Kaba Kod

Ekrana oyuncu tipi seçimi için ekran getir.

class MainMenu extends javax.swing.JFrame {

public MainMenu() {

initComponents();

}

Seçim sonrası oyun alanına yönlendir.

Selection st = new Selection();

st.setSelectedGameType(2);

this.setVisible(false);

GameBoard GB = new GameBoard();

GB.setVisible(true);

Oyun alanı açılırken rastgele kartları dağıt.

CommonCards = new ArrayList();

List<Integer> cardIndex = new ArrayList();

cardIndex.add(0);cardIndex.add(1);cardIndex.add(2);cardIndex.add(3);cardIndex.add(4);

cardIndex.add(5);cardIndex.add(6);cardIndex.add(7);cardIndex.add(8);cardIndex.add(9);

Random rand = new Random();

int random1, random2, random3, random4, random5, random6;

random1 = rand.nextInt(10);

do{random2 = rand.nextInt(10);} while(random1 == random2);

do{random3 = rand.nextInt(10);} while(random3 == random1 || random3 == random2);

do{random4 = rand.nextInt(10);} while(random4 == random1 || random4 == random2 || random4 == random3);

do{random5 = rand.nextInt(10);} while(random5 == random1 || random5 == random2 || random5 == random3 || random5 == random4);

do{random6 = rand.nextInt(10);} while(random6 == random1 || random6 == random2 || random6 == random3 || random6 == random4 || random6 == random5);

Gamer1.listeyeKartEkle(pokemonCards[random1]);cardIndex.remove((Integer)random1);

Gamer1.listeyeKartEkle(pokemonCards[random2]);cardIndex.remove((Integer)random2);

Gamer1.listeyeKartEkle(pokemonCards[random3]);cardIndex.remove((Integer)random3);

Gamer2.listeyeKartEkle(pokemonCards[random4]);cardIndex.remove((Integer)random4);

Gamer2.listeyeKartEkle(pokemonCards[random5]);cardIndex.remove((Integer)random5);

Gamer2.listeyeKartEkle(pokemonCards[random6]);cardIndex.remove((Integer)random6);

for(int i : cardIndex)

{

CommonCards.add(pokemonCards[i]);

}

for(Game.Pokemon poke : Gamer1.getKartList())

{

System.out.println("Gamer 1 pokemon : " + poke.getPokemonAdi());

}

for(Game.Pokemon poke : Gamer2.getKartList())

{

System.out.println("Gamer 2 pokemon : " + poke.getPokemonAdi());

}

for(Game.Pokemon poke : CommonCards)

{

System.out.println("Ortadaki pokemon : " + poke.getPokemonAdi());

}

jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+Gamer1.getKartList().get(0).getPokemonAdi()+".jpg")));

jButton15.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+Gamer1.getKartList().get(1).getPokemonAdi()+".jpg")));

jButton17.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+Gamer1.getKartList().get(2).getPokemonAdi()+".jpg")));

jButton19.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

jButton20.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

jButton22.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

jButton10.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

jButton8.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

jButton11.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

jButton12.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Image/closedcard.png")));

Oyun tipi kontrolü yap. (Oyuncu vs Bilgisayar mı?)

if(Selection.getSelectedGameType()==2)

{

// Bilgisayar vs Bilgisayar ise

Gamer1 = new Game.Bilgisayar();

Gamer2 = new Game.Bilgisayar();

AssignCards();

jLabel1.setText("Oyuncu 1 : Bilgisayar");

jLabel2.setText("Oyuncu 2 : Bilgisayar");

ShowHideScreen(true);

ResetScreen();

Computer1Play();

}

else

{

Gamer1 = new Game.Kullanici();

Gamer2 = new Game.Bilgisayar();

AssignCards();

jLabel1.setText("Oyuncu 1 : Kullanıcı");

jLabel2.setText("Oyuncu 2 : Bilgisayar");

ShowHideScreen(true);

ResetScreen();

}

Oyuncu vs Bilgisayar ise oyuncu butona basınca kartı oyna.

if(Roundc==0 && Roundp==0)

{

jButton1.setEnabled(false);

SetLeftArenaText(Gamer1.getKartList().get(0), 0);

Gamer1.getKartList().get(0).setKartKullanildimi(true);

Computer2Play(0);

Roundp++;

}

else if(Roundc==1 && Roundp==1)

{

jButton1.setEnabled(false);

SetLeftArenaText(Gamer1.getKartList().get(0), 0);

Gamer1.getKartList().get(0).setKartKullanildimi(true);

Computer2Play(0);

Roundp++;

}

else if(Roundc==2 && Roundp==2)

{

jButton1.setEnabled(false);

SetLeftArenaText(Gamer1.getKartList().get(0), 0);

Gamer1.getKartList().get(0).setKartKullanildimi(true);

Computer2Play(0);

Roundc++;

}

else if(Roundc<5 && Roundc>2 && Roundp==2)

{

jButton1.setEnabled(false);

SetLeftArenaText(Gamer1.getKartList().get(0), 0);

Gamer1.getKartList().get(0).setKartKullanildimi(true);

Computer2Play(0);

// jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+Gamer1.getKartList().get(0).getPokemonAdi()+".jpg")));

Roundc++;

}

else

{

JOptionPane.showMessageDialog(null, "Öncelikle Ortadan Kart Seçmelisiniz", "Uyarı !", 1);

}

Bilgisayar vs Bilgisayar ise random seçim ile oyunu başlat.

or(Game.Pokemon poke: Gamer1.getKartList())

{

if(poke.getKartKullanildimi() == false)

{

isGamer1Empty = false;

break;

}

}

Game.Pokemon pokemon = null;

Game.Bilgisayar cpu = null;

if(!isGamer1Empty)

{

pokemon = Gamer1.kartSec(null);

cpu = (Game.Bilgisayar)Gamer1;

}

List<Game.Pokemon> activeCommonCards = new ArrayList();

for(Game.Pokemon poke: CommonCards)

{

if(poke.getKartKullanildimi() == false)

{

activeCommonCards.add(poke);

}

}

if(!isGamer1Empty)

{activeCommonCards.add(pokemon);}

System.out.println(activeCommonCards.size());

Random rand = new Random();

int pokeIndex = rand.nextInt(activeCommonCards.size());

Game.Pokemon pokemonSelected1 = activeCommonCards.get(pokeIndex);

if(pokemonSelected1 != pokemon)

{

if(jButton16.getToolTipText() == "99" || jButton9.getToolTipText() == "99")

{

if(jButton16.getToolTipText() == "99")

{

// jButton16.setText("<html>" + pokemonSelected1.getPokemonAdi() + "<br /><br />" + "Güç : " + pokemonSelected1.hasarPuaniGoster() + "</html>");

jButton16.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+pokemonSelected1.getPokemonAdi()+".jpg")));

jButton16.setToolTipText(""+pokeIndex);

jButton16.setEnabled(false);

}

else

{

//jButton9.setText("<html>" + pokemonSelected1.getPokemonAdi() + "<br /><br />" + "Güç : " + pokemonSelected1.hasarPuaniGoster() + "</html>");

jButton9.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+pokemonSelected1.getPokemonAdi()+".jpg")));

jButton9.setToolTipText(""+pokeIndex);

jButton9.setEnabled(false);

}

// jButton18.setText("<html>" + pokemonSelected1.getPokemonAdi() + "<br /><br />" + "Güç : " + pokemonSelected1.hasarPuaniGoster() + "</html>");

jButton18.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+pokemonSelected1.getPokemonAdi()+".jpg")));

jButton18.setToolTipText(""+pokeIndex);

int commonIndex = 0;

for(Game.Pokemon poke: CommonCards)

{

if(poke.getPokemonID() == pokemonSelected1.getPokemonID())

{

break;

}

commonIndex++;

}

CommonCards.get(commonIndex).setKartKullanildimi(true);

Computer2PlayAfterCommon(commonIndex);

DisableCommonCard(CommonCards.get(commonIndex).getPokemonAdi());

}

else

{

pokemonSelected1 = pokemon;

Gamer1.getKartList().get(cpu.\_lastCardIndex).setKartKullanildimi(true);

jButton18.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+Gamer1.getKartList().get(cpu.\_lastCardIndex).getPokemonAdi()+".jpg")));

jButton18.setToolTipText(""+cpu.\_lastCardIndex);

Computer2Play(cpu.\_lastCardIndex);

ShowIconGamer1Card(Gamer1.getKartList().get(cpu.\_lastCardIndex).getPokemonAdi());

}

}

else

{

SetRightArenaText(pokemon, cpu.\_lastCardIndex);

Gamer1.getKartList().get(cpu.\_lastCardIndex).setKartKullanildimi(true);

jButton18.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+Gamer1.getKartList().get(cpu.\_lastCardIndex).getPokemonAdi()+".jpg")));

jButton18.setToolTipText(""+cpu.\_lastCardIndex);

Computer2Play(cpu.\_lastCardIndex);

ShowIconGamer1Card(Gamer1.getKartList().get(cpu.\_lastCardIndex).getPokemonAdi());

}

Oyuncu ya da bilgisayar olarak ilk kullanıcı kartını oynadıktan sonra bilgisayar 2 rastgele kart seçerek kartını oynar ve log alanını gunceller.

Game.Pokemon pokemon = Gamer2.kartSec(null);

Game.Bilgisayar cpu = (Game.Bilgisayar)Gamer2;

List<Game.Pokemon> activeCommonCards = new ArrayList();

for(Game.Pokemon poke: CommonCards)

{

if(poke.getKartKullanildimi() == false)

{

activeCommonCards.add(poke);

}

}

activeCommonCards.add(pokemon);

Random rand = new Random();

int pokeIndex = rand.nextInt(activeCommonCards.size());

Game.Pokemon pokemonSelected1 = activeCommonCards.get(pokeIndex);

if(pokemonSelected1 != pokemon)

{

if(jButton21.getToolTipText() == "99" || jButton23.getToolTipText() == "99")

{

if(jButton21.getToolTipText() == "99")

{

// jButton21.setText("<html>" + pokemonSelected1.getPokemonAdi() + "<br /><br />" + "Güç : " + pokemonSelected1.hasarPuaniGoster() + "</html>");

jButton21.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+pokemonSelected1.getPokemonAdi()+".jpg")));

jButton21.setToolTipText(""+pokeIndex);

jButton21.setEnabled(false);

}

else

{

// jButton23.setText("<html>" + pokemonSelected1.getPokemonAdi() + "<br /><br />" + "Güç : " + pokemonSelected1.hasarPuaniGoster() + "</html>");

jButton23.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+pokemonSelected1.getPokemonAdi()+".jpg")));

jButton23.setToolTipText(""+pokeIndex);

jButton23.setEnabled(false);

}

int commonIndex = 0;

for(Game.Pokemon poke: CommonCards)

{

if(poke.getPokemonID() == pokemonSelected1.getPokemonID())

{

break;

}

commonIndex++;

}

CommonCards.get(commonIndex).setKartKullanildimi(true);

DisableCommonCard(CommonCards.get(commonIndex).getPokemonAdi());

}

else

{

pokemon.setKartKullanildimi(true);

pokemonSelected1 = pokemon;

pokemonSelected1.setKartKullanildimi(true);

SetRightArenaText(pokemon, cpu.\_lastCardIndex);

ShowIconGamer2Card(pokemonSelected1.getPokemonAdi());

}

}

else

{

pokemon.setKartKullanildimi(true);

pokemonSelected1 = pokemon;

pokemonSelected1.setKartKullanildimi(true);

SetRightArenaText(pokemon, cpu.\_lastCardIndex);

ShowIconGamer2Card(pokemonSelected1.getPokemonAdi());

}

jTextArea1.append("Oyuncu 1 : " + Gamer1.getKartList().get(gamer1Index).getPokemonAdi() + " : " + Gamer1.getKartList().get(gamer1Index).hasarPuaniGoster() + " vs ");

jTextArea1.append("Oyuncu 2 : " + pokemonSelected1.getPokemonAdi() + " : " + pokemonSelected1.hasarPuaniGoster());

//jButton2.setText("<html>" + pokemonSelected1.getPokemonAdi() + "<br /><br />" + "Güç : " + pokemonSelected1.hasarPuaniGoster() + "</html>");

jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+pokemonSelected1.getPokemonAdi()+".jpg")));

jButton2.setToolTipText(""+pokeIndex);

if(Gamer1.getKartList().get(gamer1Index).hasarPuaniGoster() > pokemonSelected1.hasarPuaniGoster())

{

jTextArea1.append(" Kazanan OYUNCU 1 \n");

this.Oyuncu1Skor++;

}

else if(Gamer1.getKartList().get(gamer1Index).hasarPuaniGoster() < pokemonSelected1.hasarPuaniGoster())

{

jTextArea1.append(" Kazanan OYUNCU 2 \n");

this.Oyuncu2Skor++;

}

else

{

jTextArea1.append(" BERABERE \n");

this.Oyuncu1Skor++;

this.Oyuncu2Skor++;

}

jLabel9.setText("Skor : " + this.Oyuncu1Skor);

jLabel10.setText("Skor : " + this.Oyuncu2Skor);

if(Gamer1.getOyuncuId() == 2)

Computer1Play();

else

{

if(CheckGameIsFinish())

{

PrintResult();

}

}

Bilgisayar tarafından kart seçimi rastgele yapılmalıdır.

@Override

Pokemon kartSec(Pokemon selectedCard){

Random rand = new Random();

int index = 0;

List<Pokemon> pokelist = new ArrayList();

for(Pokemon poke: super.getKartList())

{

if(poke.getKartKullanildimi() == false)

pokelist.add(poke);

}

index = rand.nextInt(pokelist.size());

Pokemon pokemon = pokelist.get(index);

int tempCount = 0;

for(Pokemon subPoke: super.getKartList())

{

if(subPoke.getPokemonID() == pokemon.getPokemonID())

{

break;

}

tempCount++;

}

this.\_lastCardIndex = tempCount;

return pokemon;

}

Her hamle sonrası arena alanı güncellenir.

pokearenabuton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+ pokemon.getPokemonAdi()+".jpg")));

pokearenabuton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Cards/"+ pokemon.getPokemonAdi()+".jpg")));

switch (index) {

case 0:

oyuncu1buton.setEnabled(false);

break;

case 1:

oyuncu2buton.setEnabled(false);

break;

case 2:

oyuncu2buton.setEnabled(false);

break;

default:

break;

}

Her hamle öncesinde oyunun son durumu kontrol edilir.

for(Game.Pokemon poke: CommonCards)

{

if(poke.getKartKullanildimi() == false)

{

return false;

}

}

for(Game.Pokemon poke: Gamer1.getKartList())

{

if(poke.getKartKullanildimi() == false)

{

return false;

}

}

return true;

Oyun bitti ise ekrana gerekli mesajlar çıkarılır.

if(Oyuncu1Skor > Oyuncu2Skor)

{

JOptionPane.showMessageDialog(null, "Oyun sona erdi. Kazanan Oyuncu 1 !", "Oyuncu 1 Kazandı !", 1);

jTextArea1.append("Oyun sona erdi. Kazanan Oyuncu 1 !");

}

else if(Oyuncu1Skor < Oyuncu2Skor)

{

JOptionPane.showMessageDialog(null, "Oyun sona erdi. Kazanan Oyuncu 2 !", "Oyuncu 2 Kazandı !", 1);

jTextArea1.append("Oyun sona erdi. Kazanan Oyuncu 2 !");

}

else

{

JOptionPane.showMessageDialog(null, "Oyun sona erdi. Oyun Berabere Bitti !", "Dostluk Kazandı !", 1);

jTextArea1.append("Oyun sona erdi. Oyun Berabere Bitti !");

}

isShowCards.setEnabled(false);