

■ 修正前 (12 章 WalkingChat.java)

```

import java.applet.Applet;
import java.awt.*;
import java.awt.BorderLayout;
import java.awt.event.*;
import java.awt.AWTEvent;
import java.lang.Math;
import java.util.Enumeration;
import java.net.*;
import java.io.*;
import java.util.*;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.universe.SimpleUniverse;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.behaviors.mouse.*;

public class WalkingChat extends Applet implements ActionListener{
    public SimpleUniverse simpleUniverse;           //仮想空間
    public BranchGroup scene;                       //サブグラフのルート
    public TransformGroup eyeportTrans;            //視点位置変更用 TransformGroup
    public Transform3D eyeportTf3;                //視点位置変更用 Transform3D
    public TransformGroup eyerotTrans;             //アングル変更用 TransformGroup
    public Transform3D eyerotTf3;                 //アングル変更用 Transform3D
    Button leftBtn = new Button("←");              //アバター左回転用ボタン
    Button forwardBtn = new Button("↑");           //アバター前進用ボタン
    Button rightBtn = new Button("→");            //アバター右回転用ボタン
    Button backBtn = new Button("↓");             //アバター後進用ボタン
    Button sendBtn = new Button("send");          //メッセージ送信用ボタン
    TextField tf = new TextField(24);              //メッセージ入力用テキストフィールド
    Button enterBtn = new Button("enter");         //チャット参加用ボタン
    Button exitBtn = new Button("exit");          //チャット終了用ボタン
    Button eyeforwardBtn = new Button("eye ↑");   //視点位置前進用ボタン
    Button eyebackBtn = new Button("eye ↓");      //視点位置後進用ボタン
    Button highangleBtn = new Button("high angle"); //視点角度上昇用ボタン
    Button lowangleBtn = new Button("low angle"); //始点角度下降用ボタン
    Socket socket = null;                          //ソケット
    DataInputStream in = null;                      //データ受信用ストリーム
    DataOutputStream out = null;                    //データ送信用ストリーム
    BufferedReader intext = null;                   //文字列データ受信用ストリーム
    PrintWriter outtext = null;                    //文字列データ送信用ストリーム
    int id = -1;                                    //クライアントの識別番号
    boolean joinflag = false;                       //参加・不参加判別用フラグ
    public float eyeZ = -5.0f;                       //視点位置の Z 座標
    public double eyeangle = -Math.PI/6.0;          //視点の向き
    public float angle = 0.0f;                       //アバターの向きデータ送受信変数
    public float avatarX = 0.0f;                     //アバターの X 座標送受信変数
    public float avatarZ = 0.0f;                     //アバターの Z 座標送受信変数
    public String str;                               //メッセージ送受信変数
    public float going = 0.5f;                       //アバター前進用変数
    public float backing = 0.5f;                     //アバター後進用変数
    public float amountX = 0.0f;                     //アバターの X 軸方向の移動量
    public float amountZ = 0.0f;                     //アバターの Z 軸方向の移動量
    public ChatAvatar client1;                       //1 人目のクライアント
    public String message1 = "Hey!";                 //クライアント 1 のメッセージ
    public Text3D messageTxt3d1;                     //Text3D オブジェクト
    public TransformGroup avatarTrans1;              //アバター移動用 TransformGroup
    public Transform3D avatarTf31;                   //アバター移動用 Transform3D
    public Transform3D rotTf31;                      //アバターの向き変更用 Transform3D
    public float angle1 = 0.0f;                       //アバターの向きの角度
    public float avatarX1 = 0.0f;                     //アバターの X 座標
    public float avatarZ1 = 0.0f;                     //アバターの Z 座標
    public ChatAvatar client2;                       //2 人目のクライアント
    public String message2 = "Wow!";                 //クライアント 2 のメッセージ
    public Text3D messageTxt3d2;                     //Text3D オブジェクト
    public TransformGroup avatarTrans2;              //アバター移動用 TransformGroup
    public Transform3D avatarTf32;                   //アバター移動用 Transform3D
    public Transform3D rotTf32;                      //アバターの向き変更用 Transform3D
    public float angle2 = 0.0f;                       //アバターの向きの角度

```

```

public float avatarX2 = 0.0f; //アバターの X 座標
public float avatarZ2 = 0.0f; //アバターの Z 座標

public static void main(String[] args){
    new MainFrame(new WalkingChat(0,450,300));//コンストラクタ呼び出し
}

public WalkingChat(){
    joinChat(); //サーバに接続するメソッド
    setLayout(new BorderLayout()); //キャンパスのレイアウト
    Canvas3D canvas3D = new Canvas3D(null);//Canvas3D オブジェクト
    add("Center", canvas3D); //Canvas3D オブジェクトの配置
    Panel northPanel = new Panel(); //画面上部パネル
    northPanel.add(enterBtn); //チャット参加用ボタン設置
    northPanel.add(eyeforwardBtn); //視点位置前進用ボタン設置
    northPanel.add(eyebackBtn); //視点位置後進用ボタン設置
    northPanel.add(highangleBtn); //視点角度上昇用ボタン設置
    northPanel.add(lowangleBtn); //始点角度下降用ボタン設置
    northPanel.add(exitBtn); //チャット終了用ボタン設置
    add("North",northPanel); //パネルを画面上部に配置
    Panel southPanel= new Panel(); //画面下部パネル設置
    southPanel.add(leftBtn); //アバター左回転用ボタン設置
    southPanel.add(forwardBtn); //アバター前進用ボタン設置
    southPanel.add(rightBtn); //アバター右回転用ボタン設置
    southPanel.add(backBtn); //アバター後進用ボタン設置
    southPanel.add(tf); //メッセージ入力用テキストフィールド設置
    southPanel.add(sendBtn); //メッセージ送信用ボタン設置
    add("South",southPanel); //パネルを画面下部に配置
    enterBtn.addActionListener(this); //イベント処理機能付与
    leftBtn.addActionListener(this); //イベント処理機能付与
    forwardBtn.addActionListener(this); //イベント処理機能付与
    rightBtn.addActionListener(this); //イベント処理機能付与
    backBtn.addActionListener(this); //イベント処理機能付与
    sendBtn.addActionListener(this); //イベント処理機能付与
    exitBtn.addActionListener(this); //イベント処理機能付与
    eyeforwardBtn.addActionListener(this); //イベント処理機能付与
    eyebackBtn.addActionListener(this); //イベント処理機能付与
    highangleBtn.addActionListener(this); //イベント処理機能付与
    lowangleBtn.addActionListener(this); //イベント処理機能付与
    //仮想空間
    SimpleUniverse simpleUniverse = new SimpleUniverse(canvas3D);
    eyeportTrans = new TransformGroup();//視点位置変更用 TransformGroup
    //許可ビット設定
    eyeportTrans.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    eyeportTf3 = new Transform3D(); //視点位置変更用 Transform3D
    eyeportTf3.set(new Vector3d(0.0,6.0,eyeZ)); //視点位置の初期座標
    eyerotTf3 = new Transform3D(); //視点角度変更用 Transform3D
    eyerotTf3.rotX(eyeangle); //X 軸を中心に視点角度変更
    eyeportTf3.mul(eyerotTf3); //2 つの Transform3D を掛算
    //視点のデフォルト座標
    eyeportTrans =
        simpleUniverse.getViewingPlatform().getViewPlatformTransform();
    eyeportTrans.setTransform(eyeportTf3);
    scene = createSceneGraph(); //シーングラフ構築用メソッド
    scene.setCapability(BranchGroup.ALLOW_BOUNDS_READ);//許可ビット
    simpleUniverse.addBranchGraph(scene); //サブグラフの登録
}

```

■ 修正後 (12 章 WalkingChat.java)

```

import java.applet.Applet;
import java.awt.*;
import java.awt.BorderLayout;
import java.awt.event.*;
import java.awt.AWTEvent;
import java.awt.GraphicsConfiguration; //←追加
import java.lang.Math;
import java.util.Enumeration;
import java.net.*;
import java.io.*;
import java.util.*;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.universe.SimpleUniverse;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.behaviors.mouse.*;

public class WalkingChat extends Applet implements ActionListener{
    public SimpleUniverse simpleUniverse; //仮想空間
    public BranchGroup scene; //サブグラフのルート
    public TransformGroup eyeportTrans; //視点位置変更用 TransformGroup
    public Transform3D eyeportTf3; //視点位置変更用 Transform3D
    public TransformGroup eyerotTrans; //アングル変更用 TransformGroup
    public Transform3D eyerotTf3; //アングル変更用 Transform3D
    GraphicsConfiguration config = SimpleUniverse.getPreferredConfiguration(); //←追加
    Button leftBtn = new Button("←"); //アバター左回転用ボタン
    Button forwardBtn = new Button("↑"); //アバター前進用ボタン
    Button rightBtn = new Button("→"); //アバター右回転用ボタン
    Button backBtn = new Button("↓"); //アバター後進用ボタン
    Button sendBtn = new Button("send"); //メッセージ送信用ボタン
    TextField tf = new TextField(24); //メッセージ入力用テキストフィールド
    Button enterBtn = new Button("enter"); //チャット参加用ボタン
    Button exitBtn = new Button("exit"); //チャット終了用ボタン
    Button eyeforwardBtn = new Button("eye ↑"); //視点位置前進用ボタン
    Button eyebackBtn = new Button("eye ↓"); //視点位置後進用ボタン
    Button highangleBtn = new Button("high angle"); //視点角度上昇用ボタン
    Button lowangleBtn = new Button("low angle"); //視点角度下降用ボタン
    Socket socket = null; //ソケット
    DataInputStream in = null; //データ受信用ストリーム
    DataOutputStream out = null; //データ送信用ストリーム
    BufferedReader intext = null; //文字列データ受信用ストリーム
    PrintWriter outtext = null; //文字列データ送信用ストリーム
    int id = -1; //クライアントの識別番号
    boolean joinflag = false; //参加・不参加判別用フラグ
    public float eyeZ = -5.0f; //視点位置の Z 座標
    public double eyeangle = -Math.PI/6.0; //視点の向き
    public float angle = 0.0f; //アバターの向きデータ送受信変数
    public float avatarX = 0.0f; //アバターの X 座標送受信変数
    public float avatarZ = 0.0f; //アバターの Z 座標送受信変数
    public String str; //メッセージ送受信変数
    public float going = 0.5f; //アバター前進用変数
    public float backing = 0.5f; //アバター後進用変数
    public float amountX = 0.0f; //アバターの X 軸方向の移動量
    public float amountZ = 0.0f; //アバターの Z 軸方向の移動量
    public ChatAvatar client1; //1 人目のクライアント
    public String message1 = "Hey!"; //クライアント 1 のメッセージ
    public Text3D messageTxt3d1; //Text3D オブジェクト
    public TransformGroup avatarTrans1; //アバター移動用 TransformGroup
    public Transform3D avatarTf31; //アバター移動用 Transform3D
    public Transform3D rotTf31; //アバターの向き変更用 Transform3D
    public float angle1 = 0.0f; //アバターの向きの角度
    public float avatarX1 = 0.0f; //アバターの X 座標
    public float avatarZ1 = 0.0f; //アバターの Z 座標
    public ChatAvatar client2; //2 人目のクライアント
    public String message2 = "Wow!"; //クライアント 2 のメッセージ
    public Text3D messageTxt3d2; //Text3D オブジェクト

```

```

public TransformGroup avatarTrans2; //アバター移動用 TransformGroup
public Transform3D avatarTf32; //アバター移動用 Transform3D
public Transform3D rotTf32; //アバターの向き変更用 Transform3D
public float angle2 = 0.0f; //アバターの向きの角度
public float avatarX2 = 0.0f; //アバターの X 座標
public float avatarZ2 = 0.0f; //アバターの Z 座標

public static void main(String[] args){
    new MainFrame(new WalkingChat(0,450,300)); //コンストラクタ呼び出し
}

public WalkingChat(){
    joinChat(); //サーバに接続するメソッド
    setLayout(new BorderLayout()); //キャンパスのレイアウト
    Canvas3D canvas3D = new Canvas3D(config); //←変更
    add("Center", canvas3D); //Canvas3D オブジェクトの配置
    Panel northPanel = new Panel(); //画面上部パネル
    northPanel.add(enterBtn); //チャット参加用ボタン設置
    northPanel.add(eyeforwardBtn); //視点位置前進用ボタン設置
    northPanel.add(eyebackBtn); //視点位置後進用ボタン設置
    northPanel.add(highangleBtn); //視点角度上昇用ボタン設置
    northPanel.add(lowangleBtn); //始点角度下降用ボタン設置
    northPanel.add(exitBtn); //チャット終了用ボタン設置
    add("North",northPanel); //パネルを画面上部に配置
    Panel southPanel= new Panel(); //画面下部パネル設置
    southPanel.add(leftBtn); //アバター左回転用ボタン設置
    southPanel.add(forwardBtn); //アバター前進用ボタン設置
    southPanel.add(rightBtn); //アバター右回転用ボタン設置
    southPanel.add(backBtn); //アバター後進用ボタン設置
    southPanel.add(tf); //メッセージ入力用テキストフィールド設置
    southPanel.add(sendBtn); //メッセージ送信用ボタン設置
    add("South",southPanel); //パネルを画面下部に配置
    enterBtn.addActionListener(this); //イベント処理機能付与
    leftBtn.addActionListener(this); //イベント処理機能付与
    forwardBtn.addActionListener(this); //イベント処理機能付与
    rightBtn.addActionListener(this); //イベント処理機能付与
    backBtn.addActionListener(this); //イベント処理機能付与
    sendBtn.addActionListener(this); //イベント処理機能付与
    exitBtn.addActionListener(this); //イベント処理機能付与
    eyeforwardBtn.addActionListener(this); //イベント処理機能付与
    eyebackBtn.addActionListener(this); //イベント処理機能付与
    highangleBtn.addActionListener(this); //イベント処理機能付与
    lowangleBtn.addActionListener(this); //イベント処理機能付与
    //仮想空間
    SimpleUniverse simpleUniverse = new SimpleUniverse(canvas3D);
    eyeportTrans = new TransformGroup(); //視点位置変更用 TransformGroup
    //許可ビット設定
    eyeportTrans.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    eyeportTf3 = new Transform3D(); //視点位置変更用 Transform3D
    eyeportTf3.set(new Vector3d(0.0,6.0,eyeZ)); //視点位置の初期座標
    eyerotTf3 = new Transform3D(); //視点角度変更用 Transform3D
    eyerotTf3.rotX(eyeangle); //X 軸を中心に視点角度変更
    eyeportTf3.mul(eyerotTf3); //2 つの Transform3D を掛算
    //視点のデフォルト座標
    eyeportTrans =
        simpleUniverse.getViewingPlatform().getViewPlatformTransform();
    eyeportTrans.setTransform(eyeportTf3);
    scene = createSceneGraph(); //シーングラフ構築用メソッド
    scene.setCapability(BranchGroup.ALLOW_BOUNDS_READ); //許可ビット
    simpleUniverse.addBranchGraph(scene); //サブグラフの登録
}

```

— 【以下略】 —