

TASK GUIDE (B3.07)

A. Objectives.

Student will start programming for PlayActivity. This task will introduce:

- how to gather Intent passing variables,
- how to use Timer with Handler,
- how to use Popup Menu, and
- how to create and access Fragment.

B. Requirements.

Hardware:

- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality

Software:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- JDK 8
- Android Studio IDE (Minimum 3.2) with AndroidX library.

C. Resources.

Documents:

- Guide

Test code:

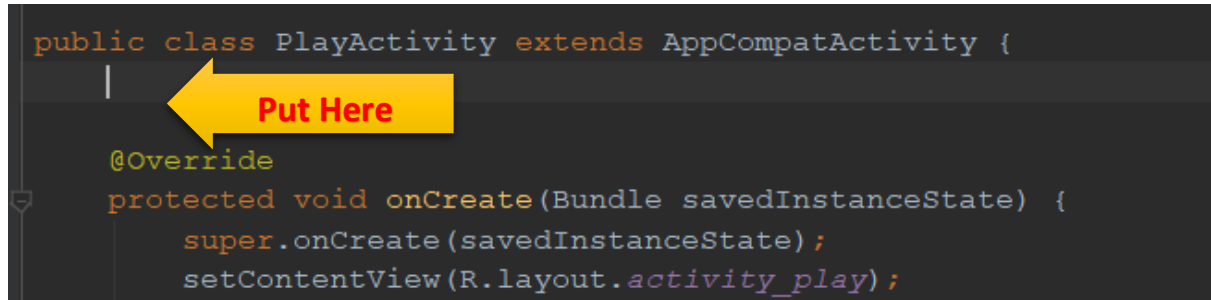
- TestB3MultiActivities071.java

D. Task Description.

Student start programming with PlayActivity.

E. Specification.

1. Open "PlayActivity.java" in java folder folder.
2. Declare all variables that represents all widgets in activity_play.xml.



```
public class PlayActivity extends AppCompatActivity {  
    |  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_play);  
    }  
}
```

A yellow arrow with the text "Put Here" points to the line between the class declaration and the onCreate method.

Using this template:

```
private <Widget_Type> <variable_name>;
```

All the widgets are:

Widget Type	Variable_name
ImageButton	'homeScoreBtn', 'homeYellowBtn', 'homeRedBtn', 'awayScoreBtn', 'awayYellowBtn', 'awayRedBtn'
Button	'startMatch', 'finishMatch'
ImageView	'homeLogo', 'awayLogo'
TextView	'timerTxt;', 'homeTeamTxt', 'awayTeamTxt', 'homeScoreTxt', 'awayScoreTxt'

3. Also define some variables like below:

```
private ArrayList<String> eventList = new ArrayList<>();  
private ArrayList<String> homePlayer, awayPlayer;  
final String separator = "@";  
  
long MillisecondTime, StartTime, TimeBuff, UpdateTime = 0L ;  
int Hours, Seconds, Minutes;  
Handler handler;
```

4. In the onCreate method, define all widgets variables, which has been declared in point 1, to the related widget id using this template:

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_play);

    <variable_name> = (<Widget_type>) findViewById(R.id.<widget_id>);
    .
    .
}

```

Also assign new Handler like below:

```

handler = new Handler;

```

5. To load the data from MainActivity as former Intent, create a method 'loadTeamData', like below:

```

private void loadTeamData() {
    Intent intent = getIntent();
    String homeTeam = intent.getStringExtra("HOME_TEAM_NAME");
    if (!homeTeam.isEmpty()) homeTeamTxt.setText(homeTeam);

    String awayTeam = intent.getStringExtra("AWAY_TEAM_NAME");
    if (!awayTeam.isEmpty()) awayTeamTxt.setText(awayTeam);

    homePlayer = intent.getStringArrayListExtra("HOME_TEAM_PLAYER");
    awayPlayer = intent.getStringArrayListExtra("AWAY_TEAM_PLAYER");

    try {
        String homeImgPath = intent.getStringExtra("HOME_IMG_URI");
        if (!homeImgPath.isEmpty()) {
            Uri homeUri = Uri.parse(homeImgPath);
            InputStream imageStream1 =
                getContentResolver().openInputStream(homeUri);
            Bitmap selectedImage1 = BitmapFactory.decodeStream(imageStream1);
            homeLogo.setImageBitmap(selectedImage1);
        }

        String awayImgPath = intent.getStringExtra("AWAY_IMG_URI");
        if (!awayImgPath.isEmpty()) {
            Uri awayUri = Uri.parse(awayImgPath);
            InputStream imageStream2 =
                getContentResolver().openInputStream(awayUri);
            Bitmap selectedImage2 = BitmapFactory.decodeStream(imageStream2);
            awayLogo.setImageBitmap(selectedImage2);
        }
    } catch (Exception e) {
        Toast.makeText(PlayActivity.this, "Can't load image",
            Toast.LENGTH_LONG).show();
    }
}

```

The code shows a Toast will be appear when one or both of team logo can't be loaded.

6. Then, to show the Footer as a fragment, create a method like below:

```
private void showFooter() {  
    FragmentManager fragmentManager = getSupportFragmentManager();  
    FragmentTransaction fragmentTransaction =  
        fragmentManager.beginTransaction();  
  
    FooterFragment footer = new FooterFragment();  
    fragmentTransaction.add(R.id.footer, footer, "footer");  
    fragmentTransaction.commit();  
}
```

7. Then, create an Override method 'onStart' which calls method in point 6 and 7, like below:

```
@Override  
protected void onStart() {  
    super.onStart();  
    loadTeamData();  
    showFooter();  
}
```

8. Create a new method to set the Enabled of match Buttons, like below.

```
private void setEventButtonClickable(boolean status) {  
    homeScoreBtn.setEnabled(status);  
    homeYellowBtn.setEnabled(status);  
    homeRedBtn.setEnabled(status);  
    awayScoreBtn.setEnabled(status);  
    awayYellowBtn.setEnabled(status);  
    awayRedBtn.setEnabled(status);  
}
```

9. Create a new method to response when start Button is clicked, like below.

```

private void startBtnClick() {
    if (startMatch.getText().equals("Start")) {
        StartTime = SystemClock.uptimeMillis();
        handler.postDelayed(runnable, 1000);

        //Setup the buttons
        startMatch.setText("Stop");
        setEventButtonClickable(true);
        finishMatch.setEnabled(false);
    } else if (startMatch.getText().equals("Stop")) {
        TimeBuff += MillisecondTime;
        handler.removeCallbacks(runnable);

        //Setup the buttons
        startMatch.setText("Start");
        setEventButtonClickable(false);
        finishMatch.setEnabled(true);
    }
}

```

10. In the onCreate method, create an OnClickListener for Button 'startMatch' with call method 'startBtnClick' in point 9 like below:

```

startMatch.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startBtnClick();
    }
});

```

11. Create a Runnable class to run the Timer in background, like below:

```

public Runnable runnable = new Runnable() {
    public void run() {
        MillisecondTime = SystemClock.uptimeMillis() - StartTime;
        UpdateTime = TimeBuff + MillisecondTime;
        Seconds = (int) (UpdateTime / 1000);
        Hours = Seconds/3600;
        Seconds = Seconds % 3600;
        Minutes = Seconds / 60;
        Seconds = Seconds % 60;

        timerTxt.setText("" + String.format("%02d", Hours) + ":"
            + String.format("%02d", Minutes) + ":"
            + String.format("%02d", Seconds));

        handler.postDelayed(this, 0);
    }
};

```

12.Create a new method to add Event which is shown on Footer fragment, like below.

```
private void addEvent(String eventName, String playerName,
    String teamName, String currTime) {
    String data = eventName+separator+playerName+separator+
        teamName+separator+currTime;
    eventList.add(data);

    Fragment footerFragment =
        getSupportFragmentManager().findFragmentByTag("footer");
    View footerView = footerFragment.getView();
    TextView news = footerView.findViewById(R.id.newsTxt);
    String newsText = eventName+" for "+teamName+" by "+
        playerName+" at "+currTime;
    news.setText(newsText,TextView.BufferType.EDITABLE);
}
```

13.Create a new method to add Score which is shown on main screen, like below.

```
private void addScore(TextView scoreTxt) {
    int score = Integer.parseInt(scoreTxt.getText().toString())+1;
    scoreTxt.setText(""+score,TextView.BufferType.EDITABLE);
}
```

14.Create a new method to show the PopupMenu and response it with utilizing the method 'addEvent' and 'addScore', like below.

```

private void showPopupMenu(ArrayList<String> list, View anchor) {
    //Creating the instance of PopupMenu
    Context wrapper = new ContextThemeWrapper
        (getApplicationContext(), R.style.PopupMenu);
    PopupMenu popup = new PopupMenu(wrapper, anchor);
    for (int i=0; i<list.size(); i++) {
        popup.getMenu().add(list.get(i));
    }
    String currTime = timerTxt.getText().toString();

    //registering popup with OnMenuItemClickListener
    popup.setOnMenuItemClickListener(new
        PopupMenu.OnMenuItemClickListener() {
        public boolean onMenuItemClick(MenuItem item) {
            switch (anchor.getId()) {
                case R.id.addHomeScore: {
                    addEvent("Goal", item.getTitle().toString(),
                        homeTeamTxt.getText().toString(), currTime);
                    addScore(homeScoreTxt);
                    break;
                }
                case R.id.addHomeYellow:
                    addEvent("Yellow Card", item.getTitle().toString(),
                        homeTeamTxt.getText().toString(), currTime);
                    break;
                case R.id.addHomeRed:
                    addEvent("Red Card", item.getTitle().toString(),
                        homeTeamTxt.getText().toString(), currTime);
                    break;
                case R.id.addAwayScore: {
                    addEvent("Goal", item.getTitle().toString(),
                        awayTeamTxt.getText().toString(), currTime);
                    addScore(awayScoreTxt);
                    break;
                }
                case R.id.addAwayYellow:
                    addEvent("Yellow Card", item.getTitle().toString(),
                        awayTeamTxt.getText().toString(), currTime);
                    break;
                case R.id.addAwayRed:
                    addEvent("Red Card", item.getTitle().toString(),
                        awayTeamTxt.getText().toString(), currTime);
                    break;
                default: break;
            }

            return true;
        }
    });
    popup.show();
}

```

15. In the onCreate method, create some OnClickListener for each Red Card, Yellow Card, and Score Buttons Home and Away which call 'showPopupMenu' like below:

```

homeScoreBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showPopupMenu(homePlayer,homeScoreBtn);
    }
});

homeYellowBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showPopupMenu(homePlayer,homeYellowBtn);
    }
});

homeRedBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showPopupMenu(homePlayer,homeRedBtn);
    }
});

awayScoreBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showPopupMenu(awayPlayer,awayScoreBtn);
    }
});

awayYellowBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showPopupMenu(awayPlayer,awayYellowBtn);
    }
});

awayRedBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showPopupMenu(awayPlayer,awayRedBtn);
    }
});

```

16.Create a new method to show LogActivity as Intent with passing some variables, like below.


```

private void openLogActivity() {
    String matchResult, scoreResult;
    int homeScore = Integer.parseInt(homeScoreTxt.getText().toString());
    int awayScore = Integer.parseInt(awayScoreTxt.getText().toString());
    if (homeScore==awayScore) {
        matchResult = "Draw";
        scoreResult = homeScore+"-"+awayScore;
    } else if (homeScore>awayScore) {
        matchResult = homeTeamTxt.getText().toString()+" Win!!";
        scoreResult = homeScore+"-"+awayScore;
    } else {
        matchResult = awayTeamTxt.getText().toString()+" Win!!";
        scoreResult = awayScore+"-"+homeScore;
    }
    Intent log = new Intent(getApplicationContext(),LogActivity.class);
    log.putExtra("MATCH_RESULT", matchResult);
    log.putExtra("MATCH_SCORE", scoreResult);
    log.putStringArrayListExtra("MATCH_EVENT", eventList);

    startActivity(log);
}

```

17. In the onCreate method, create an OnClickListener for Button 'finishMatch' with call method 'openLogActivity' in point 16 like below:

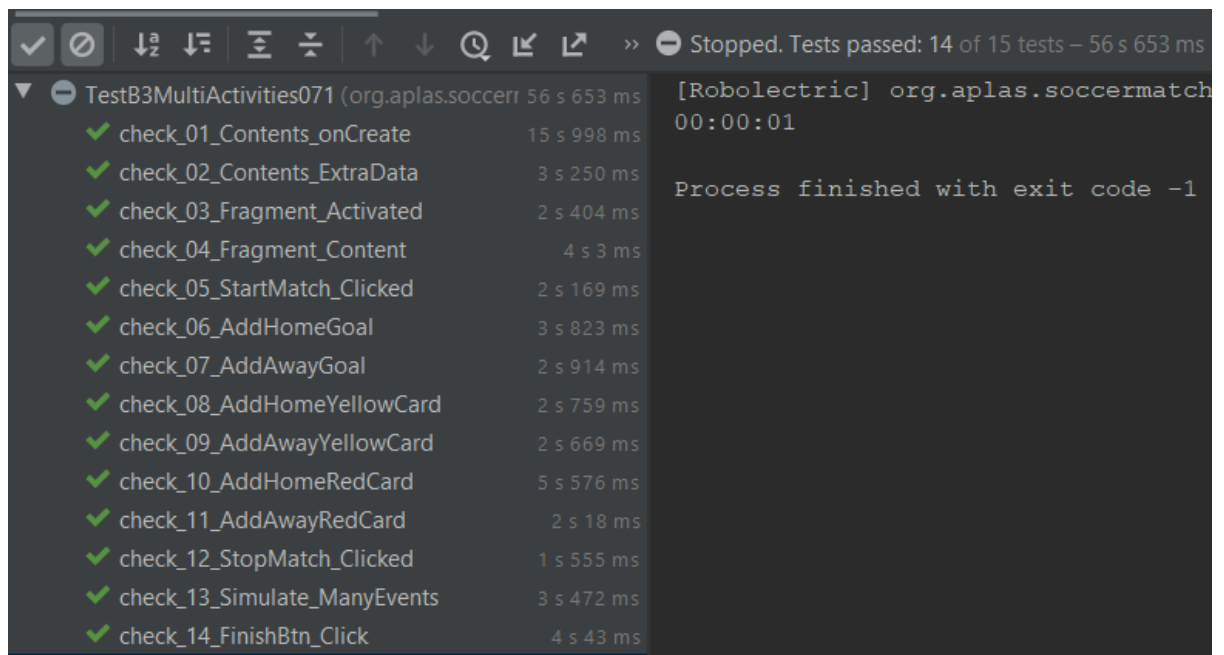
```

finishMatch.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openLogActivity();
    }
});

```

F. Testing.

1. Copy "TestB3MultiActivities071.java" file to "org.aplas.soccermatch (test)" folder.
2. Right click on the "TestB3MultiActivities071.java" file then choose Run. It may take long time to execute.
3. Get the result of your task. If passed you will get green check like below. If the test failed, you will get orange check get the messages and you must check your work again.



You have to try until get all green checks and continue to the next task.

G. Note.

You can modify the style and the layout arrangement of the Play activity by yourself to improve the UI design.