Ra for Android

By Benjamin Sklar

Last Modified: 2017-06-30

# Overview

‘Ra for Android’ is my class project for [UW PCE](https://www.pce.uw.edu/) certificate [Android Application Development](https://www.pce.uw.edu/certificates/android-application-development). The project is to port the board game ‘Ra’ to the Android platform. ‘Ra’ is an auction and set-collection game with an Ancient Egyptian theme.

# Links

Ra description: <https://boardgamegeek.com/boardgame/12/ra>

Ra iOS version: <https://videogamegeek.com/videogame/86774/reiner-knizias-ra>  
 <https://itunes.apple.com/us/app/reiner-knizias-ra/id400213892?mt=8>

GIT repository: <https://github.com/brskl/ra>

# Summary of versions

## Version 0.1

The first version was my project for Autumn quarter 2016. The object is to get the game to function on the Android platform for a basic phone platform (tested using an emulator) with a textual interface showing the game state and buttons & dialogs to do game actions.

See ‘Project Presentation - Ra 2016-12.pptx’

## Version 0.2

The second version will/was my project for Winter quarter 2017. The object is to greatly improve the visual interface using graphics and other UI functionality that will be taught in the class for the winter quarter.

## Version 0.2.1

This minor update is to significantly improve the graphics of the application. This includes: modifying the layout to work on various tablet devices in landscape orientation, adding more drawables in various resolutions, and add animation.

## Version 0.3

The third version will/did primarily add networking/connectivity for multi-device play from Spring quarter 2017. Additionally posting to leaderboard (probably an AWS DynamoDB) and sharing of the final score will be added.

## Version 1.0

The first major version will be to make a commercial quality app, to be placed online for download. May charge for app, but will at least be a demonstration of my work to potential employers.

# Detail

## Version 0.1

### Overview

Ra for Android, v0.1 provides a textual interface with player actions occurring via buttons and dialogs. Full game functionality covering all the rules of the game. There is no networking, but multiple human can play on the same device. All game state is contained in the ‘Game’ class singleton object. The AI is very simplistic, more for testing purposes than actual game-play.

### Text abbreviations for the various tiles:

* DP: disaster – pharaoh
* DN: disaster – Nile
* DC: disaster – civilization
* DM: disaster – monuments
* M1-M8: Monuments, 8 types
* C1-C5: Civilization, 5 types
* P: Pharaoh
* N: Nile, NF: Nile Flood
* Au: Gold
* G: God
* R: RA

### Activities

* MainActivity – Initial activity providing player option to start a new game, resume an old game, or quit
* NewGameActivity – provides options to setup a new game including, number of players, name, and if player is AI or not. Currently there is no selection for AI player, and there is no mechanism for remote players.
* GameActivity – main game screen with action buttons. Shows Epoch #, # of Ra tiles, player’s Sun tokens, and current tiles available for auction. It does not show tiles each player has.
* TilesActivity – shows number of tiles each player currently has.
* ScoreActivity – shows score for each player if epoch ended at this point.

### Classes

Additonal classes other than activities

* Game – Contains all game state including player information using the ‘Player’ class
* Player – contains player information with 2 sub-classes
  + PlayerHuman
  + PlayerAI
* MyRandom – wrapper for pseudo-random number generator.

### Ideas

* Add ‘Ok’ button to ScoreActivity and TilesActivity to return to GameActivity
* Make TilesActivity and ScoreActivity have ‘ActionBar, Up button’ as a different way to return to GameActivity
* Add NavigationDrawer (see 1/24/2017 class notes) to GameActivity to move to ScoreActivity and TilesActivity instead of buttons.
* After API 23, android asks for permission at run-time not install time.  
  Can this be used to make app work only locally if permission is denied. Permission can be revoked in Android settings.

## Version 0.2

### Overview

Ra for Android, v0.2 implements various features taught during the winter quarter 2016. These include:

* Multiple layouts & Fragments: Provide different layouts for different devices and for orientation Portrait vs. Landscape
* Graphics: replace game elements that are currently text with graphics. Various types of drawables will be used.
* Strings: Make use of ‘Plural’ resources and provide Spanish (ES) and French (FR)
* ‘Up’ navigation from ‘Score’ and ‘Tiles’ views *up* to main game activity.   
  Add ‘swiping’ between detail views of individual players scores and tiles.
* Saving of game state.
* Some basic animation in movement of game elements
* Consider making use of ‘Application’ class
* Add Preferences/settings activity

### Implemented

* Strings: made use of ‘Plural’ strings
* Strings: translated into Spanish (ES) and French (FR)
* New launcher icon
* ‘Up’ navigation from ‘Score’ and ‘Tiles’ activities to ‘Game’ activity
* Saving of game state by saving ‘Game’ singleton from GameActivity
* Splash image that animates (alpha 0.0->1.0) on launch in MainActivity
* Updated ‘Score’ and ‘Tiles’ activity layouts to use images in header row.  
  (TODO: want to transpose row/column for portrait mode but haven’t been able to find a way to do this)
* Different layouts for ‘Score’ and ‘Tiles’ for portrait/landscape (just use bigger images in landscape, want to change table orientation, see previous item)
* Game activity, auction items: replaced text with LinearLayout of ImageViews that show PNGs of tiles.
* Created custom SunImageView (extends ImageView) class to drawText a number on top of a Sun Tile and use in Auction display
* Use SunImageView to display player’s suns and sun in auction.

## Version 0.2.1

### Overview

Ra for Android, v0.2.1 implements improved graphics:

* Improved layouts for various devices
* Improved drawables for various DPI density values
* Animation during game play.

### Layout

An important feature of the app is to work on various devices. The layouts targeted will be important not just for the activity layouts but also drawables size and resolution. The devices to be targeted will be limited to SmartPhones and Tablets; Wearables are too small and TV not the primary use-case. The initial list of devices to be targeted will include a range of ‘Density’ and ‘Size’ values. The emulators to be tested on from Android Studio AVD will include (Try to find density xxhdpi and xxxhdpi):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Category | Resolution | Density | Physical size | size |
| Nexus 10 | Tablet | 2560x1600 | xhdpi | 10.05” | xlarge |
| Nexus 7 | Tablet | 1200x1920 | Xhdpi | 7.02” | Large |
| Nexus 4 | Phone | 768x1280 | Xhdpi | 4.7” | Normal |
| Nexus One | Phone | 480x800 | Hdpi | 3.7” | Normal |
| 3.4” WQVGA | Phone | 240x432 | Ldpi | 3.4” | Small |
| 5.1” WVGA | Phone | 480x800 | Mdpi | 5.1” | large |

See: <https://developer.android.com/guide/practices/screens_support.html>

### Drawables

Need to determine what the size of drawables in ***dp*** units will be in the various layouts so as to specify layout\_width and layout\_height correctly. Initially for the ImageViews using specific dp values, but need to change to “wrap\_content” and have drawables in various sizes and resolutions based in screen size, density, and possibly smallest-screen-width. Advantage of using “wrap\_content” is that the layout can remain the same while there are different drawables which there needs to be anyway.  
Note: Sun-tiles and auction tiles have a background border so that the border is visible when empty; should the image be the full imageView so as to cover the border, or within the 1dp border? Also if “wrap\_content” is used and no image is set, the ImageView is shrunk to a 0x0 or 1x1 box, but resizes correctly when image resource is set. May need to have ‘border’ drawable that is a PNG rather than a shape XML file.

Additional images are needed for the various tiles not available in v0.2. This includes Ra-tile, and differentiation of god-tiles (maybe), civilization-tiles, and monument-tiles. Also the ‘Draw’ button should be replaced with an image of a bag.

### Animation

* On ‘Draw’ tiles fly from bag to appropriate location
* On auction won, winning sun-tile fly to auction, and auction tiles move to winning player
* On bid, sun-tile used becomes prominent.

### Implemented

* Using same layout file for different screen sizes, but use @dimen values in various places and have multiple dimen.xml files for various minimum smaller screen width values; currently have default (sw320dp), sw360dp, sw600dp, sw800dp.  
  In addition the layout has been greatly rearranged (landscape orientation only) to better represent the Ra game board.
* Replaced ‘Draw’ button with Image button.
* Added image for ‘Ra’ tile.
* Settings activity added (currently just has ‘Random Seed’ but more will be added)
  + Animation enabled (TODO: speed of animation)
* Animation, when drawn tile moves from ‘bag’ image to center to location on board
* Animation, when auction is won, tiles move from center to player center & fade

### TODO

* Animation issue: currently problem with animating multiple tiles, need to remove each imageView, but problem doing so from parent AnimationSet or from individual tiles smaller AnimationSet. Keep working on it. Currently when auction won, each tile has an imageView and an AnimationSet with a Translate&AlphaAnimation; these individual AnimationSets are combined (with later tiles having a delay) into a parent AnimationSet. Try with no delay between individual tile animations so all end at the same time.
* See if there is a better way to do ‘Sun tiles’ instead of a custom ImageView class, perhaps using LayerDrawable
* Make player Sun tiles look a bit different between available and used tiles.
* Get better artwork for PNG files and make in several resolutions  
  Improved drawables in various DPI resolutions
* Make ‘Score’ and ‘Tiles’ activities into fragments
* Replace ‘Auction’ button with ‘Ra’ tile image.
* Replace ‘God’ button with ‘God’ tile image.
* Write a game log file (different from Android Log() method for debugging)
* Sounds for various game events
* Add animations for various board-piece movement such as: draw tile from bag, acquire tiles won at auction.
* Add setting so that user can but does not have to press ‘OK’, ‘OK’, ‘OK’ when AI does something, have it move automatically with some amount of delay
* Add more detail to Tiles activity so that tiles for each individual player can be seen graphically
* Investigate ways to replace ‘Bid’ and ‘Resolve Disaster’ dialogs with clicking tiles on an activity
* Make first activity a splash screen only and have it launch MainActivity with ‘New’, ‘Resume’, ‘Quit’ button.
* Add ‘Help’ information
* Add ‘About me’ or ‘Credits’. Give credits to any art/icon/sounds I grab.

## Version 0.3

### Overview

Ra for Android, v0.3 implements various features taught during the spring quarter 2017:

* Connectivity and multi-device play, both ‘nearby’ via BlueTooth & WiFi, and far via Internet.  
  This is the primary new feature.
* Final score: sharing and posting to leaderboard (probably AWS DynamoDB with static website).
* Continued improvement of UI: better animation, drawables, improvement of layouts

### Implemented

* Score Activity, on game over, have Share menu item to send text with final score.
* On game over, upload score to AWS DynamoDB table.

#### Notes

* Consider using Snackbar instead of static textview in Ra application for game status. Particularly when don’t wait for AI.

### Implemented after final presentation

* Removed upload to AWS
* Added display of PDF rules file

## Version 0.4

### Overview

Work on various features: improve animation, cleanup display of PDF file (rules)

### Notes

* See if can simplify activity\_game.xml by making root viewGroup a *ConstraintLayout*.
* See if can make RelativeLayouts for players a separate class and layout XML file.
* Still using older Animation classes.  
  Consider using newer Animator classes, *View* exposes properties that can be accessed with ValueAnimator instances.

### Implemented

* Improved animation by making root ViewGroup of Game-activity a *RelativeLayout*.
* Improved animation by creating animated imageViews in GameActivity.onCreate() instead of just before animation.
* Made GameActivityAnimation hold a single imageView and be listener for that imageView. Have static methods to create 1 or more GameActivityAnimation instances.

## Version 1.0