nhân và gói tiện ích thì mọi chương trình java đều sử dụng các lớp từ gói java.awt và javax.vecmath. Gói java.awt định nghĩa các lớp AWT cho phép tạo cửa sổ cho việc dựng hình. Gói javax.vecmath định nghĩa các lớp cho điểm, vector ma trận và những đối tượng toán học khác.

- Trong phần còn lại của giáo trình này từ visual object(đối tượng quan sát) được sử dụng thay cho “một đối tượng trong đồ thị khung cảnh”. (chẳng hạn: một hình lập phương hay hình cầu). Từ object được sử dụng cho một instance của một lớp. Content tương ứng với những đối tượng quan sát trong một đồ thị khung cảnh.

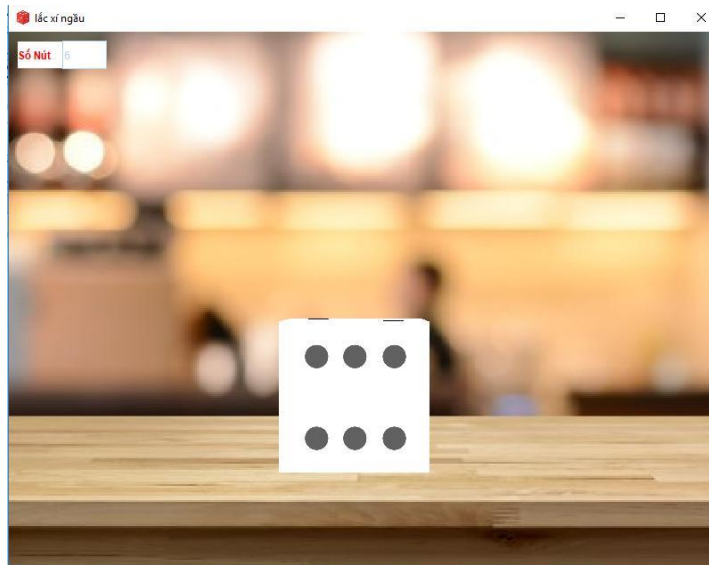
3. Chương trình xúc xắc 3D

3.1. Cài đặt và demo



Hình 5. Giao diện khi mở chạy chương trình xúc xắc 3D

- Giao diện sau khi chạy ứng dụng.



Hình 6. Ấn vào các nút w,s,a,d để xoay xúc xắc.

- Nếu ấn nút w,s,a,d dùng để xoay lên xuống trái phải. Nếu ấn 2,8 sẽ di chuyển vị trí của xí ngẫu lên xuống so với mặt bàn.
- Textfield dùng để hiển thị số nút hiện trên cục xí ngẫu.

3.2. Thuật toán và Code:

- Hàm di chuyển mặt xí ngẫu lên trên, xuống dưới.

```
public void DiLen() {

    for(int i=0; i<=3;i++) { // xoay lên 1 mặt
        try {
            Thread.sleep(20); // tốc độ xoay nhanh chậm của xí
            ngẫu
        } catch (InterruptedException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
        t3dstep.set(new Vector3d(0.0, 0.1, 0.0));
        tg.getTransform(t3d);
        t3d.mul(t3dstep);
        tg.setTransform(t3d);
    }
}
```

```

    }

    }
    public void DiXuong() {

        for(int i=0; i<=3;i++) { // xoay xuống 1 mặt
            try {
                Thread.sleep(20); // tốc độ xoay nhanh chậm
của xí ngẫu

            } catch (InterruptedException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
            t3dstep.set(new Vector3d(0.0, -0.1, 0.0));
            tg.getTransform(t3d);
            t3d.mul(t3dstep);
            tg.setTransform(t3d);
        }
    }
}

```

- Hàm xoay phải xoay trái, phải.

```

public void XoayPhai() {

        for(int i=0; i<=15;i++) {
            try {
                Thread.sleep(20); // tốc độ xoay nhanh
chậm của xí ngẫu

            } catch (InterruptedException e1) {
                e1.printStackTrace();
            }
        }
    }
}

```



```

                                t3dstep.rotY(Math.PI / 32);
                                tg.getTransform(t3d);
                                t3d.get(matrix);
                                t3d.setTranslation(new Vector3d(0.0, 0.0,
0.0));

                                t3d.mul(t3dstep);
                                t3d.setTranslation(new Vector3d(matrix.m03,
matrix.m13, matrix.m23));

                                tg.setTransform(t3d);
                                }
                                if((Y==1)||(Y==5)) {
                                    X = X + 1;
                                    if(X>6) {
                                        X=3;
                                    }
                                }
                                else if ((Y==0)||(Y==2)||(Y==4)||(Y==6)) {
                                    X=X+0;
                                }
                                else {
                                    X = X - 1;
                                    if(X<0) {
                                        X=3;
                                    }
                                }

                                }

                                public void XoayTrai() {

```

```

for(int i=0; i<=15;i++) {
    try {
        Thread.sleep(20); // tốc độ xoay nhanh
        chậm của xí ngẫu
    } catch (InterruptedException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

    t3dstep.rotY(-Math.PI / 32);
    tg.getTransform(t3d);
    t3d.get(matrix);
    t3d.setTranslation(new Vector3d(0.0, 0.0,
0.0));

    t3d.mul(t3dstep);
    t3d.setTranslation(new Vector3d(matrix.m03,
matrix.m13, matrix.m23));

    tg.setTransform(t3d);
}
if((Y==1)||(Y==5)) {
    X = X - 1;
    if(X<0) {
        X=3;
    }
}
else if ((Y==0)||(Y==2)||(Y==4)||(Y==6)) {
    X=X+0;
}
else {

```

```

X = X + 1;
if(X>6) {
    X=3;
}
}
}

```

C. PHẦN KẾT LUẬN

- Những vấn đề đạt được :
 - + Hệ thống giải quyết được các mục tiêu đã đề ra ở mục 3 phần A.
 - + Hệ thống được hoàn thành đúng thời hạn và đúng với những mục tiêu đã đề ra với phạm vi dự án trong môn học.
- Những vấn đề chưa đạt:
 - + Hệ thống chưa đáp ứng tốt khi nhu cầu sử dụng ở quy mô lớn với lượng tin tức lớn và đa dạng.
 - + Do hạn chế về thời gian và qui mô của dự án nên một số chức năng chưa được hoàn thiện như : Tăng điểm, tính tổng điểm cho con rắn và xúc xắc.
 - + Chưa thể trở thành một trò chơi hoàn thiện.
- Hướng phát triển của đề tài:
 - + Hoàn thiện những tính năng còn thiếu.
 - + Nâng cấp hệ thống máy chủ để đảm bảo trải nghiệm cho người dùng.
 - + Cập nhật thêm những tính năng mới cho hai ứng dụng.
 - + Thay đổi giao diện thu hút và gây hứng thú cho người dùng.

D. TÀI LIỆU THAM KHẢO

[1][Online].Java2D, 3D là gì?: <http://www.zbook.vn/ebook/lap-trinh-do-hoa-tren-java-2d-3d-21327/>

[2][Online].Thamkhaocode: <https://www.w3schools.com/>.

[3][Huỳnh Thị Nhật Hằng].Giáo trình đồ họa máy tính : Trường Đại Học Tiền Giang – Khoa Công Nghệ Thông Tin.