|  |
| --- |
| package com.anderson.christoph.scarnesdice; |
|  |  |
|  | import android.os.Bundle; |
|  | import android.os.Handler; |
|  | import android.support.design.widget.FloatingActionButton; |
|  | import android.support.design.widget.Snackbar; |
|  | import android.support.v7.app.AppCompatActivity; |
|  | import android.support.v7.widget.Toolbar; |
|  | import android.util.Log; |
|  | import android.view.View; |
|  | import android.view.Menu; |
|  | import android.view.MenuItem; |
|  | import android.widget.ImageView; |
|  | import android.widget.TextView; |
|  | import android.widget.Button; |
|  |  |
|  | import com.google.common.collect.ImmutableMap; |
|  |  |
|  | import java.util.Random; |
|  |  |
|  | public class MyActivity extends AppCompatActivity { |
|  |  |
|  | private int MAX\_COMP\_TURN\_SCORE = 20; |
|  | private int userScore = 0; |
|  | private int roundScore = 0; |
|  | private int computerScore = 0; |
|  | private boolean userTurn; |
|  | private boolean gameOver; |
|  | private final int WINNING\_SCORE = 100; |
|  | private Random rand; |
|  |  |
|  | private ImmutableMap<Integer, Integer> diceResources = ImmutableMap.<Integer, Integer>builder() |
|  | .put(1, R.drawable.dice1) |
|  | .put(2, R.drawable.dice2) |
|  | .put(3, R.drawable.dice3) |
|  | .put(4, R.drawable.dice4) |
|  | .put(5, R.drawable.dice5) |
|  | .put(6, R.drawable.dice6) |
|  | .build(); |
|  |  |
|  | @Override |
|  | protected void onCreate(Bundle savedInstanceState) { |
|  | super.onCreate(savedInstanceState); |
|  | setContentView(R.layout.activity\_my); |
|  | Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar); |
|  | setSupportActionBar(toolbar); |
|  |  |
|  | rand = new Random(); |
|  | userTurn = true; |
|  | gameOver = false; |
|  |  |
|  | Button rollButton = (Button) findViewById(R.id.rollButton); |
|  | rollButton.setOnClickListener(new View.OnClickListener() { |
|  | @Override |
|  | public void onClick(View view) { |
|  | if (userTurn && !gameOver) { |
|  | roll(); |
|  | } |
|  | } |
|  | }); |
|  |  |
|  | Button holdButton = (Button) findViewById(R.id.holdButton); |
|  | holdButton.setOnClickListener(new View.OnClickListener() { |
|  | @Override |
|  | public void onClick(View view) { |
|  | // if (userTurn && !gameOver) { |
|  | hold(); |
|  | // updateScore(); |
|  | // if (!gameOver) computerTurn(); |
|  | // } |
|  | } |
|  | }); |
|  |  |
|  | Button resetButton = (Button) findViewById(R.id.resetButton); |
|  | resetButton.setOnClickListener(new View.OnClickListener() { |
|  | @Override |
|  | public void onClick(View view) { |
|  | reset(); |
|  | } |
|  | }); |
|  |  |
|  | } |
|  |  |
|  | public void roll() { |
|  | int roll = rand.nextInt(6) + 1; |
|  |  |
|  | ImageView dieImg = (ImageView)findViewById(R.id.dieView); |
|  | dieImg.setImageResource(diceResources.get(roll)); |
|  |  |
|  | if (roll == 1) { |
|  | roundScore = 0; |
|  | updateRoundScore(0); |
|  | switchTurn(); |
|  | } else { |
|  | roundScore += roll; |
|  | updateRoundScore(roundScore); |
|  | } |
|  | } |
|  |  |
|  | public void hold() { |
|  |  |
|  | if (userTurn) userScore += roundScore; |
|  | else computerScore += roundScore; |
|  |  |
|  | updateScore(); |
|  |  |
|  | roundScore = 0; |
|  | updateRoundScore(0); |
|  |  |
|  | if ((userScore > WINNING\_SCORE) || (computerScore > WINNING\_SCORE)) gameOver(); |
|  |  |
|  | if (!gameOver) switchTurn(); |
|  | } |
|  |  |
|  | public void switchTurn() { |
|  |  |
|  | TextView turnView = (TextView) findViewById(R.id.turnView); |
|  |  |
|  | if (userTurn) turnView.setText("Computer's Turn"); |
|  | else turnView.setText("Your Turn"); |
|  |  |
|  | userTurn = !userTurn; |
|  |  |
|  | // ImageView dieImg = (ImageView)findViewById(R.id.dieView); |
|  | // dieImg.setImageResource(diceResources.get(1)); |
|  |  |
|  | if (userTurn) { |
|  | Button rollButton = (Button) findViewById(R.id.rollButton); |
|  | rollButton.setEnabled(true); |
|  | Button holdButton = (Button) findViewById(R.id.holdButton); |
|  | holdButton.setEnabled(true); |
|  | } |
|  |  |
|  | if (!userTurn) computerTurn(); |
|  | } |
|  |  |
|  | public void computerTurn() { |
|  |  |
|  | final Handler handler = new Handler(); |
|  |  |
|  | // disable roll & hold buttons |
|  | Button rollButton = (Button) findViewById(R.id.rollButton); |
|  | rollButton.setEnabled(false); |
|  | Button holdButton = (Button) findViewById(R.id.holdButton); |
|  | holdButton.setEnabled(false); |
|  |  |
|  | handler.postDelayed(new Runnable() { |
|  | @Override |
|  | public void run() { |
|  | if ((roundScore + computerScore >= WINNING\_SCORE) || (roundScore >= MAX\_COMP\_TURN\_SCORE)) hold(); |
|  | else { |
|  | roll(); |
|  | } |
|  |  |
|  | if (!userTurn) handler.postDelayed(this, 1000); |
|  | else updateRoundScore(0); |
|  | } |
|  | }, 1000); |
|  |  |
|  | updateScore(); |
|  |  |
|  | roundScore = 0; |
|  | updateRoundScore(0); |
|  |  |
|  | // rollButton.setEnabled(true); |
|  | // holdButton.setEnabled(true); |
|  | } |
|  |  |
|  | public void gameOver() { |
|  | // disable roll & hold buttons |
|  | Button rollButton = (Button) findViewById(R.id.rollButton); |
|  | rollButton.setEnabled(false); |
|  | Button holdButton = (Button) findViewById(R.id.holdButton); |
|  | holdButton.setEnabled(false); |
|  |  |
|  | gameOver = true; |
|  | } |
|  |  |
|  | // ---------------------------DONE------------------------------- |
|  |  |
|  | public void reset() { |
|  | userScore = 0; |
|  | computerScore = 0; |
|  |  |
|  | roundScore = 0; |
|  | updateRoundScore(0); |
|  |  |
|  | userTurn = true; |
|  | gameOver = false; |
|  |  |
|  | TextView scoreView = (TextView)findViewById(R.id.scoreView); |
|  | scoreView.setText("Your Score: 0 Computer Score: 0"); |
|  |  |
|  | TextView turnView = (TextView)findViewById(R.id.turnView); |
|  | turnView.setText("Your turn"); |
|  |  |
|  | ImageView dieImg = (ImageView)findViewById(R.id.dieView); |
|  | dieImg.setImageResource(diceResources.get(1)); |
|  |  |
|  | Button rollButton = (Button) findViewById(R.id.rollButton); |
|  | rollButton.setEnabled(true); |
|  | Button holdButton = (Button) findViewById(R.id.holdButton); |
|  | holdButton.setEnabled(true); |
|  |  |
|  | } |
|  |  |
|  | public void updateScore() { |
|  | TextView scoreView = (TextView) findViewById(R.id.scoreView); |
|  | if (gameOver) { |
|  | if (userScore >= WINNING\_SCORE) { |
|  | scoreView.setText("Your Score: " + userScore + " You won!"); |
|  | } else { |
|  | scoreView.setText("Computer's Score: " + computerScore + " Computer won!"); |
|  | } |
|  | } else { |
|  | scoreView.setText("Your Score: " + userScore + " Computer Score: " + computerScore); |
|  | } |
|  | } |
|  |  |
|  | private void updateRoundScore(int score) { |
|  | TextView turnScoreView = (TextView)findViewById(R.id.turnScore); |
|  | turnScoreView.setText("Turn score: " + score); |
|  | } |
|  |  |
|  |  |
|  | @Override |
|  | public boolean onCreateOptionsMenu(Menu menu) { |
|  | // Inflate the menu; this adds items to the action bar if it is present. |
|  | getMenuInflater().inflate(R.menu.menu\_my, menu); |
|  | return true; |
|  | } |
|  |  |
|  | @Override |
|  | public boolean onOptionsItemSelected(MenuItem item) { |
|  | // Handle action bar item clicks here. The action bar will |
|  | // automatically handle clicks on the Home/Up button, so long |
|  | // as you specify a parent activity in AndroidManifest.xml. |
|  | int id = item.getItemId(); |
|  |  |
|  | //noinspection SimplifiableIfStatement |
|  | if (id == R.id.action\_settings) { |
|  | return true; |
|  | } |
|  |  |
|  | return super.onOptionsItemSelected(item); |
|  | } |
|  |  |
|  | } |