

פרק 23 ב - Canvas and View

: canvas אוביייקט שנותן לנו אפשרות לצייר עליו תמונות או text במצעות פונקציות .

- Paint אוביייקט שמכיל את התכונות כגון גודל, צבע, פונט ועוד

- Path אוביייקט שנותן לצייר צורות שונות

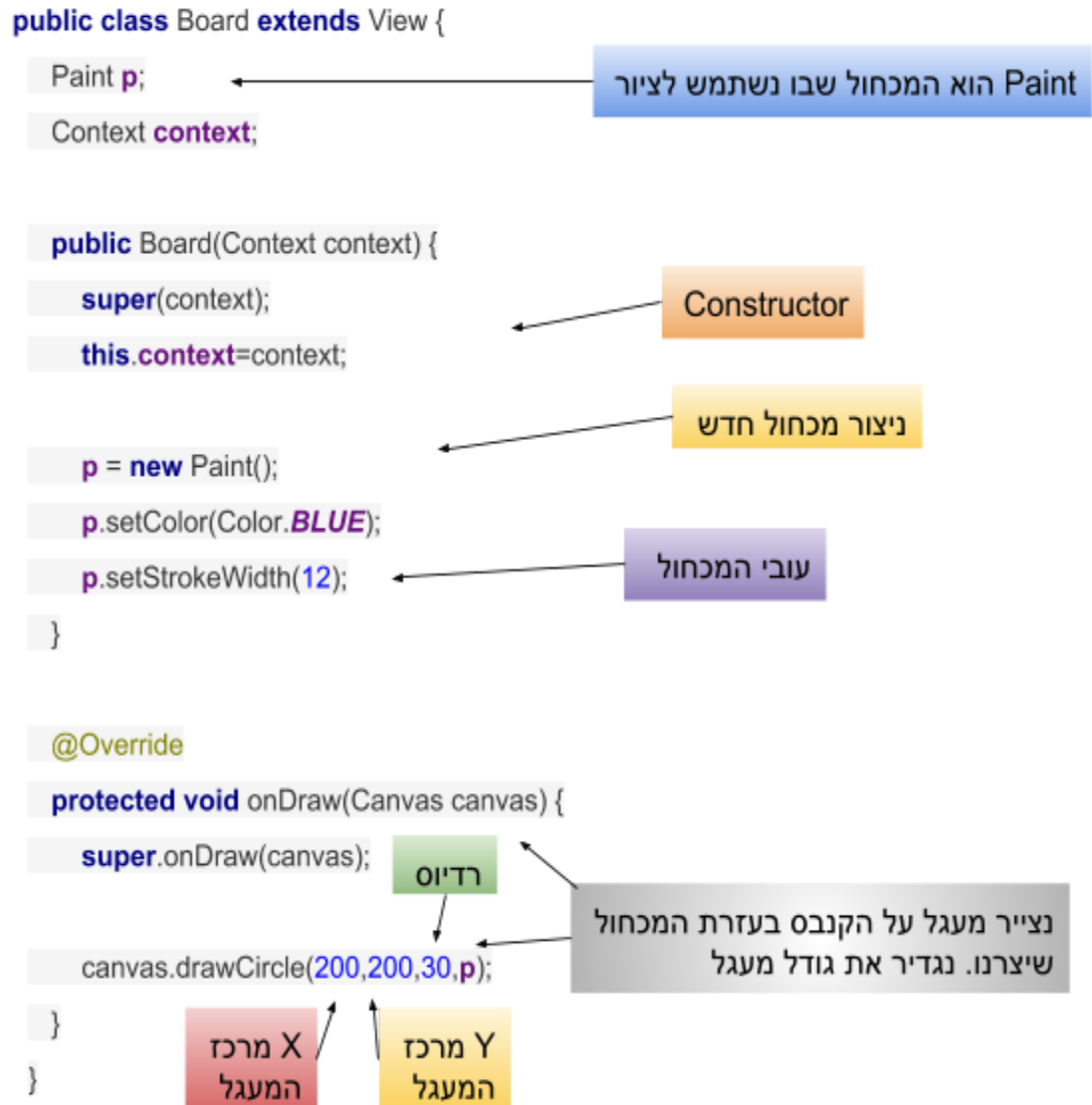
class view

onDraw : פונקציה שמקבלת כפרמטר canvas שעליו ניתן לצייר.

כאשר נוצר אוביייקט view אזי הפונקציה onDraw נוצרת אוטומטית.

דוגמה ראשונה לשימוש ב View:

שלב 1 - ניצור class חדש:



שלב 2 - ניצור מופע של הקנבס ב Main:

```
public class MainActivity extends AppCompatActivity {
```

```
    Board board;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    // setContentView(R.layout.activity_main);
```

```
    // linearLayout = (LinearLayout)findViewById(R.id.game);
```

```
    // linearLayout.addView(board);
```

```
    board = new Board(this);
```

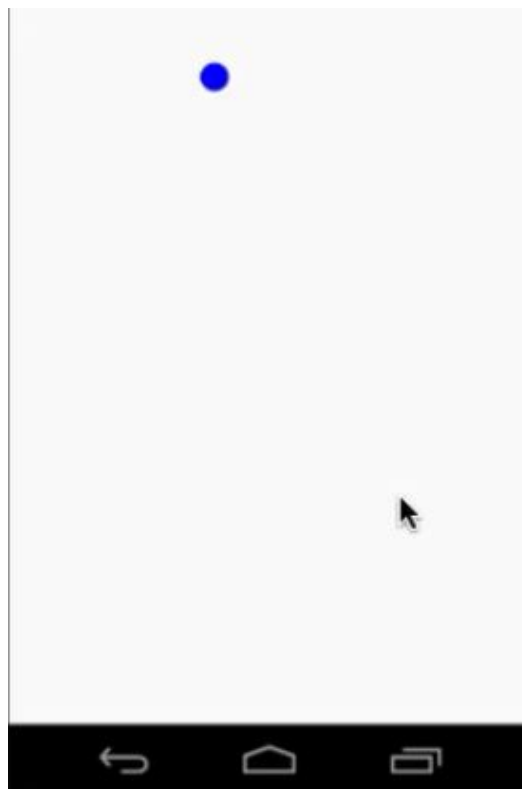
```
    setContentView(board);
```

```
}
```

```
}
```

נוכל לשים את הלוח
בתוך LinearLayout

נוכל להגדיר את הלוח להיות
ה layout של ה MainActivity



דוגמה 2 - ציור זז על גבי הקנבס:

נזיז את הכדור שבנינו בדוגמה 1:

```
public class Board extends View {
```

```
    float x,y,r;
```

```
    float deltax,deltay;
```

```
    Paint p;
```

```
    Context context;
```

ה x ו ה y של מרכז המעגל

השינוי ב x וב y
של מרכז המעגל

```
public Board(Context context) {
```

```
    super(context);
```

```
    this.context=context;
```

```
    p = new Paint();
```

```
    p.setColor(Color.BLUE);
```

```
    p.setStrokeWidth(12);
```

```
    x =10;
```

```
    y=100;
```

```
    deltax =10;
```

```
    deltay =5;
```

```
    r = 100;
```

```
}
```

@Override

```
protected void onDraw(Canvas canvas) {
```

```
    super.onDraw(canvas);
```

```
    canvas.drawCircle(x,y,r,p);
```

```
    x = x + deltax;
```

```
    y = y + deltay;
```

נשנה את ה x בכל פעם
שנקרא ל onDraw

```
    if(x < 0 || x > canvas.getWidth() - r)
```

```
        deltax = -deltax;
```

במידה וחרגנו מגבולות
הקנבס נשנה את כיוון
התזוזה של המעגל

```
    if(y < 0 || y > canvas.getHeight() - r )
```

```
        deltay = -deltay;
```

```
    invalidate();
```

invalidate() היא פונקציה שמוחקת
את המסך וקוראת ל onDraw שוב

```
    }
```

```
}
```

```
public class MainActivity extends AppCompatActivity {
```

```
    Board board;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        // setContentView(R.layout.activity_main);
```

```
        // linearLayout = (LinearLayout)findViewById(R.id.game);
```

```
        // linearLayout.addView(board);
```

```
        board = new Board(this);
```

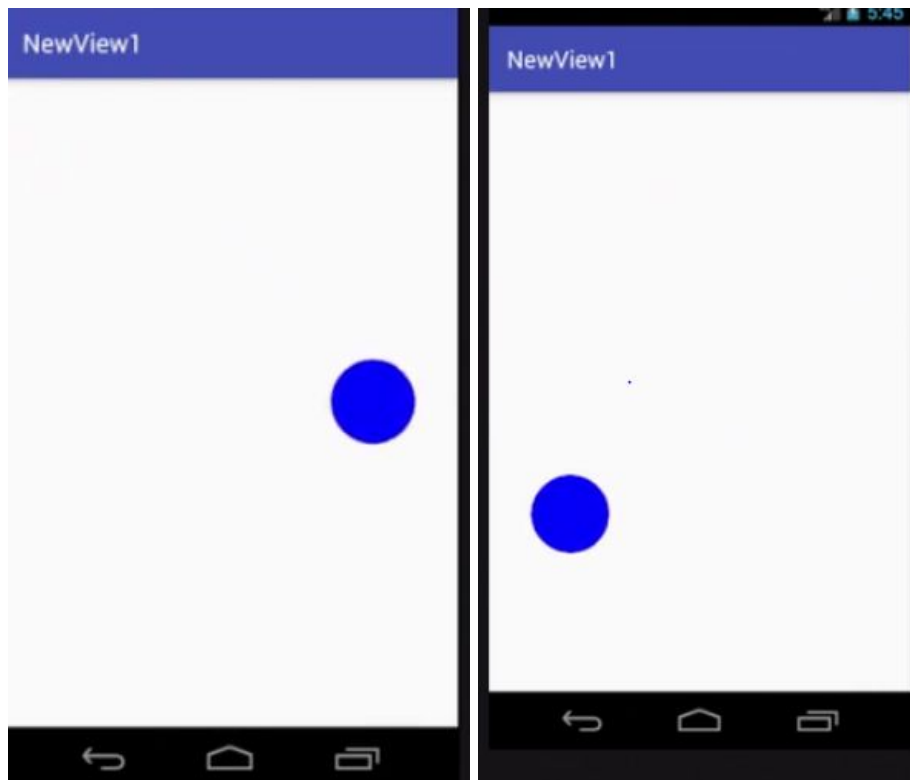
```
        setContentView(board);
```

```
    }
```

```
}
```

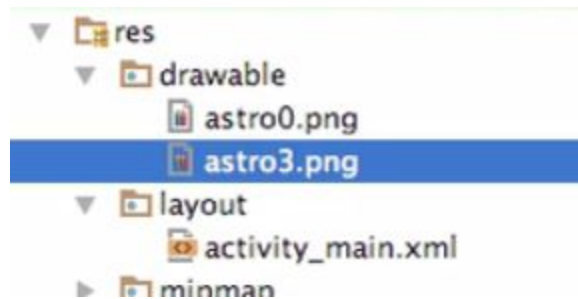
נוכל לשים את הלוח
בתוך LinearLayout

נוכל להגדיר את הלוח להיות
ה layout של ה MainActivity



דוגמה לציור תמונה על קנבס:

שלב 1 - נעלה תמונה לתיקיית drawable מהמחשב שלנו:



נוסיף את התמונה לקנבס שבנינו בדוגמה הקודמת:


```
public class Board extends View {
    Bitmap bitmap;
    float x,y,r;
    float deltax,deltay;
    Paint p;
    Context context;
```

← Bitmap לאחסון התמונה

```
public Board(Context context) {
    super(context);
    this.context=context;
```

```
    p = new Paint();
    p.setColor(Color.BLUE);
    p.setStrokeWidth(12);
    x = 10;
    y = 100;
    deltax = 10;
    deltay = 5;
    r = 100;
```

← נשמור את התמונה שהעלטנו ב Bitmap

```
    bitmap = BitmapFactory.decodeResource(getResources(),R.drawable.astro0);
}
```

@Override

```
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    //canvas.drawCircle(x,y,r,p);
    x = x + deltax;
    y = y + deltay;
```

← נשנה את ה x וה y בכל פעם כך שיראה כאילו התמונה זזה על גבי המסך

```
    canvas.drawBitmap(bitmap,x,y,null);
```

← נצייר את התמונה על הקנבס ב x ו y שהגדרנו

```
    if(x < 0 || x > canvas.getWidth() - bitmap.getWidth())
        deltax = -deltax;
```

```
    if(y < 0 || y > canvas.getHeight() - bitmap.getHeight() )
        deltay = -deltay;
```

```
    invalidate();
```

```
}
```

```
}
```

```
public class MainActivity extends AppCompatActivity {
```

```
    Board board;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    // setContentView(R.layout.activity_main);
```

```
    // linearLayout = (LinearLayout)findViewById(R.id.game);
```

```
    // linearLayout.addView(board);
```

```
    board = new Board(this);
```

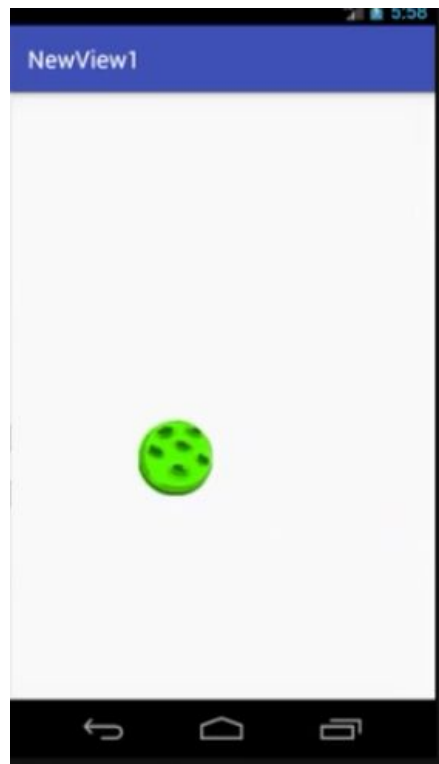
```
    setContentView(board);
```

```
}
```

```
}
```

נוכל לשים את הלוח
בתוך LinearLayout

נוכל להגדיר את הלוח להיות
ה layout של ה Main



דוגמה לציור טקסט על קנבס:

נוסיף את הטקסט בפונקציית onDraw (נוסיף לדוגמה הקודמת)

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);

    //canvas.drawCircle(x,y,r,p);

    x = x + deltax;
    y = y + deltax;

    canvas.drawBitmap(bitmap,x,y,null);

    if(x < 0 || x > canvas.getWidth() - bitmap.getWidth())
        deltax = -deltax;

    if(y < 0 || y > canvas.getHeight() - bitmap.getHeight() )
        deltax = -deltax;

    canvas.drawColor(Color.YELLOW);
    Paint paintTitle=new Paint();
    paintTitle.setARGB(50, 20, 50, 100);
    paintTitle.setTextAlign(Paint.Align.CENTER);
    paintTitle.setTextSize(70);
    canvas.drawText("Hello Uzi", canvas.getWidth()/2,75, paintTitle);

    invalidate();
}
```

ניצור מכחול חדש

ניתן ערכים של red, green, blue
לצבע המכחול (נוכל לתת גם צבע
במקום לקבוע את הערכים בעצמנו)

מרכז הטקסט יהיה x של
אמצע המסך y =75

```
public class MainActivity extends AppCompatActivity {

    Board board;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // setContentView(R.layout.activity_main);

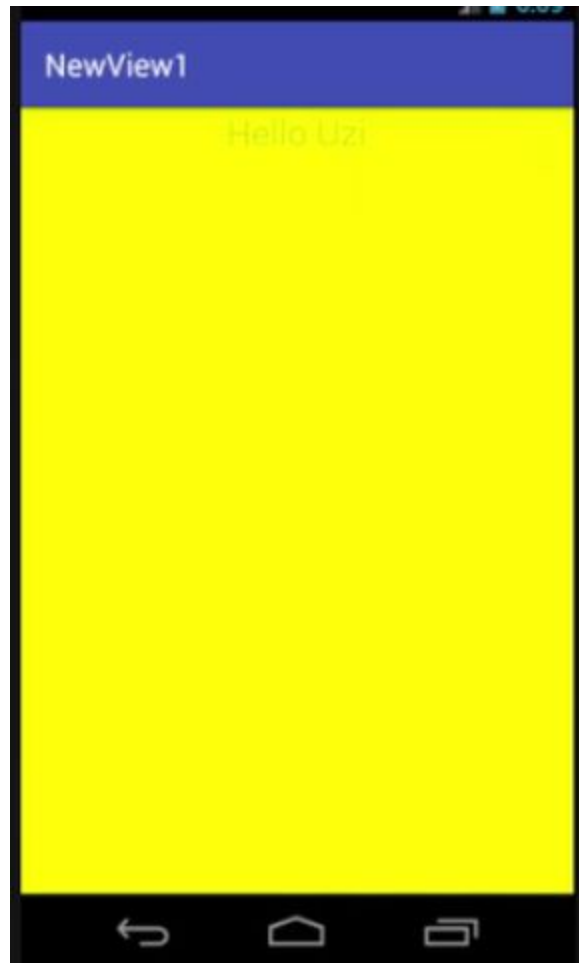
        // linearLayout = (LinearLayout)findViewById(R.id.game);

        // linearLayout.addView(board);

        board = new Board(this);
        setContentView(board);
    }
}
```

נוכל לשים את הלוח
בתוך LinearLayout

נוכל להגדיר את הלוח להיות
ה layout של ה MainActivity



דוגמה לציור מלבן על קנבס:

```

y = y + deltay;

canvas.drawBitmap(bitmap,x,y,null);

if(x < 0 || x > canvas.getWidth() - bitmap.getWidth())
    deltax = -deltax;

if(y < 0 || y > canvas.getHeight() - bitmap.getHeight() )
    deltay = -deltay;

canvas.drawColor(Color.YELLOW);
Paint paintTitle=new Paint();
paintTitle.setARGB(50, 20, 50, 100);
paintTitle.setTextAlign(Paint.Align.CENTER);
paintTitle.setTextSize(70);
canvas.drawText("Hello Uzi",canvas.getWidth()/2,75, paintTitle);

Rect r = new Rect(100,100,200,200);
canvas.drawRect(r,paintTitle);

invalidate();
}

```

X פינה שמאלית עליונה
 Y פינה שמאלית עליונה
 X פינה ימנית תחתונה
 Y פינה ימנית תחתונה

```
public class MainActivity extends AppCompatActivity {
```

```
    Board board;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        // setContentView(R.layout.activity_main);
```

```
        // linearLayout = (LinearLayout)findViewById(R.id.game);
```

```
        // linearLayout.addView(board);
```

```
        board = new Board(this);
```

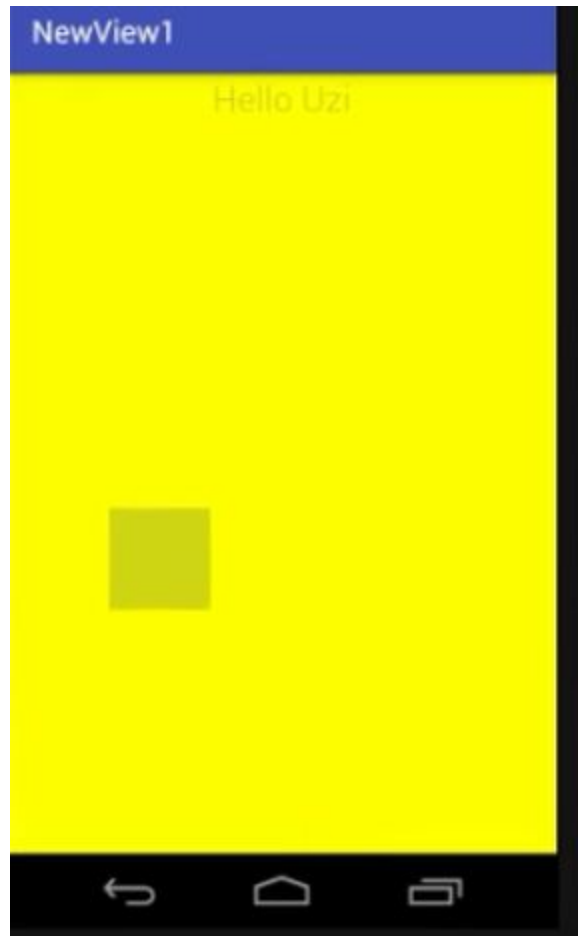
```
        setContentView(board);
```

```
    }
```

```
}
```

נוכל לשים את הלוח
בתוך LinearLayout

נוכל להגדיר את הלוח להיות
ה layout של ה Main



נוכל לעשות בעזרת הקנבס המון משחקים:
לתפוס את התמונה/ מלבן, מלבן שרודף אחרי עיגול וכו'.