1. Quickplay
    - Create random ship
2. Background
    - a. connecting controller to activity
    - b. Controller methods
        - 1. loadContent
            - a. Load Level and level's images at start of every level
        - 2. update – moving objects
            - a. propagation from controller – model.update
        - 3. draw – everything
            - a. propagation from controller – model.draw
        - 4. unloadContent (all level images) at end of every level
    - c. update/draw every 1/60 of second
3. Drawing
    - a. World/Device coordinate system
        - 1. World
            - a. Width and Height – developer defined
            - b. 0,0 is upper left
        - 2. Device
            - a. Width and Height
                - 1. How to get? Runtime, get device resolution

```
    Display display = getWindowManager().getDefaultDisplay();
    Point size = new Point();
    display.getRealSize(size);
    int width = size.x;
    int height = size.y;
```

        - 3. Translate from
            - a. World to device
            - b. Device to world
    - b. Useful classes
        - 1. PointF
        - 2. RectF
            - a. Bounding Boxes for moving objects, World coordinates, Viewport
            - b. static boolean intersects(RectF a, RectF b)
        - 1. DrawingHelper
            - a. drawImage
                - 1. imageId
                - 2. float x – device x (translate from World Coordinates)
                - 3. float y – device y (translate from World Coordinates)
                - 4. int rotation degrees – degrees counter-clockwise from 0
                    - a. Where is 0?
                    - b. How would you find out?
                - 5. double scale x
                - 6. double scale y

  - 7. alpha -- 255
- b. DrawingHelper.
  - 1. drawPointWith
  - 2. drawFilledCircle
  - 3. drawFilledRectangle -- minimap

4. Update Moving objects
   - a. Object Info
     - 1. PointF position
     - 2. RectF Bounding Box
       - a. include height and width
       - b. derived from object’s width/height and position
         - 1. position is upper left corner
           - a. x position minus ½ width of object
           - b. y position minus ½ height of object
       - c. The update values delta x and delta y for the bounding box are the same for the object
     - 3. Velocity/Speed -- double
     - 4. Direction in radians
       - 1. Use trig to compute delta x and delta y
     - 5. Rotation angle in degrees
     - 6. Scale
   - b. Use velocity to adjust position
     - 1. moveObject(PointF objPosition, RectF boundingBox, double speed, double angleInRadians -- direction, elaspedTime)
     - 2. ricichetObject(PointF objPosition, RectF boundingBox, double angleRadian – direction, float worldWidth, float worldHeight)
       - a. only for Asteroids,Spaceship, and Laser bolts
   - c. Adjusting Viewport
   - d. Growing asteroid – increase scale
5. Collisions
   - a. Use RectF.intersects
   - b. Spaceship
     - 1. reduce hit points
     - 2. If 0 – you loose
     - 3. make invincible for a while
   - c. Asteroid
     - 1. If first generation then split
       - a. compute number of splits
       - b. adjust scale by number of splits
       - c. make number of splits -1 copies
         - 1. set in random direction
     - 2. else remove from current level