



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΙΩΑΝΝΙΝΩΝ

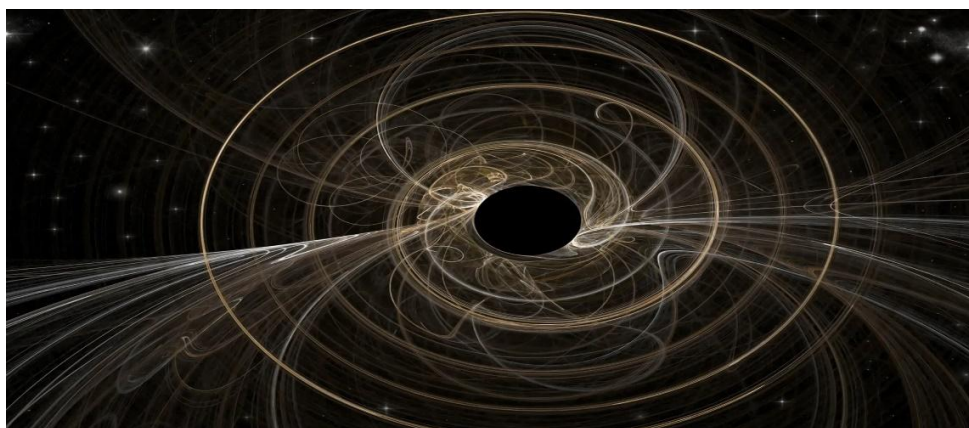
**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΩΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ANDROID ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ  
BLACK BODY**

Σωτήρης Ηλιού



Επιβλέπων: Ιωάννης Τσούλος

Αναπληρωτής Καθηγητής

Άρτα, Απρίλιος, 2023

**ANDROID APPLICATION IMPLEMENTATION FOR BLACK BODY  
PROBLEM**

## **Εγκρίθηκε από τριμελή εξεταστική επιτροπή**

Τόπος, Ημερομηνία

### **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Επιβλέπων καθηγητής

Όνομα Επίθετο,

2. Μέλος επιτροπής

Όνομα Επίθετο,

3. Μέλος επιτροπής

Όνομα Επίθετο,

© Ηλιού, Σωτήριος, 2023.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

## Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Σωτήρης, Ηλιού

Υπογραφή

A handwritten signature in blue ink, consisting of stylized letters and a long horizontal stroke extending to the right.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Είναι σημαντική η αναγνώριση της βοήθειας που έλαβε ο φοιτητής/ η φοιτήτρια κατά τη διάρκεια της προπαρασκευής της εργασίας του. Η βοήθεια μπορεί να είναι ακαδημαϊκή, τεχνική, γραμματειακή, διοικητική και προσωπική (π.χ. οικογένεια). Δεν υπερβαίνει τη μία παράγραφο.

*[Διαγράψτε αυτή τη σελίδα αν δεν τη χρειάζεστε.]*

## ΠΕΡΙΛΗΨΗ

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι η επίλυση του προβλήματος του Μέλανος Σώματος και την απεικόνιση του φάσματος της ακτινοβολία του. Μετά την διεξαγωγή της σκεπτικής ερευνάς, η λύση που προτάθηκε είναι η σχεδίαση και ανάπτυξη μιας Android εφαρμογής για την απεικόνιση του φάσματος της ακτινοβολίας του μέλανος σώματος.

Το αντικείμενο της διπλωματικής είναι ο σχεδιασμός και η ανάπτυξη μιας διαδραστική Android εφαρμογής για τον υπολογισμό και την αναπαράσταση της κυματομορφής του φάσματος της ακτινοβολίας του μέλαν σώματος που διαμορφώνεται ανάλογα με τη θερμοκρασία, καθώς και πληροφορίες σχετικά με το άστρο όπως ο φασματικός τύπος και την ακτίνα του.

**Λέξεις-κλειδιά:** φάσμα ακτινοβολίας μέλαν σώματος, πρόβλημα μέλαν σώματος, φασματικός τύπος αστέρα, εφαρμογή κινητών συσκευών.

## **ABSTRACT**

The purpose of this thesis is to solve the problem of the Black Body and to visualize the spectrum of its radiation. After conducting the conceptual research, the proposed solution is the design and development of an Android application for the visualization of the black body radiation spectrum.

The subject of the thesis is the design and development of an interactive Android application for the calculation and representation of the spectrum waveform of the blackbody radiation modulated by temperature, as well as information about the star such as the spectral type and radius of.

**Keywords:** black body radiation spectrum, black body problem, spectral type of a star, Android application.



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ .....	6
ΠΕΡΙΛΗΨΗ .....	7
ABSTRACT.....	8
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	9
1. Εισαγωγή.....	11
1.1 Ιστορία κινητών τηλεφώνων .....	11
1.2 Τι είναι το Android.....	13
1.3 Εφαρμογές Android .....	14
1.4 Εξέλιξη του Android .....	14
1.5 Αρχιτεκτονική του Android .....	18
1.5.1  Πυρήνας (Linux Kernel) .....	19
1.5.2  Αφαιρετικό Επίπεδο Υλικού (Hardware Abstraction Layer) .....	20
1.5.3  Βιβλιοθήκες.....	20
1.5.4  Η εικονική μηχανή Dalvik.....	22
1.5.5  Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime).....	24
1.5.6  Πλαίσιο Εφαρμογής (Application Framework) .....	25
1.6 Στο Εσωτερικό μιας Εφαρμογής του Android .....	26
1.6.1  Το αρχείο AndroidManifest.xml.....	26
1.6.2  Οι φάκελοι src & res.....	28
1.6.3  Οι υπόλοιποι φάκελοι του project.....	30
1.6.4  Δομικά Μέρη μιας Εφαρμογής .....	31
1.7 Σύγκριση με IOS .....	35
1.8 Ασφάλεια του Android.....	37
1.9 Το Πρόβλημα που επιλύει η εφαρμογή.....	38
2. Περιβάλλον Ανάπτυξης.....	38

2.1 Android Studio .....	40
2.2 Εγκατάσταση.....	41
2.3 Δημιουργία νέου Project .....	47
2.4 Πλοήγηση στο Project.....	51
2.5 Δημιουργία User Interface .....	52
2.6 Android Virtual Device (AVD) Manger .....	57
2.7 Εκτέλεση και Debugging .....	61
2.8 Προετοιμασία της Εφαρμογής για Κυκλοφορία .....	66
2.9 Παράδειγμα κώδικα σε Java και Kotlin .....	68
3. Η Προτεινόμενη Εφαρμογή .....	70
3.1 Θεωρητικό Πλαίσιο.....	70
3.2 Η Βιβλιοθήκη Graph View .....	72
3.3 Τμήματα της Εφαρμογής.....	72
3.4 Παρόμοια Λογισμικά .....	74
3.5 Μελλοντικές Προσθήκες.....	77
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	78

# 1. Εισαγωγή

Κυρίως κείμενο της εργασίας. Η δομή του κυρίως σώματος της εργασίας εξαρτάται από το είδος της εργασίας π.χ. πτυχιακή, διπλωματική εργασία ή διδακτορική διατριβή.

## 1.1 Ιστορία κινητών τηλεφώνων



Εικόνα 1.1 Εξέλιξη των κινητών τηλεφώνων

Η ιστορία των κινητών τηλεφώνων μπορεί να εντοπιστεί στις αρχές του 20ου αιώνα, όταν αναπτύχθηκαν για πρώτη φορά οι τεχνολογίες ασύρματης επικοινωνίας. Η έννοια του κινητού τηλεφώνου εισήχθη για πρώτη φορά στα τέλη της δεκαετίας του 1940 και στις αρχές της δεκαετίας του 1950, με την ανάπτυξη τηλεφώνων αυτοκινήτων που χρησιμοποιούσαν ραδιοσυχνότητες για τη μετάδοση σημάτων. Ωστόσο, μόλις τη δεκαετία του 1970 τα κινητά τηλέφωνα έγιναν ευρέως διαθέσιμα για δημόσια χρήση.

Το πρώτο κινητό τηλέφωνο που διατίθεται στο εμπόριο ήταν το Motorola DynaTAC 8000X, το οποίο κυκλοφόρησε το 1983. Αυτό το τηλέφωνο ήταν μεγάλο, βαρύ και ακριβό και μπορούσε να κάνει μόνο κλήσεις. Με τα χρόνια, τα κινητά τηλέφωνα έγιναν μικρότερα, ελαφρύτερα και πιο προσιτά. Αρχισαν επίσης να ενσωματώνουν νέες δυνατότητες όπως η αποστολή γραπτών μηνυμάτων, η περιήγηση στο Διαδίκτυο και οι δυνατότητες πολυμέσων.

Στις αρχές της δεκαετίας του 1990, η ανάπτυξη των ψηφιακών δικτύων κινητής τηλεφωνίας οδήγησε σε σημαντική αύξηση της δημοτικότητας των κινητών τηλεφώνων. Τα ψηφιακά δίκτυα επέτρεψαν την αποτελεσματικότερη χρήση των ραδιοσυχνοτήτων, η οποία με τη σειρά της οδήγησε σε καλύτερη ποιότητα κλήσεων και βελτιωμένη κάλυψη. Το 1992, το πρώτο smartphone παρουσιάστηκε από την IBM. Ονομαζόταν Simon και είχε οθόνη αφής και δυνατότητα αποστολής και λήψης φαξ, email και μηνυμάτων τηλεειδοποίησης.

Στα τέλη της δεκαετίας του 1990, η Nokia έγινε ο κυρίαρχος παίκτης στην αγορά κινητών τηλεφώνων, εν μέρει χάρη στη φιλική προς τον χρήστη διεπαφή και τον καινοτόμο σχεδιασμό της. Η εισαγωγή του Nokia 3210 το 1999, με τις προσαρμόσιμες προσόψεις του, έκανε τα κινητά τηλέφωνα αξεσουάρ μόδας αλλά και συσκευή επικοινωνίας. Στις αρχές της δεκαετίας του 2000, τα κινητά τηλέφωνα είχαν γίνει ουσιαστικό μέρος της καθημερινής ζωής για πολλούς ανθρώπους.

Η εισαγωγή των δικτύων κινητής τηλεφωνίας 3G στις αρχές της δεκαετίας του 2000 επέτρεψε ακόμη πιο προηγμένες λειτουργίες, όπως βιντεοκλήσεις και πρόσβαση στο Διαδίκτυο μέσω κινητού τηλεφώνου. Η κυκλοφορία του iPhone το 2007 έφερε επανάσταση στη βιομηχανία κινητών τηλεφώνων με το διαισθητικό περιβάλλον χρήστη και την πλήρη οθόνη αφής. Αυτό οδήγησε σε μια νέα εποχή smartphone με προηγμένες δυνατότητες, όπως πλοήγηση GPS, πληρωμές μέσω κινητού τηλεφώνου και επαυξημένη πραγματικότητα.

Σήμερα, τα κινητά τηλέφωνα αποτελούν αναπόσπαστο κομμάτι της σύγχρονης κοινωνίας, με δισεκατομμύρια ανθρώπους σε όλο τον κόσμο να τα χρησιμοποιούν καθημερινά. Έχουν αλλάξει τον τρόπο με τον οποίο επικοινωνούμε, εργαζόμαστε και έχουμε πρόσβαση σε πληροφορίες και συνεχίζουν να εξελίσσονται με γρήγορους ρυθμούς.<sup>1,2,3,4</sup>

## 1.2 Τι είναι το Android



*Εικόνα 1.2 Android OS*

Το λειτουργικό σύστημα Android αναπτύχθηκε από την Google και κυκλοφόρησε το 2008, ο πυρήνας του βασίζεται στο πύρινα του λειτουργικού συστήματος Linux και είναι ανοιχτού κώδικα, που σημαίνει ότι οποιοσδήποτε μπορεί να έχει πρόσβαση και να τροποποιήσει τον πηγαίο κώδικα του. Το λειτουργικό σύστημα Android παρέχει στους προγραμματιστές μια πλατφόρμα για τη δημιουργία εφαρμογών για κινητές συσκευές χρησιμοποιώντας το Android Software Development Kit (SDK). Το SDK περιλαμβάνει ένα σύνολο εργαλείων και API που επιτρέπουν στους προγραμματιστές να δημιουργούν εφαρμογές για την πλατφόρμα Android. Οι Native εφαρμογές Android μπορούν να γραφτούν σε γλώσσες προγραμματισμού όπως Java, Kotlin ή C++ και μπορούν να διανεμηθούν μέσω του Google Play Store ή άλλων καταστημάτων εφαρμογών.

Η διεπαφή χρήστη του λειτουργικού συστήματος Android βασίζεται σε ένα σύστημα εικονικών οθονών που ονομάζονται activities. Κάθε activity αντιπροσωπεύει μια ενιαία οθόνη με διεπαφή χρήστη και οι χρήστες μπορούν να πλοηγηθούν μεταξύ των activities σύροντας ή αγγίζοντας. Το λειτουργικό σύστημα Android υποστηρίζει επίσης γραφικά στοιχεία, τα οποία είναι μικρές εφαρμογές που μπορούν να τοποθετηθούν στην αρχική οθόνη για γρήγορη πρόσβαση.

Το λειτουργικό σύστημα Android περιλαμβάνει υποστήριξη για ένα ευρύ φάσμα επιλογών συνδεσιμότητας, όπως Wi-Fi, Bluetooth, BLE (Bluetooth Low Energy), NFC (Near Field Communication), GPS (Global Positioning System), και δίκτυα κινητής τηλεφωνίας (2G, 3G, 4G και 5G). Οι συσκευές Android μπορούν επίσης να λειτουργήσουν ως hotspot,

επιτρέποντας σε άλλες συσκευές να συνδέονται στο διαδίκτυο μέσω της σύνδεσης δεδομένων κινητής τηλεφωνίας της συσκευής. Το λειτουργικό σύστημα Android περιλαμβάνει επίσης υποστήριξη για μια σειρά ασύρματων τεχνολογιών όπως το Miracast, το οποίο επιτρέπει στους χρήστες να μεταδίδουν περιεχόμενο από τη συσκευή τους σε μια συμβατή τηλεόραση ή οθόνη και το Chromecast, το οποίο επιτρέπει στους χρήστες να μεταδίδουν περιεχόμενο από τη συσκευή τους σε μια τηλεόραση χρησιμοποιώντας μια Google Chromecast συσκευή.<sup>5,6,7,8,9,10</sup>

### **1.3 Εφαρμογές Android**

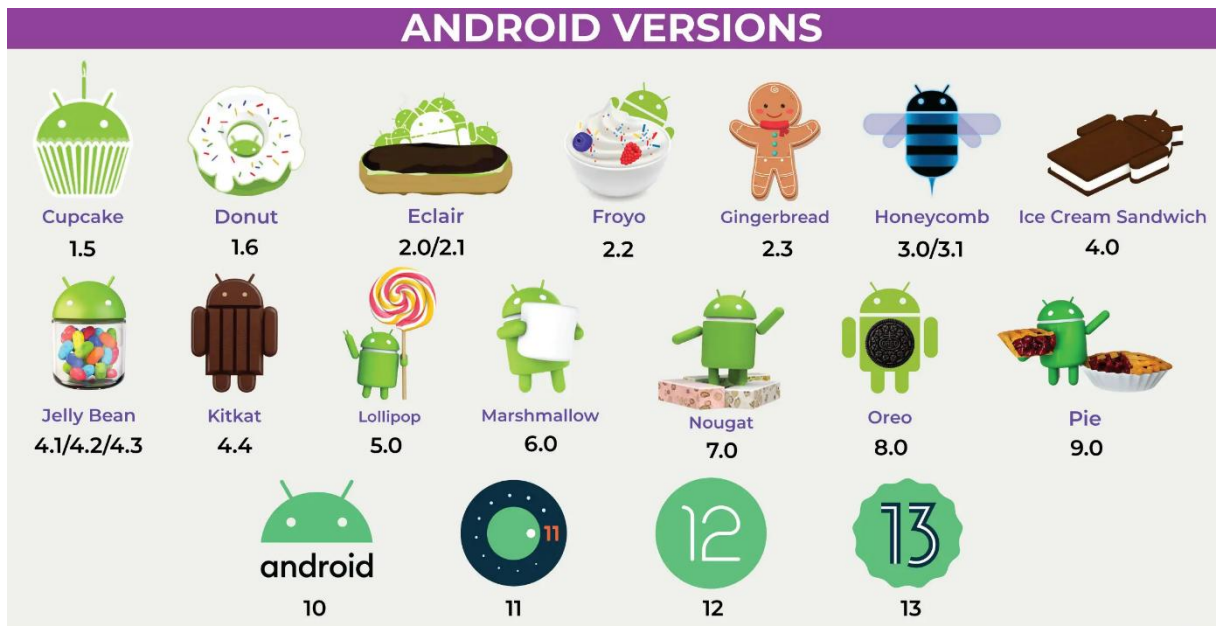
Οι εφαρμογές Android παρουσιάστηκαν για πρώτη φορά το 2008 με την κυκλοφορία του λειτουργικού συστήματος Android. Από τότε, η πλατφόρμα έχει εξελιχθεί και γίνεται το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα κινητής τηλεφωνίας στον κόσμο. Οι Native εφαρμογές Android αναπτύσσονται χρησιμοποιώντας Java ή Kotlin ως γλώσσες προγραμματισμού και μπορούν να ληφθούν επισήμα από το Google Play Store ή από άλλα ανεπίσημα καταστήματα εφαρμογών Android. Οι εφαρμογές Android προσφέρουν ένα ευρύ φάσμα λειτουργιών, όπως ειδοποιήσεις push, ενσωμάτωση μέσω κοινωνικής δικτύωσης και αγορές εντός εφαρμογής.

Για την ανάπτυξη των εφαρμογών Android, μπορούν να χρησιμοποιήσουν διάφορα ενσωματωμένα περιβάλλοντα ανάπτυξης (IDE) όπως το Android Studio, το Eclipse ή το IntelliJ IDEA. Το Android Studio συνιστάται από την Google και είναι το πιο ευρέως χρησιμοποιούμενο IDE για την ανάπτυξη εφαρμογών Android.

Το τελικό αποτέλεσμα της διαδικασίας ανάπτυξης εφαρμογών Android είναι ένα αρχείο APK (Android Package). Το αρχείο APK περιέχει όλους τους πόρους και τον κώδικα που απαιτούνται για την εγκατάσταση και την εκτέλεση της εφαρμογής σε μια συσκευή Android και είναι ψηφιακά υπογεγραμμένα από τον προγραμματιστή προκειμένου να διασφαλιστεί η αυθεντικότητά τους.

### **1.4 Εξέλιξη του Android**

Το λειτουργικό σύστημα έχει υποστεί σημαντική εξέλιξη από την έναρξή του το 2008. Έχει εξελιχθεί από ένα απλό λειτουργικό σύστημα σε μια ισχυρή πλατφόρμα που προσφέρει ένα ευρύ φάσμα δυνατοτήτων και λειτουργιών που εξελίσσονται σε κάθε νέα έκδοση του, επίσης έχει την ευελιξία και μπορεί να λειτουργήσει σε διάφορες συσκευές όπως Smartphone, tablet, τηλεοράσεις, αυτοκίνητα και φορητές συσκευές.



Εικόνα 1.4 Android OS Versions

Η πρώτη έκδοση του Android ήταν το Android 1.0 και κυκλοφόρησε τον Σεπτέμβριο του 2008 με την ονομασία Astro, ήταν σχεδιασμένο για συσκευές με οθόνη αφής και πρόσφερε κάποιες βασικές λειτουργίες όπως περιήγηση στο web, email, κατάσταση εφαρμογών για τη λήψη εφαρμογών και λήψη φωτογραφιών. Το Android 1.0 ήταν επίσης το πρώτο λειτουργικό σύστημα που εισήγαγε γραφικά στοιχεία, τα οποία επέτρεπαν στους χρήστες να έχουν πρόσβαση σε πληροφορίες χωρίς να ξεκινήσουν μια εφαρμογή.

Το Android 1.5, γνωστό και ως Cupcake, κυκλοφόρησε τον Απρίλιο του 2009 και ήταν η πρώτη έκδοση που έχει όνομα με θέμα το επιδόρπιο, περιείχε νέες δυνατότητες όπως πληκτρολόγιο οθόνης, γραφικά στοιχεία στην αρχική οθόνη, υποστήριξη συνδεσιμότητας μέσω Bluetooth και εγγραφή βίντεο.

Το Android 1.6, με την ονομασία Donut, κυκλοφόρησε το Σεπτέμβριο του 2009, οι νέες δυνατότητες περιλάμβαναν υποστήριξη για διαφορετικά μεγέθη και αναλύσεις οθόνης, φωνητική αναζήτηση και μετατροπή κειμένου σε ομιλία.

Το Android 2.0, γνωστό και ως Éclair, κυκλοφόρησε τον Οκτώβριο του 2009 και εισήγαγε νέες δυνατότητες όπως η Πλοήγηση στους Χάρτες Google, υποστήριξη πολλαπλών λογαριασμών και η ανανεωμένη διεπαφή χρήστη. Το Android 2.1, επίσης ήταν μέρος της οικογένειας Éclair, πρόσθεσε υποστήριξη για ζωντανές ταπετσαρίες και μια νέα εφαρμογή γκαλερί.

Το Android 2.2, ή Froyo, κυκλοφόρησε τον Μάιο του 2010 και έφερε σημαντικές βελτιώσεις στην απόδοση και πρόσθεσε υποστήριξη για το Adobe Flash, με το οποίο οι χρήστες Android μπορούσαν να απολαμβάνουν μια πιο πλούσια εμπειρία περιήγησης, συμπεριλαμβανομένης της αναπαραγωγής βίντεο, δια δραστικών παιχνιδιών και άλλου περιεχομένου πολυμέσων, δυνατότητα αποθήκευση εφαρμογών στη κάρτα μνήμης και υποστήριξη Wi-Fi Hotspot.

Το Android 2.3, ή Gingerbread, κυκλοφόρησε τον Δεκέμβριο του 2010, επικεντρώθηκε στη βελτίωση της διεπαφής χρήστη και στην προσθήκη νέων λειτουργιών όπως υποστήριξη NFC και βελτιωμένη λειτουργικότητα αντιγραφής-επικόλλησης.

Το Android 3.0, γνωστό και ως Honeycomb, κυκλοφόρησε τον Φεβρουάριο του 2011, ήταν η πρώτη έκδοση που σχεδιαστικέ αποκλειστικά για Tablets περιλάμβανε ένα επανασχεδιασμένο πληκτρολόγιο, υποστήριξη για πολυπύρηνους επεξεργαστές, και δυνατότητα για πολλαπλές καρτέλες στο πρόγραμμα περιήγησης για ευκολότερη χρήση του ιστού.

Το Android 4.0, γνωστό και ως Ice Cream Sandwich, κυκλοφόρησε τον Οκτώβριο του 2011, εισήγαγε μια νέα σχεδίαση διεπαφής χρήστη, που ονομάζεται Holo, η οποία έγινε η βάση για την τρέχουσα σχεδιαστική γλώσσα του Android.

Το Android 4.1 ή αλλιώς Jelly Bean κυκλοφόρησε τον Ιούνιο του 2012 με μια πιο βελτιωμένη απόδοση και έφερε νέες δυνατότητες όπως επεκτάσιμες ειδοποιήσεις και το Google Now.

Το Android 4.4 ή KitKat κυκλοφόρησε τον Σεπτέμβριο του 2013. Επικεντρώθηκε στη βελτίωση της απόδοσης και στη βελτιστοποίηση του λειτουργικού συστήματος ώστε να λειτουργεί σε συσκευές με λιγότερη μνήμη. Το KitKat εισήγαγε επίσης νέες δυνατότητες, όπως τη λειτουργία καθηλωτικής λειτουργίας, η οποία επέτρεπε στις εφαρμογές να αποκρύψουν τη γραμμή κατάστασης και τα κουμπιά πλοήγησης για μια πιο καθηλωτική εμπειρία.

Το Android 5, ή Lollipop, κυκλοφόρησε τον Νοέμβριο του 2014. Εισήγαγε ένα νέο τρόπο σχεδιασμού, που ονομάζεται Material Design, η οποία έφερε ένα πιο πολύχρωμο και ανταποκρινόμενο περιβάλλον χρήστη. Το Lollipop πρόσθεσε επίσης νέες δυνατότητες,



όπως υποστήριξη πολλών χρηστών, ειδοποιήσεις στην οθόνη κλειδώματος, υποστήριξη πολλαπλών καρτών SIM και βελτιωμένη διάρκεια ζωής της μπαταρίας.

Το Android 6 ή Marshmallow κυκλοφόρησε τον Οκτώβριο του 2015. Εισήγαγε νέες δυνατότητες όπως το Doze, το οποίο βοήθησε στη βελτίωση της διάρκειας ζωής της μπαταρίας θέτοντας τη συσκευή σε βαθύ ύπνο όταν δεν ήταν σε χρήση, στο Marshmallow παρουσιάστηκε επίσης υποστήριξη για αισθητήρες δακτυλικών αποτυπωμάτων και το USB-C type.

Το Android 7, ή Nougat, κυκλοφόρησε τον Αύγουστο του 2016. Έφερε νέες δυνατότητες, όπως πολλαπλές εργασίες σε split-screen και μια βελτιωμένη διαχείριση ειδοποιήσεων.

Το Android 8 ή Oreo κυκλοφόρησε τον Αύγουστο του 2017 και εισήγαγε νέες δυνατότητες, όπως τη λειτουργία εικόνας σε εικόνα και βελτιωμένη απόδοση.

Το Android 9, ή αλλιώς Pie, κυκλοφόρησε τον Αύγουστο του 2018, εισάγοντας νέες δυνατότητες, όπως η πλοήγηση με χειρονομίες και ένα νέο πίνακα ελέγχου ψηφιακής ευεξίας για να βοηθήσει τους χρήστες να διαχειρίζονται τη χρήση του τηλεφώνου τους.

Το Android 10 ή Q, κυκλοφόρησε τον Σεπτέμβριο του 2019 και έφερε νέες δυνατότητες, όπως την επιλογή για αλλαγής θέματος σε σκοτεινή λειτουργία σε όλο το σύστημα και βελτιωμένα στοιχεία ελέγχου απορρήτου και ήταν η πρώτη έκδοση που δεν έφερε πλέον όνομα με θέμα το επιδόρπιο.

Το Android 11 ή R και με εσωτερικό κωδικό όνομα Red Velvet Cake και με εσωτερικό κωδικό όνομα Snow Cone, κυκλοφόρησε το Σεπτέμβριο του 2020. Επικεντρώθηκε στη βελτίωση της επικοινωνίας και πρόσθεσε νέες δυνατότητες, όπως συννεφάκια συνομιλίας, που επέτρεπαν στους χρήστες να διατηρούν σημαντικές συνομιλίες ανοιχτές ακόμη και όταν χρησιμοποιούν άλλες εφαρμογές. Το Android 11.0 εισήγαγε επίσης νέα στοιχεία ελέγχου πολυμέσων και βελτιωμένες λειτουργίες ασφαλείας.

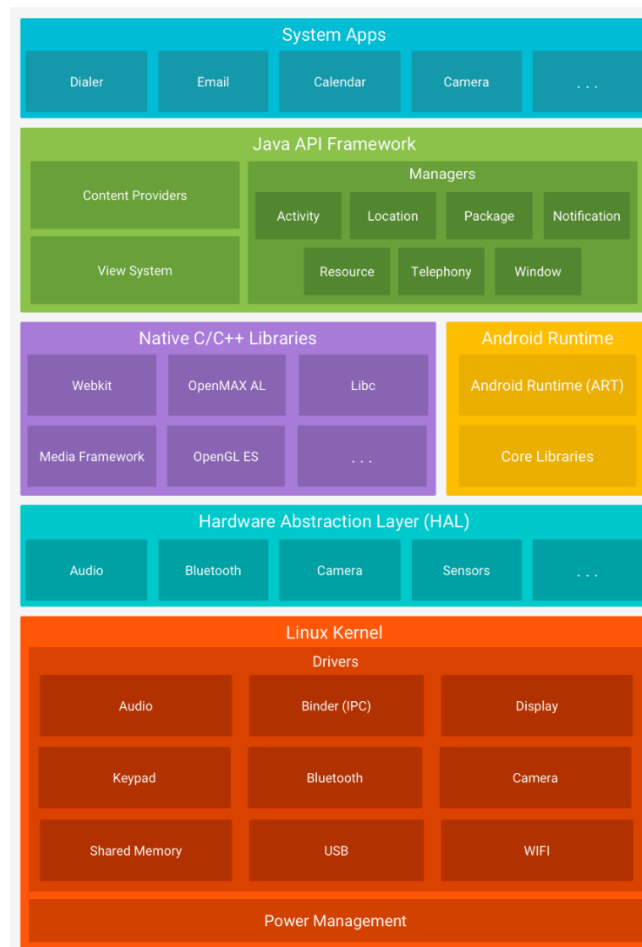
Το Android 12 ή S και με εσωτερικό κωδικό όνομα Snow Cone, κυκλοφόρησε τον Οκτώβριο του 2021 και έφερε νέα χαρακτηριστικά όπως το σύστημα σχεδίασης Materials You το οποίο αναπροσαρμόζει τα χρώματα του θέματος του χρήστη ανάλογα με την ταπετσαρία της αρχικής οθόνης, ένα ανανεωμένο σύστημα ειδοποιήσεων, βελτιωμένα στοιχεία πίνακα ελέγχου απορρήτου και νέες λειτουργίες για συσκευές με μεγάλες οθόνες όπως είναι τα Tablets και τα Foldable κινητά τηλεφώνά.

Το Android 13 ή T και με εσωτερικό κωδικό όνομα Tiramisu, κυκλοφόρησε τον Αύγουστο του 2022 και φέρει χαρακτηριστικά όπως BLE Audio, ανανεωμένη αθόρυβη λειτουργία συστήματος και υποστήριξη Fast Pair.

Το Android 14 ή U και με εσωτερικό κωδικό όνομα Upside Down Cake, αναμένεται να κυκλοφορήσει τον Αύγουστο του 2023.<sup>5,6,7,8,9,10,11</sup>

## 1.5 Αρχιτεκτονική του Android

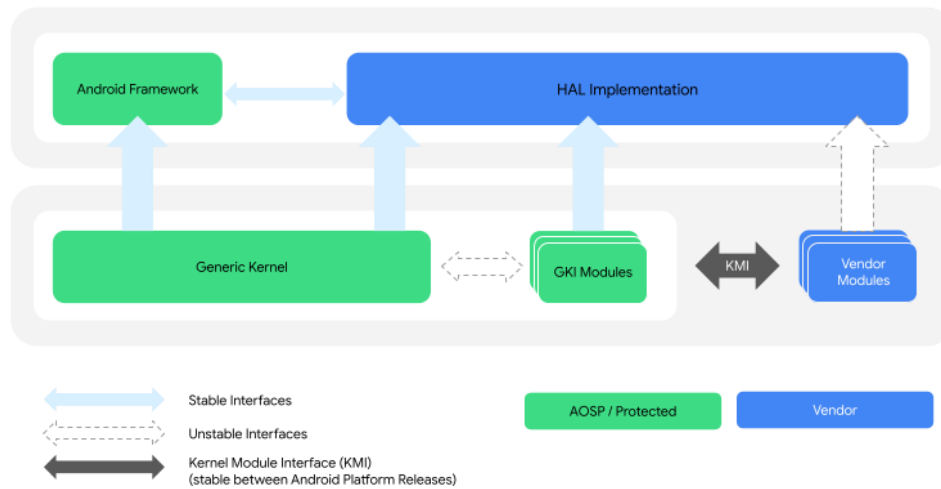
Το λειτουργικό σύστημα Android έχει σχεδιαστεί με μια πολυεπίπεδη αρχιτεκτονική που αποτελείται από πέντε κύρια επίπεδα: τον πυρήνα Linux, το αφαιρετικό επίπεδο υλικού, το επίπεδο χρόνου εκτέλεσης συστήματος που περιλαμβάνει τις βιβλιοθήκες και τον χρόνο εκτέλεσης Εφαρμογής (Android Runtime), το πλαίσιο εφαρμογών (Application Framework) και το επίπεδο εφαρμογών. Αυτή η αρχιτεκτονική επιτρέπει στο Android να παρέχει μια ευέλικτη και ισχυρή πλατφόρμα για την ανάπτυξη εφαρμογών για κινητές συσκευές.<sup>6,8,10,13,14,15</sup>



Εικόνα 1.5 Android OS Platform Architecture

### 1.5.1 Πυρήνας (Linux Kernel)

Ο πυρήνας του Android είναι ένας πυρήνας λειτουργικού συστήματος ανοιχτού κώδικα που βασίζεται στο Linux, είναι ένα κρίσιμο στοιχείο της αρχιτεκτονικής του λειτουργικού συστήματος Android, παρέχοντας βασικές λειτουργίες για τη λειτουργία του λειτουργικού συστήματος σε διάφορες συσκευές και αρχιτεκτονικές επεξεργαστών. Ο πυρήνας του Android OS είναι υπεύθυνος για τη διαχείριση διαφόρων πτυχών του υλικού και του λογισμικού της συσκευής, συμπεριλαμβανομένης της διαχείρισης διεργασιών, τη διαχείριση πόρων μνήμης, της διαχείρισης συσκευών εισόδου/εξόδου, διαχείριση συστημάτων αρχείων και τη αλληλεπίδραση με τα προγράμματα οδήγησης των συσκευών. Ένα από τα βασικά χαρακτηριστικά του πυρήνα του Android είναι η ικανότητά του να διαχειρίζεται πολλαπλές διεργασίες και νήματα ταυτόχρονα, επιτρέποντας την αποτελεσματική εκτέλεση πολλαπλών εργασιών και την κατανομή πόρων, παρέχει χαρακτηριστικά απομόνωσης διεργασιών και ασφάλειας για να διασφαλίσει ότι κάθε διεργασία εκτελείται ανεξάρτητα και με ασφάλεια χωρίς να παρεμβαίνει σε άλλες διεργασίες ή να διακυβεύει τη συνολική σταθερότητα του συστήματος. Επιπλέον παρέχει προγράμματα οδήγησης συσκευών που επιτρέπουν την επικοινωνία με διάφορα στοιχεία υλικού, όπως η οθόνη, ο ήχος, η κάμερα, το Wi-Fi και οι αισθητήρες. Αυτά τα προγράμματα οδήγησης συσκευών επιτρέπουν στο λειτουργικό σύστημα Android να διασυνδέεται με το υλικό της συσκευής, επιτρέποντας λειτουργίες όπως είσοδο αφής, αναπαραγωγή ήχου, λήψη κάμερας και ασύρματη επικοινωνία. Ο πυρήνας του λειτουργικού συστήματος Android περιλαμβάνει επίσης λειτουργίες διαχείρισης ενέργειας για τη βελτιστοποίηση της απόδοσης της συσκευής και της διάρκειας ζωής της μπαταρίας. Εφαρμόζει μηχανισμούς όπως κλιμάκωση συχνότητας CPU, δυναμική κλιμάκωση τάσης και αναστολή/συνέχιση σε όλο το σύστημα για τη αποτελεσματική διαχείριση της κατανάλωσης ενέργειας της συσκευής. Ο πυρήνας του λειτουργικού συστήματος Android περιλαμβάνει επίσης υποστήριξη για λειτουργίες ασφαλείας, όπως λίστες ελέγχου πρόσβασης, πολιτικές SELinux (Security-Enhanced Linux) και μηχανισμούς sandboxing, για την προστασία της συσκευής και των δεδομένων της από πιθανές απειλές ασφαλείας.



Εικόνα 1.5.1 Πυρήνας Android OS

### 1.5.2 Αφαιρετικό Επίπεδο Υλικού (Hardware Abstraction Layer)

Το Αφαιρετικό Επίπεδο Υλικού (HAL) είναι ένα κρίσιμο επίπεδο της αρχιτεκτονικής του Android που λειτουργεί ως γέφυρα μεταξύ των επιπέδων υλικού και λογισμικού. Παρέχει μια τυποποιημένη διεπαφή για την πρόσβαση και τον έλεγχο συσκευών υλικού, όπως αισθητήρες, κάμερες, συσκευές ήχου και άλλα. Το HAL υλοποιείται ως ένα σύνολο βιβλιοθηκών για συγκεκριμένες συσκευές που επικοινωνούν με το υλικό και εκθέτουν ένα API για τα στοιχεία υψηλότερου επιπέδου του λειτουργικού συστήματος Android. Αυτό επιτρέπει στις εφαρμογές Android να αλληλοεπιδρούν με διαφορετικές συσκευές υλικού με ενοποιημένο τρόπο, χωρίς να χρειάζεται να αντιμετωπίζουν τις περιπλοκές παραλλαγές υλικού. Το HAL είναι απαραίτητο για την ανεξαρτητοποίηση υλικού, επιτρέποντας στους κατασκευαστές συσκευών να αναπτύσσουν και να κυκλοφορούν συσκευές με διαφορετικές διαμορφώσεις υλικού, διατηρώντας παράλληλα τη συμβατότητα με το λειτουργικό σύστημα Android και τις εφαρμογές. Παρέχει επίσης ευελιξία στους προμηθευτές υλικού να ενημερώνουν ή να τροποποιούν τις βιβλιοθήκες τους για συγκεκριμένες συσκευές χωρίς να επηρεάζονται τα επίπεδα λογισμικού υψηλότερου επιπέδου, διευκολύνοντας την παράδοση ενημερώσεων και βελτιώσεων στο οικοσύστημα Android.<sup>14,17</sup>

### 1.5.3 Βιβλιοθήκες

Οι βιβλιοθήκες αποτελούν ουσιαστικό στοιχείο της αρχιτεκτονικής του λειτουργικού συστήματος Android, παρέχοντας πρόσθετες λειτουργίες στο σύστημα. Αυτές οι βιβλιοθήκες είναι γραμμένες σε C/C++ και περιλαμβάνονται στο λειτουργικό σύστημα

Android για να βελτιώσουν τις δυνατότητές του. Επιτρέπουν επίσης στις εφαρμογές να αξιοποιούν προηγμένες δυνατότητες, όπως υποστήριξη πολυμέσων, απόδοση Ιστού και τρισδιάστατα γραφικά, καθιστώντας το λειτουργικό σύστημα Android μια ευέλικτη πλατφόρμα για ένα ευρύ φάσμα εφαρμογών. Μερικές από τις βασικές βιβλιοθήκες στην αρχιτεκτονική του λειτουργικού συστήματος Android αναφέρονται παρακάτω.

**Surface Manager:** Αυτή η βιβλιοθήκη διαχειρίζεται το σύστημα απεικόνισης στην οθόνη, συμπεριλαμβανομένου του χειρισμού της απόδοσης γραφικών διεπαφών χρήστη (GUI), της διαχείρισης παραθύρων και του χειρισμού συμβάντων εισόδου, όπως είναι η αφή και οι χειρονομίες.

**Media Framework:** Η βιβλιοθήκη Media Framework παρέχει υποστήριξη πολυμέσων στο λειτουργικό σύστημα Android OS, συμπεριλαμβανομένης της αναπαραγωγής ήχου και βίντεο, εγγραφής και streaming. Περιλαμβάνει επίσης κωδικοποιητές για διάφορες μορφές ήχου και βίντεο, καθώς και API για τη διαχείριση περιεχομένου πολυμέσων.

**Βιβλιοθήκη SQLite:** Η βιβλιοθήκη SQLite είναι ένα ισχυρό και ελαφρύ σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που είναι ενσωματωμένο στο λειτουργικό σύστημα Android. Επιτρέπει στις εφαρμογές να αποθηκεύουν, να ανακτούν και να διαχειρίζονται δομημένα δεδομένα σε μια τοπική βάση δεδομένων, καθιστώντας το ένα κρίσιμο στοιχείο για τη διατήρηση των δεδομένων σε εφαρμογές Android.

**Βιβλιοθήκη WebKit:** Η βιβλιοθήκη WebKit χρησιμοποιείται για την υποστήριξη του Ιστού στο λειτουργικό σύστημα Android, επιτρέποντας την περιήγηση στον ιστό και τις εφαρμογές που βασίζονται στον ιστό. Περιλαμβάνει μια μηχανή υποστήριξης που μπορεί να απεικονίζει ιστοσελίδες, να χειρίζεται JavaScript και να υποστηρίζει HTML και CSS για τη προβολή περιεχομένου ιστού.

**Βιβλιοθήκη OpenGL ES:** Η βιβλιοθήκη OpenGL ES (Embedded Systems) είναι μια βιβλιοθήκη απόδοσης γραφικών που παρέχει υποστήριξη για τρισδιάστατα γραφικά στο λειτουργικό σύστημα Android. Επιτρέπει στις εφαρμογές την απόδοση τρισδιάστατων γραφικών, συμπεριλαμβανομένων παιχνιδιών και άλλων οπτικά εμπλουτισμένων εφαρμογών, σε συσκευές που υποστηρίζουν OpenGL ES.

**Βιβλιοθήκη Bionic:** Η βιβλιοθήκη Bionic είναι μια βιβλιοθήκη σε επίπεδο συστήματος που παρέχει ένα σύνολο λειτουργιών χαμηλού επιπέδου και κλήσεων συστήματος που

έχουν σχεδιαστεί ειδικά για το λειτουργικό σύστημα Android. Περιλαμβάνει λειτουργίες για διαχείριση μνήμης, threading , αρχεία I/O και άλλες λειτουργίες σε επίπεδο συστήματος. <sup>6,8,10,13,14</sup>

#### **1.5.4 Η εικονική μηχανή Dalvik**

Η εικονική μηχανή Dalvik (DVM) είναι ένα βασικό στοιχείο της αρχιτεκτονικής του λειτουργικού συστήματος Android, υπεύθυνη για την εκτέλεση εφαρμογών Android. Το DVM αναπτύχθηκε από την Google για την πλατφόρμα Android ως εναλλακτική λύση στο παραδοσιακό Java Virtual Machine (JVM) που χρησιμοποιείται σε εφαρμογές Java. Το DVM έχει σχεδιαστεί για να είναι εξαιρετικά αποδοτικό και βελτιστοποιημένο για κινητές συσκευές, οι οποίες έχουν περιορισμένους πόρους, όπως CPU, μνήμη και διάρκεια μπαταρίας. Σε αντίθεση με το JVM, το οποίο χρησιμοποιεί ένα μοντέλο εκτέλεσης που βασίζεται σε στοίβα, το DVM χρησιμοποιεί ένα μοντέλο εκτέλεσης που βασίζεται σε μητρώο, το οποίο επιτρέπει την πιο αποτελεσματική εκτέλεση κώδικα. Το DVM λειτουργεί σε bytecode Dalvik, ο οποίος είναι μια ειδική μορφή bytecode βελτιστοποιημένη για κινητές συσκευές. Το bytecode Dalvik δημιουργείται από τον πηγαίο κώδικα Java κατά τη διαδικασία δημιουργίας μιας εφαρμογής Android. Είναι μια συμπαγής και ελαφριά μορφή που έχει σχεδιαστεί για να εκτελείται αποτελεσματικά σε κινητές συσκευές με περιορισμένους πόρους.

Το Android Virtual Dalvik Machine (AVM) έχει πολλά χαρακτηριστικά που το καθιστούν κατάλληλο για την πλατφόρμα Android κάποια από αυτά περιγράφονται παρακάτω.

**Εκτέλεση βάσει μητρώου:** Το DVM χρησιμοποιεί ένα μοντέλο εκτέλεσης που βασίζεται σε μητρώο, το οποίο επιτρέπει ταχύτερη εκτέλεση κώδικα σε σύγκριση με το μοντέλο που βασίζεται σε στοίβα που χρησιμοποιείται στο JVM. Η εκτέλεση βάσει μητρώου εξαλείφει την επιβάρυνση του χειρισμού στοίβας και μειώνει τον αριθμό των λειτουργιών πρόσβασης στη μνήμη, με αποτέλεσμα βελτιωμένη απόδοση.

**Μεταγλώττιση Just-In-Time (JIT):** Το DVM περιλαμβάνει έναν μεταγλωττιστή JIT που μεταγλωττίζει δυναμικά τον bytecode Dalvik σε κώδικα μηχανής κατά τη διάρκεια του χρόνου εκτέλεσης. Αυτό επιτρέπει την ταχύτερη εκτέλεση του κώδικα αποφεύγοντας το βήμα ερμηνείας, το οποίο είναι συνήθως πιο αργό. Ο μεταγλωττιστής JIT βελτιστοποιεί τον κώδικα για τη συγκεκριμένη αρχιτεκτονική υλικού της συσκευής, με αποτέλεσμα βελτιωμένη απόδοση.

**App-Level Sandboxing:** Το DVM παρέχει sandboxing σε επίπεδο εφαρμογής, το οποίο διασφαλίζει ότι κάθε εφαρμογή Android εκτελείται στη δική της απομονωμένη διαδικασία με τη δική DVM. Αυτό παρέχει υψηλότερο επίπεδο ασφάλειας εμποδίζοντας μια εφαρμογή να έχει πρόσβαση στη μνήμη ή στους πόρους μιας άλλης εφαρμογής, ενισχύοντας τη συνολική ασφάλεια της πλατφόρμας Android.

**Συλλογή σκουπιδιών:** Το DVM περιλαμβάνει έναν συλλέκτη απορριμμάτων που διαχειρίζεται αυτόματα την κατανομή μνήμης για εφαρμογές Android. Ο συλλέκτης απορριμμάτων ελευθερώνει τη μνήμη που δεν χρησιμοποιείται πλέον, αποτρέποντας τις διαρροές μνήμης και βελτιώνοντας τη συνολική διαχείριση της μνήμης.

**Φόρτωση δυναμικής κλάσης:** Το DVM υποστηρίζει δυναμική φόρτωση κλάσης, η οποία επιτρέπει το φόρτωμα κλάσεων από τις εφαρμογές Android κατά το χρόνο εκτέλεσης. Αυτό επιτρέπει τη δυναμική εκτέλεση κώδικα και την επεκτασιμότητα σε εφαρμογές Android, επιτρέποντας την ευέλικτη και αρθρωτή ανάπτυξη εφαρμογών.

**Δια λειτουργικότητα με Java:** Ενώ το DVM χρησιμοποιεί διαφορετική μορφή bytecode από το JVM, έχει σχεδιαστεί για να είναι δια λειτουργικό με Java. Οι εφαρμογές Android μπορούν να αναπτυχθούν χρησιμοποιώντας γλώσσα προγραμματισμού Java και βιβλιοθήκες Java και μπορούν να αξιοποιήσουν τον υπάρχοντα κώδικα και εργαλεία Java, διευκολύνοντας τους προγραμματιστές Java να αναπτύξουν για την πλατφόρμα Android.

Το 2014 η Google ξεκινώντας με την έκδοση Android 5.0 Lollipop έχει μεταβεί στη χρήση του Android Runtime (ART) ως προεπιλεγμένο περιβάλλον χρόνου εκτέλεσης, αντικαθιστώντας το DVM. Το ART χρησιμοποιεί μεταγλώττιση ahead-of-time (AOT) αντί για μεταγλώττιση just-in-time (JIT) που χρησιμοποιείται στο DVM και έχει αρκετές βελτιώσεις απόδοσης σε σχέση με το DVM. Το ART συνεχίζει να υποστηρίζει τη χρήση του Dalvik bytecode για συμβατότητα με παλιότερες εφαρμογές Android που έχουν μεταγλωττιστεί χρησιμοποιώντας DVM, αλλά εισάγει επίσης νέες δυνατότητες όπως βελτιωμένη συλλογή απορριμμάτων, μειωμένο αποτύπωμα μνήμης και ταχύτερους χρόνους εκκίνησης εφαρμογών. Το ART έχει σχεδιαστεί για να είναι συμβατό με υπάρχουσες εφαρμογές Android γραμμένες σε Java ή άλλες γλώσσες JVM και παρέχει μια απρόσκοπτη μετάβαση για προγραμματιστές.<sup>6,8,18,19</sup>

### 1.5.5 Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime)

Το Android Runtime (ART) είναι το προεπιλεγμένο περιβάλλον χρόνου εκτέλεσης που χρησιμοποιείται σε συσκευές Android για την εκτέλεση εφαρμογών. Το ART είναι μια σημαντική ενημέρωση της παλαιότερης Dalvik Virtual Machine (DVM) και φέρνει αρκετές βελτιώσεις όσον αφορά την απόδοση, την αποτελεσματικότητα και τη συμβατότητα προς τα πίσω.

Το ART χρησιμοποιεί συλλογή Ahead-Of-Time (AOT) σε αντίθεση με τη συλλογή Just-In-Time (JIT) που χρησιμοποιείται στο DVM. Κατά τη διαδικασία εγκατάστασης της εφαρμογής, το bytecode της εφαρμογής μεταγλωττίζεται σε κώδικα μηχανής, ο οποίος στη συνέχεια εκτελείται απευθείας από την CPU της συσκευής κατά τη διάρκεια του χρόνου εκτέλεσης. Αυτό εξαλείφει την ανάγκη για ερμηνεία του bytecode κατά τη διάρκεια του χρόνου εκτέλεσης, με αποτέλεσμα ταχύτερους χρόνους εκκίνησης της εφαρμογής και βελτιωμένη συνολική απόδοση.

Το ART ενσωματώνει έναν βελτιωμένο μηχανισμό συλλογής σκουπιδιών που μειώνει τη συχνότητα και τη διάρκεια των παύσεων συλλογής σκουπιδιών, οδηγώντας σε ομαλότερη απόδοση της εφαρμογής και μειωμένο αποτύπωμα μνήμης. Τεχνικές όπως η συλλογή σκουπιδιών κατά γενιά και η ταυτόχρονη συλλογή σκουπιδιών χρησιμοποιούνται για την ελαχιστοποίηση του αντίκτυπου της συλλογής σκουπιδιών στην απόδοση της εφαρμογής.

Η ART εισάγει έναν αριθμό βελτιστοποιήσεων σε επίπεδο μεταγλωττιστή για τη βελτίωση της απόδοσης της εφαρμογής. Αυτές οι βελτιστοποιήσεις περιλαμβάνουν καλύτερη χρήση συνόλου εντολών, βελτιωμένο προφίλ και ενσωμάτωση κώδικα και καλύτερη υποστήριξη για επιτάχυνση υλικού. Αυτές οι βελτιστοποιήσεις έχουν ως αποτέλεσμα βελτιωμένη απόδοση και ανταπόκριση της εφαρμογής σε συσκευές Android.

Το ART διατηρεί συμβατότητα προς τα πίσω με υπάρχουσες εφαρμογές Android γραμμένες σε Java ή άλλες γλώσσες JVM. Συνεχίζει να υποστηρίζει τη χρήση του Dalvik bytecode, επιτρέποντας σε παλαιότερες εφαρμογές που έχουν μεταγλωττιστεί χρησιμοποιώντας DVM να εκτελούνται χωρίς προβλήματα σε νεότερες εκδόσεις Android που χρησιμοποιούν το ART ως περιβάλλον χρόνου εκτέλεσης.

Παρά τα πολυάριθμα πλεονεκτήματά της, η ART έχει επίσης ορισμένους περιορισμούς και προκλήσεις. Μια πρόκληση είναι ο αυξημένος χρόνος εγκατάστασης της εφαρμογής λόγω της μεταγλώττισης του AOT κατά τη διαδικασία εγκατάστασης. Ένας άλλος περιορισμός



είναι ο αυξημένος χώρος αποθήκευσης που απαιτείται για την αποθήκευση του μεταγλωττισμένου κώδικα μηχανής. Ωστόσο, αυτοί οι περιορισμοί αντισταθμίζονται από τα οφέλη και τις βελτιώσεις που φέρνει το ART στην πλατφόρμα του Android.<sup>6,8,10,13,14,18</sup>

### **1.5.6 Πλαίσιο Εφαρμογής (Application Framework)**

Το Πλαίσιο Εφαρμογής του Android είναι ένα ισχυρό και ολοκληρωμένο αρχιτεκτονικό πλαίσιο που παρέχει στους προγραμματιστές ένα πλούσιο σύνολο εργαλείων, βιβλιοθηκών και API (Application Programming Interface) για τη δημιουργία σύγχρονων εφαρμογών Android. Περιλαμβάνει διάφορες δυνατότητες και λειτουργίες, όπως δημιουργία διεπαφής χρήστη, διαχείριση πόρων, αποθήκευση δεδομένων και δικτύωση, πολυμέσα και αισθητήρες, προσβασιμότητα και εντοπισμός, ειδοποιήσεις και άδειες, κύκλος ζωής εφαρμογών και στοιχεία εφαρμογών, ασφάλεια και δικαιώματα εφαρμογών και δοκιμή και εντοπισμός σφαλμάτων. Η κατανόηση και η αποτελεσματική χρήση του Πλαισίου Εφαρμογής είναι ζωτικής σημασίας για την ανάπτυξη υψηλής ποιότητας, φιλικών προς το χρήστη και ασφαλών εφαρμογών Android που ανταποκρίνονται στις ανάγκες και τις προσδοκίες των σύγχρονων χρηστών.

Το Πλαίσιο Εφαρμογής περιλαμβάνει λειτουργίες για τη δημιουργία διεπαφών χρήστη σε εφαρμογές Android, όπως διατάξεις, προβολές και γραφικά στοιχεία. Παρέχει εργαλεία και βιβλιοθήκες για τη διαχείριση πόρων, συμπεριλαμβανομένων συμβολοσειρών, χρωμάτων, στυλ και θεμάτων, που επιτρέπουν έναν συνεπή και οπτικά ελκυστικό σχεδιασμό εφαρμογής. Περιλαμβάνει επίσης API για το χειρισμό της αποθήκευσης δεδομένων, συμπεριλαμβανομένων των βάσεων δεδομένων SQLite, των SharedPreferences και της πρόσβασης στο σύστημα αρχείων, επιτρέποντας στις εφαρμογές να αποθηκεύουν και να ανακτούν δεδομένα αποτελεσματικά.

Επιπλέον, το Πλαίσιο Εφαρμογής περιλαμβάνει API για χειρισμό δικτύωσης, όπως αιτήματα HTTP, επικοινωνία WebSocket και διαχείριση συνδεσιμότητας δικτύου. Αυτό επιτρέπει στις εφαρμογές να επικοινωνούν με απομακρυσμένους διακομιστές, να λαμβάνουν και να ανεβάζουν δεδομένα και να χειρίζονται συμβάντα που σχετίζονται με το δίκτυο. Το Πλαίσιο Εφαρμογής περιλαμβάνει επίσης υποστήριξη για λειτουργίες πολυμέσων, όπως αναπαραγωγή ήχου και βίντεο, λήψη πολυμέσων από την κάμερα ή το μικρόφωνο και χειρισμό εικόνων και γραφικών.

Ακόμα, το Πλαίσιο Εφαρμογής περιλαμβάνει API για χειρισμό αισθητήρων, όπως επιταχυνσιόμετρο, γυροσκόπιο και GPS, επιτρέποντας στις εφαρμογές να έχουν πρόσβαση και να χρησιμοποιούν δεδομένα αισθητήρων για διάφορους σκοπούς, όπως ανίχνευση κίνησης, παρακολούθηση τοποθεσίας και επαυξημένη πραγματικότητα. Το Πλαίσιο Εφαρμογής παρέχει επίσης δυνατότητες προσβασιμότητας, καθιστώντας τις εφαρμογές πιο περιεκτικές και προσβάσιμες σε χρήστες με αναπηρίες και υποστηρίζοντας διαφορετικές γλώσσες, περιοχές και πολιτισμούς.

Το Πλαίσιο Εφαρμογής περιλαμβάνει λειτουργίες για το χειρισμό ειδοποιήσεων, όπως η εμφάνιση ειδοποιήσεων στο δίσκο συστήματος, ο χειρισμός των αλληλεπιδράσεων των χρηστών με τις ειδοποιήσεις και η διαχείριση των καναλιών ειδοποιήσεων. Περιλαμβάνει επίσης API για το χειρισμό αδειών εφαρμογών, επιτρέποντας στις εφαρμογές να ζητούν και να διαχειρίζονται άδειες πρόσβασης σε ευαίσθητους πόρους και λειτουργίες της συσκευής, διασφαλίζοντας την ασφάλεια και το απόρρητο των δεδομένων χρήστη.

Επιπλέον, το Πλαίσιο Εφαρμογής περιλαμβάνει λειτουργίες για τη διαχείριση του κύκλου ζωής της εφαρμογής και των στοιχείων της εφαρμογής, όπως δραστηριότητες, υπηρεσίες, παρόχους περιεχομένου και δέκτες εκπομπής. Παρέχει API για το χειρισμό αλλαγών κατάστασης εφαρμογών και διαχείριση εργασιών, διασφαλίζοντας ότι οι εφαρμογές συμπεριφέρονται σωστά, αποτελεσματικά και με τρόπο φιλικό προς το χρήστη.

Το Πλαίσιο Εφαρμογής περιλαμβάνει επίσης διάφορες λειτουργίες ασφαλείας και API για τη διασφάλιση της ασφάλειας των εφαρμογών Android, όπως sandboxing, υπογραφή εφαρμογών και άδειες εφαρμογών. Παρέχει εργαλεία και βιβλιοθήκες για τη δοκιμή και τον εντοπισμό σφαλμάτων εφαρμογών Android, όπως το Android Debug Bridge (ADB), αυτοματοποιημένες δοκιμές, δοκιμές μονάδων και δοκιμές ενοποίησης, δίνοντας τη δυνατότητα στους προγραμματιστές να διασφαλίσουν την ποιότητα, τη σταθερότητα και την απόδοση των εφαρμογών τους. <sup>6,8,10,13</sup>

## **1.6 Στο Εσωτερικό μιας Εφαρμογής του Android**

### **1.6.1 Το αρχείο AndroidManifest.xml**

Το αρχείο Android Manifest χρησιμεύει ως σχέδιο για το λειτουργικό σύστημα Android για την κατανόηση και τη διαχείριση των διαφόρων πτυχών μιας εφαρμογής. Περιέχει σημαντικές πληροφορίες σχετικά με την εφαρμογή, όπως το όνομα του πακέτου, το όνομα της εφαρμογής, τον αριθμό έκδοσης, τα δικαιώματα, τις δραστηριότητες, τις υπηρεσίες,

τους δέκτες και τους παρόχους. Ο πρωταρχικός σκοπός του αρχείου Android Manifest είναι να δηλώνει τα στοιχεία μιας εφαρμογής και να ορίζει τα χαρακτηριστικά τους, κάτι που επιτρέπει στο λειτουργικό σύστημα Android να εκτελεί και να διαχειρίζεται σωστά την εφαρμογή. Το αρχείο Android Manifest παίζει επίσης σημαντικό ρόλο στη διασφάλιση της ασφάλειας και του απορρήτου μιας εφαρμογής και των χρηστών της. Η δήλωση αδειών στο αρχείο Android Manifest επιτρέπει στο χρήστη να κατανοήσει σε ποιους πόρους ή δυνατότητες απαιτεί πρόσβαση μια εφαρμογή και παρέχει στον χρήστη τη δυνατότητα να εκχωρήσει ή να αρνηθεί αυτές τις άδειες. Αυτό βοηθά στην προστασία των προσωπικών πληροφοριών του χρήστη και αποτρέπει τη μη εξουσιοδοτημένη πρόσβαση σε ευαίσθητους πόρους στη συσκευή Android.

Το αρχείο Android Manifest είναι ένα αρχείο XML (eXtensible Markup Language) που ακολουθεί μια συγκεκριμένη δομή. Αποτελείται από πολλά βασικά στοιχεία που καθορίζουν τις ιδιότητες και τα χαρακτηριστικά μιας εφαρμογής Android. Τα κύρια στοιχεία του αρχείου Android Manifest περιλαμβάνουν:

<manifest>: Αυτό είναι το ριζικό στοιχείο του αρχείου Android Manifest και καθορίζει τη συνολική δομή του αρχείου. Περιέχει διάφορα χαρακτηριστικά, όπως όνομα πακέτου, κωδικό έκδοσης και όνομα έκδοσης, τα οποία προσδιορίζουν μοναδικά την εφαρμογή.

<uses-permission>: Αυτό το στοιχείο χρησιμοποιείται για να δηλώσει τα δικαιώματα που απαιτεί η εφαρμογή για πρόσβαση σε ορισμένες λειτουργίες ή πόρους της συσκευής Android, όπως πρόσβαση στην κάμερα, στο GPS ή στο διαδίκτυο. Τα δικαιώματα είναι απαραίτητα για τη διασφάλιση της ασφάλειας και του απορρήτου της εφαρμογής και των χρηστών της.

<application>: Αυτό το στοιχείο ορίζει τις ιδιότητες και τα χαρακτηριστικά της εφαρμογής. Περιέχει διάφορα χαρακτηριστικά όπως το εικονίδιο, την ετικέτα, το θέμα, τις δραστηριότητες, τις υπηρεσίες, τους δέκτες και τους παρόχους της εφαρμογής. Καθορίζει επίσης τις ελάχιστες και στοχευμένες εκδόσεις SDK, οι οποίες καθορίζουν τη συμβατότητα της εφαρμογής με διαφορετικές συσκευές Android.

<activity>: Αυτό το στοιχείο χρησιμοποιείται για να δηλώσει τις δραστηριότητες (οθόνες ή στοιχεία διεπαφής χρήστη) της εφαρμογής. Περιέχει διάφορα χαρακτηριστικά, όπως το όνομα, την ετικέτα, τη λειτουργία εκκίνησης, τα φίλτρα, τα intents και το θέμα της δραστηριότητας. Οι δραστηριότητες είναι τα δομικά στοιχεία της διεπαφής χρήστη μιας

εφαρμογής Android και είναι υπεύθυνες για την παρουσίαση του περιεχομένου της εφαρμογής και την αλληλεπίδραση με τον χρήστη.

<service>: Αυτό το στοιχείο χρησιμοποιείται για τη δήλωση των υπηρεσιών (διεργασίες παρασκηνίου ή εργασίες) της εφαρμογής. Περιέχει διάφορα χαρακτηριστικά, όπως το όνομα, την ετικέτα, τη διαδικασία, τα φίλτρα και τα intents της υπηρεσίας. Οι υπηρεσίες χρησιμοποιούνται για την εκτέλεση εργασιών που δεν απαιτούν αλληλεπίδραση με τον χρήστη και μπορούν να εκτελούνται στο παρασκήνιο ακόμα και όταν η εφαρμογή δεν βρίσκεται στο προσκήνιο.

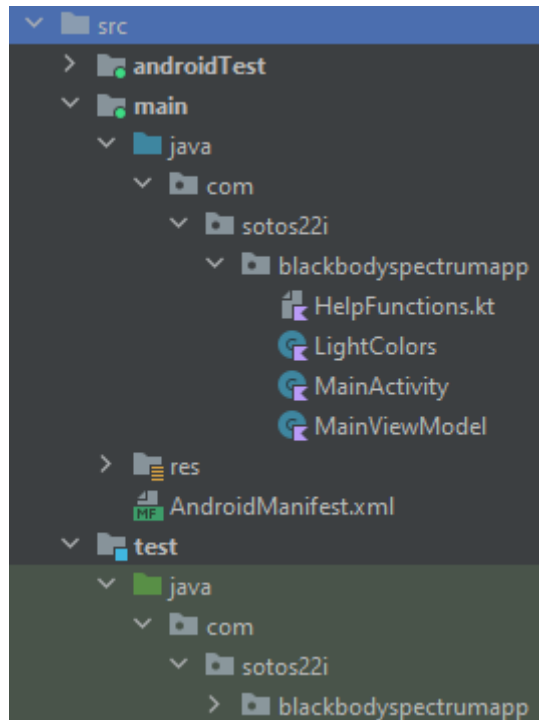
<receiver>: Αυτό το στοιχείο χρησιμοποιείται για να δηλώσει τους δέκτες (χειριστές συμβάντων ή ακροατές) της εφαρμογής. Περιέχει διάφορα χαρακτηριστικά όπως το όνομα του δέκτη, τα φίλτρα πρόθεσης και την προτεραιότητα. Οι δέκτες χρησιμοποιούνται για τη λήψη και το χειρισμό μηνυμάτων εκπομπής από το σύστημα ή άλλες εφαρμογές.

<provider>: Αυτό το στοιχείο χρησιμοποιείται για τη δήλωση των παρόχων περιεχομένου της εφαρμογής. Περιέχει διάφορα χαρακτηριστικά όπως το όνομα, την εξουσία και τα δεδομένα του παρόχου. Οι πάροχοι περιεχομένου χρησιμοποιούνται για τη διαχείριση και την κοινή χρήση δεδομένων εφαρμογών με άλλες εφαρμογές.<sup>20,21</sup>

### 1.6.2 Οι φάκελοι src & res

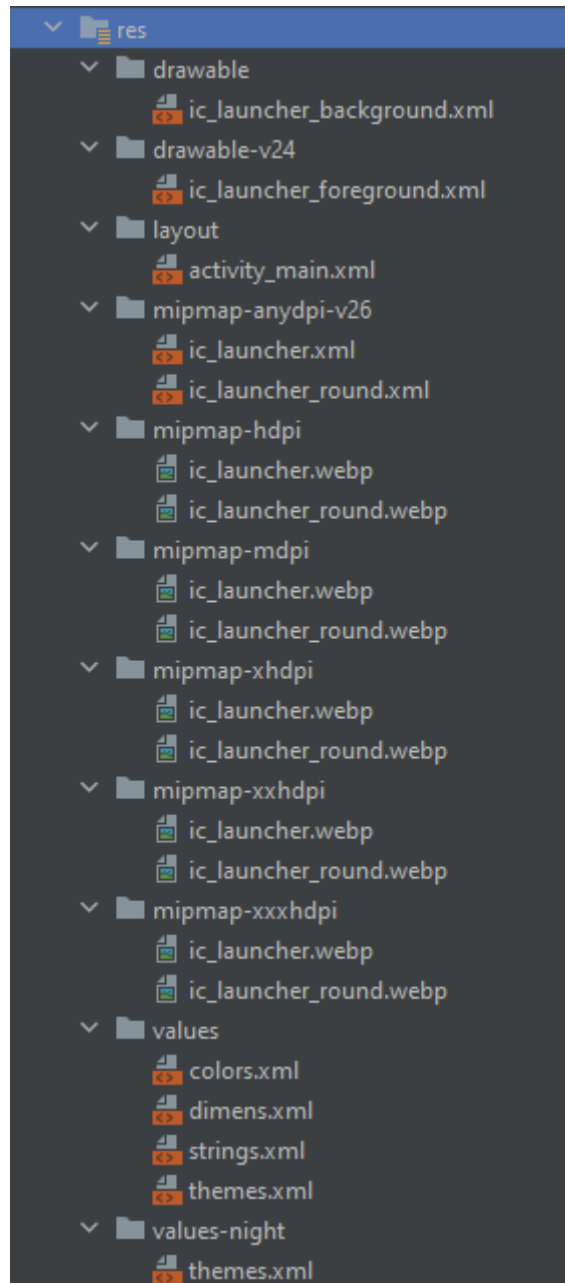
Οι φάκελοι "src" και "res" είναι κρίσιμοι κατάλογοι σε ένα έργο Android που περιέχουν τον πηγαίο κώδικα και τους πόρους της εφαρμογής, αντίστοιχα.

Ο φάκελος "src", συντομογραφία του "source", είναι ένας κατάλογος όπου οι προγραμματιστές γράφουν και αποθηκεύουν τον κώδικα Java ή Kotlin που συνθέτει τη λογική και τη λειτουργικότητα της εφαρμογής Android. Βρίσκεται στο ριζικό επίπεδο του έργου και είναι οργανωμένος σε ένα ή περισσότερα πακέτα, τα οποία είναι κατάλογοι που δομούν τον πηγαίο κώδικα με ιεραρχικό τρόπο. Ο φάκελος "src" περιέχει συνήθως το κύριο πακέτο, το οποίο έχει το όνομα του πακέτου της εφαρμογής που ορίζεται στο αρχείο δήλωσης (AndroidManifest) της εφαρμογής. Το κύριο πακέτο περιέχει τη βασική λογική της εφαρμογής, συμπεριλαμβανομένων δραστηριοτήτων, υπηρεσιών, δεκτών εκπομπής και άλλων στοιχείων που συνθέτουν τη διεπαφή χρήστη και τη λειτουργικότητα της εφαρμογής. Ο φάκελος "src" μπορεί επίσης να περιλαμβάνει πρόσθετα πακέτα για δοκιμή, κώδικα βιβλιοθήκης και πόρους.



Εικόνα 1.6.2.1 src folder

Ο φάκελος "res" είναι ένας κατάλογος που περιέχει πόρους που χρησιμοποιούνται στην εφαρμογή Android, όπως αρχεία XML, εικόνες, διατάξεις και άλλα στοιχεία. Βρίσκεται επίσης στο ριζικό επίπεδο του έργου και οργανώνεται σε υποκαταλόγους με βάση τον τύπο των πόρων. Ο φάκελος "res" συνήθως περιέχει υποκαταλόγους όπως "res/layout" για αρχεία διάταξης XML που ορίζουν την οπτική εμφάνιση της διεπαφής χρήστη της εφαρμογής, "res/drawable" για αρχεία εικόνας που χρησιμοποιούνται ως γραφικά στην εφαρμογή, "res/τιμές" για αρχεία XML που ορίζουν τιμές όπως συμβολοσειρές, χρώματα και διαστάσεις που χρησιμοποιούνται στην εφαρμογή και άλλους καταλόγους για διαφορετικούς τύπους πόρων, όπως κινούμενα σχέδια, γραμματοσειρές και στοιχεία μενού.<sup>22</sup>



Εικόνα 1.6.2.1 res folder

### 1.6.3 Οι υπόλοιποι φάκελοι του project

Ένα έργο Android αποτελείται από πολλούς φακέλους που οργανώνουν διάφορα στοιχεία της διαδικασίας ανάπτυξης εφαρμογών. Αυτοί οι φάκελοι διαδραματίζουν κρίσιμο ρόλο στη δομή και τη διαχείριση των διαφορετικών πτυχών ενός έργου Android. Η δομή του κύριου φακέλου ενός έργου Android περιλαμβάνει συνήθως τους ακόλουθους φακέλους.

**Φάκελος "app":** Ο φάκελος "app" είναι η καρδιά ενός έργου Android και περιέχει τον πηγαίο κώδικα, τους πόρους και τα αρχεία διαμόρφωσης για τα modules της εφαρμογής. Είναι ο κύριος φάκελος όπου οι προγραμματιστές γράφουν τον κώδικα Java ή Kotlin για

τις δραστηριότητες, τις υπηρεσίες, τους δέκτες εκπομπής και άλλα στοιχεία της εφαρμογής. Ο φάκελος "app" περιέχει επίσης τον φάκελο "res", ο οποίος φιλοξενεί τους πόρους της εφαρμογής, όπως αρχεία διάταξης, εικόνες, τιμές και άλλα στοιχεία.

**Φάκελος "build":** Ο φάκελος "build" δημιουργείται αυτόματα από το σύστημα κατασκευής του Android και περιέχει τα ενδιάμεσα αρχεία και τα αρχεία εξόδου που δημιουργούνται κατά τη διαδικασία κατασκευής. Αυτό περιλαμβάνει μεταγλωττισμένο κώδικα Java ή Kotlin, πόρους και άλλα αρχεία κατασκευής. Τα περιεχόμενα του φακέλου "build" είναι προσωρινά και μπορούν να αναδημιουργηθούν κατά τη διαδικασία κατασκευής της εφαρμογής.

**Φάκελος "gradle":** Ο φάκελος "gradle" περιέχει τα αρχεία διαμόρφωσης Gradle build για τη λειτουργική μονάδα εφαρμογής. Το Gradle είναι το σύστημα κατασκευής που χρησιμοποιείται σε έργα Android και ο φάκελος "gradle" φιλοξενεί το αρχείο build.gradle, το οποίο καθορίζει τις ρυθμίσεις κατασκευής, τις εξαρτήσεις και τις προσθήκες για τη λειτουργική μονάδα εφαρμογής.

**Φάκελος "libs":** Ο φάκελος "libs" χρησιμοποιείται για την αποθήκευση εξωτερικών αρχείων βιβλιοθήκης, όπως αρχεία JAR, που χρησιμοποιούνται στην εφαρμογή. Αυτά τα αρχεία βιβλιοθήκης μπορούν να συμπεριληφθούν στο αρχείο build.gradle της εφαρμογής για να προσθέσουν επιπλέον λειτουργίες στην εφαρμογή.

**Φάκελος "test":** Ο φάκελος "test" χρησιμοποιείται για την εγγραφή δοκιμαστικών περιπτώσεων για τον κώδικα της εφαρμογής. Περιέχει αρχεία πηγαίου κώδικα Java ή Kotlin για τη σύνταξη δοκιμών μονάδας (unit test), οργανωτικών (instrumented) δοκιμών και άλλους τύπους δοκιμών για τη διασφάλιση της ποιότητας και της αξιοπιστίας της εφαρμογής.<sup>22</sup>

#### 1.6.4 Δομικά Μέρη μιας Εφαρμογής

Τα βασικά Δομικά Μέρη που συνεργάζονται για να δημιουργήσουν μιας λειτουργικής εφαρμογής είναι οι δραστηριότητες, οι υπηρεσίες, οι δέκτες εκπομπής, οι πάροχοι περιεχομένου, οι προθέσεις, η διεπαφή χρήστη, οι πόροι, το αρχείο AndroidManifest, το σύστημα Gradle Build και τα Εργαλεία εντοπισμού σφαλμάτων και δοκιμών. Αυτά τα στοιχεία παρέχουν την απαραίτητη λειτουργικότητα, διεπαφή χρήστη και επικοινωνία μεταξύ των στοιχείων της εφαρμογής, διασφαλίζοντας μια ομαλή και διαδραστική εμπειρία χρήστη.

## **Δραστηριότητα (Activity)**

Μια Δραστηριότητα είναι ένα βασικό στοιχείο μιας εφαρμογής Android που αντιπροσωπεύει μια ενιαία οθόνη με διεπαφή χρήστη. Λειτουργεί ως το σημείο εισόδου για τις αλληλεπιδράσεις των χρηστών και είναι υπεύθυνο για την παρουσίαση πληροφοριών στον χρήστη και τη λήψη πληροφοριών. Μια Δραστηριότητα μπορεί να περιέχει διάφορα στοιχεία διεπαφής χρήστη, όπως προβολές, γραφικά στοιχεία και αρχεία διάταξης, για τη δημιουργία της οπτικής διεπαφής της εφαρμογής. Οι δραστηριότητες μπορούν επίσης να αλληλεπιδράσουν με άλλες δραστηριότητες εντός της ίδιας εφαρμογής ή με εξωτερικά στοιχεία μέσω του Intents.

## **Υπηρεσία (Service)**

Μια Υπηρεσία είναι ένα στοιχείο παρασκηνίου που εκτελείται χωρίς διεπαφή χρήστη και εκτελεί μακροχρόνιες εργασίες ή χειρίζεται λειτουργίες παρασκηνίου. Χρησιμοποιείται για την εκτέλεση εργασιών που θα πρέπει να συνεχιστούν ακόμα και όταν η εφαρμογή δεν χρησιμοποιείται ενεργά, όπως λήψη δεδομένων, αναπαραγωγή μουσικής ή επεξεργασία ειδοποιήσεων. Οι υπηρεσίες μπορούν να εκτελούνται με την ίδια διαδικασία με την εφαρμογή ή σε ξεχωριστή διαδικασία και μπορούν να ξεκινήσουν, να σταματήσουν ή να συνδεθούν με άλλα στοιχεία της εφαρμογής.

## **Δέκτης Εκπομπής (BroadcastReceiver)**

Ο Δέκτης Εκπομπής είναι ένα στοιχείο που μπορεί να ακούσει μηνύματα εκπομπής από παντού και να απαντήσει σε αυτά. Μπορεί να λαμβάνει και να απαντά σε συμβάντα, όπως εισερχόμενες κλήσεις, μηνύματα SMS ή αλλαγές στάθμης μπαταρίας, ακόμη και όταν η εφαρμογή δεν εκτελείται. Οι δέκτες εκπομπής μπορούν να εγγραφούν στο αρχείο AndroidManifest της εφαρμογής ή να εγγραφούν δυναμικά κατά το χρόνο εκτέλεσης. Χρησιμοποιούνται για την εκτέλεση ενεργειών ή την ενεργοποίηση άλλων στοιχείων που βασίζονται σε συμβάντα του συστήματος.

## **Πάροχος περιεχομένου (Content Provider)**

Ο πάροχος περιεχομένου είναι ένα στοιχείο που διαχειρίζεται την πρόσβαση σε ένα δομημένο σύνολο δεδομένων, όπως μια βάση δεδομένων, και επιτρέπει την κοινή χρήση δεδομένων μεταξύ εφαρμογών ή στοιχείων εντός μιας εφαρμογής. Παρέχει ένα επίπεδο αφαιρετικότητας για αλληλεπίδραση με δεδομένα και επιτρέπει σε άλλες εφαρμογές ή



στοιχεία να υποβάλλουν ερωτήματα, να εισάγουν, να ενημερώνουν ή να διαγράφουν δεδομένα. Οι πάροχοι περιεχομένου χρησιμοποιούνται για την ασφαλή κοινή χρήση δεδομένων μεταξύ εφαρμογών, την επιβολή αδειών πρόσβασης δεδομένων και τη διαχείριση της αποθήκευσης και ανάκτησης δεδομένων.

## **Intent**

Το Intent είναι ένας μηχανισμός ανταλλαγής μηνυμάτων που επιτρέπει σε στοιχεία εντός μιας εφαρμογής ή μεταξύ εφαρμογών να επικοινωνούν. Χρησιμοποιείται για την εκκίνηση δραστηριοτήτων, υπηρεσιών ή δεκτών εκπομπής και μπορεί επίσης να μεταφέρει δεδομένα μεταξύ των στοιχείων. Τα Intents μπορεί να είναι σαφείς, προσδιορίζοντας το ακριβές στοιχείο που θα ξεκινήσει ή αόριστοι επιτρέποντας στο σύστημα να επιλέξει το κατάλληλο στοιχείο με βάση τη δράση, τα δεδομένα ή την κατηγορία του Intent. Τα Intents επιτρέπουν την επικοινωνία και τον συντονισμό μεταξύ διαφορετικών στοιχείων μιας εφαρμογής ή μεταξύ διαφορετικών εφαρμογών.

## **Διεπαφή χρήστη (UI)**

Η διεπαφή χρήστη είναι το οπτικό μέρος μιας εφαρμογής Android που επιτρέπει στους χρήστες να αλληλεπιδρούν με την εφαρμογή. Περιλαμβάνει αρχεία διάταξης, προβολές και γραφικά στοιχεία που χρησιμοποιούνται για τη δημιουργία των οπτικών στοιχείων της εφαρμογής. Οι προβολές είναι στοιχεία διεπαφής χρήστη, όπως κουμπιά, πλαίσια κειμένου και εικόνες, ενώ οι διατάξεις είναι κοντέινερ που περιέχουν προβολές και καθορίζουν τη δομή και την εμφάνιση της διεπαφής χρήστη. Τα γραφικά στοιχεία είναι επαναχρησιμοποιήσιμα στοιχεία διεπαφής χρήστη, όπως λίστες, πλέγματα και αναπτυσσόμενες λίστες, που παρέχουν κοινές λειτουργίες διεπαφής χρήστη. Η διεπαφή χρήστη είναι υπεύθυνη για την παρουσίαση πληροφοριών στον χρήστη, τη λήψη πληροφοριών και την παροχή μιας ομαλής και διαδραστικής εμπειρίας χρήστη.

## **Πόροι (Resources)**

Οι πόροι είναι στοιχεία, όπως εικόνες, συμβολοσειρές και αρχεία XML, που χρησιμοποιούνται για την παροχή περιεχομένου και διαμόρφωσης για μια εφαρμογή. Είναι ξεχωριστά από τον κώδικα εφαρμογής και μπορούν να χρησιμοποιηθούν σε αρχεία κώδικα ή XML. Οι πόροι επιτρέπουν τον διαχωρισμό του περιεχομένου της εφαρμογής από τη λογική της εφαρμογής, διευκολύνοντας τη διαχείριση και την ενημέρωση του

περιεχομένου της εφαρμογής. Οι πόροι αποθηκεύονται στον κατάλογο πόρων της εφαρμογής και χρησιμοποιούνται σε αρχεία κώδικα ή XML χρησιμοποιώντας μοναδικά αναγνωριστικά.

### **Αρχείο Μανιφέστου (AndroidManifest)**

Το αρχείο μανιφέστου είναι ένα ειδικό αρχείο XML που περιέχει βασικές πληροφορίες σχετικά με την εφαρμογή Android. Περιλαμβάνει λεπτομέρειες όπως το όνομα πακέτου της εφαρμογής, το εικονίδιο της εφαρμογής, τις άδειες εφαρμογής, τα στοιχεία εφαρμογής (δραστηριότητες, υπηρεσίες, δέκτες εκπομπής) και απαιτούμενες λειτουργίες υλικού. Το αρχείο AndroidManifest χρησιμεύει ως σχέδιο για το σύστημα Android για να κατανοήσει και να ρυθμίσει σωστά την εφαρμογή. Καθορίζει τα σημεία εισόδου της εφαρμογής, τα στοιχεία της και τις σχέσεις τους, καθώς και τα δικαιώματα που απαιτούνται για την πρόσβαση σε ορισμένες λειτουργίες ή δεδομένα.

### **Σύστημα Κατασκευής Gradle**

Το Σύστημα Κατασκευής Gradle είναι ένα ισχυρό εργαλείο αυτοματοποιημένης κατασκευής που χρησιμοποιείται για τη δημιουργία, τη δοκιμή και τη δημιουργία πακέτου εφαρμογών Android. Βασίζεται στη Groovy, μια δυναμική γλώσσα δέσμης ενεργειών και επιτρέπει στους προγραμματιστές να ορίζουν τις διαμορφώσεις, τις εξαρτήσεις και τις εργασίες της εφαρμογής με ευέλικτο και επεκτάσιμο τρόπο. Το Gradle αυτοματοποιεί τη διαδικασία δημιουργίας του αρχείου APK (Android Package) της εφαρμογής, το οποίο είναι το τελικό εκτελέσιμο αρχείο που εγκαθίσταται σε συσκευές Android. Επιτρέπει επίσης την εύκολη διαχείριση βιβλιοθηκών, πόρων και διαμορφώσεων εφαρμογών τρίτων.

### **Εργαλεία εντοπισμού σφαλμάτων και δοκιμών**

Τα Εργαλεία εντοπισμού σφαλμάτων και δοκιμών είναι βασικά μέρη της διαδικασίας ανάπτυξης εφαρμογών για να διασφαλιστεί ότι η εφαρμογή λειτουργεί σωστά και πληροί τις επιθυμητές απαιτήσεις. Το Android παρέχει μια ποικιλία εργαλείων για εντοπισμό σφαλμάτων και δοκιμών, όπως το ενσωματωμένο πρόγραμμα εντοπισμού σφαλμάτων του Android Studio, το Logcat για την καταγραφή και τον εντοπισμό μηνυμάτων σφάλματος και διάφορα πλαίσια δοκιμών όπως το JUnit για δοκιμή μονάδας (unit testing), το Espresso για τη δοκιμή διεπαφής χρήστη (UI testing) και το Mockito για την προσομοίωση ψεύτικων εξαρτήσεων (mocking). Αυτά τα εργαλεία βοηθούν τους προγραμματιστές να

εντοπίζουν και να διορθώνουν σφάλματα, να ελέγχουν τη λειτουργικότητα της εφαρμογής και να διασφαλίζουν τη σταθερότητα και την απόδοση της εφαρμογής.<sup>23</sup>

## 1.7 Σύγκριση με IOS



Εικόνα 1.7 Android vs iOS

Τα λειτουργικά συστήματα κινητής τηλεφωνίας διαδραματίζουν κρίσιμο ρόλο στη σύγχρονη ψηφιακή εποχή, τροφοδοτώντας δισεκατομμύρια smartphones και tablets παγκοσμίως. Μεταξύ των πολυάριθμων διαθέσιμων λειτουργικών συστημάτων για κινητά, δύο κυρίαρχοι παίκτες ξεχωρίζουν, το λειτουργικό σύστημα Android, που αναπτύχθηκε από την Google και το iOS, που αναπτύχθηκε από την Apple.

Τόσο το λειτουργικό σύστημα Android όσο και το iOS έχουν ξεχωριστές σχεδιαστικές φιλοσοφίες και εμπειρίες χρήστη. Το λειτουργικό σύστημα Android ακολουθεί μια πιο ανοιχτή και προσαρμόσιμη προσέγγιση, επιτρέποντας στους κατασκευαστές να προσαρμόζουν τη διεπαφή χρήστη και δίνοντας στους χρήστες περισσότερο έλεγχο της εμφάνισης, των ρυθμίσεων και της αρχικής οθόνης της συσκευής τους. Από την άλλη πλευρά, το iOS ακολουθεί μια πιο κλειστή και τυποποιημένη προσέγγιση, προσφέροντας μια συνεπή και βελτιωμένη εμπειρία χρήστη σε όλες τις συσκευές Apple με έμφαση στην απλότητα, την ευκολία στη χρήση και τον αισθητικό σχεδιασμό.

Τόσο το λειτουργικό σύστημα Android όσο και το iOS διαθέτουν ισχυρά οικοσυστήματα εφαρμογών, προσφέροντας μια μεγάλη γκάμα εφαρμογών για να επιλέξουν οι χρήστες. Ωστόσο, υπάρχουν ορισμένες βασικές διαφορές μεταξύ των δύο. Το λειτουργικό σύστημα Android έχει μεγαλύτερη επιλογή εφαρμογών λόγω της ανοιχτής φύσης του, επιτρέποντας στους προγραμματιστές να διανέμουν εφαρμογές μέσω πολλών καταστημάτων εφαρμογών και να προσεγγίζουν ένα ευρύτερο κοινό. Από την άλλη πλευρά, το iOS έχει μια πιο

επιμελημένη επιλογή εφαρμογών στο Apple App Store, με μια πιο αυστηρή διαδικασία ελέγχου εφαρμογών που διασφαλίζει υψηλότερη ποιότητα εφαρμογών με λιγότερους κινδύνους ασφαλείας.

Η ασφάλεια και το απόρρητο είναι κρίσιμα ζητήματα για τις κινητές συσκευές. Το λειτουργικό σύστημα Android και το iOS έχουν διαφορετικές προσεγγίσεις για την ασφάλεια και το απόρρητο. Η φύση ανοιχτού κώδικα και η διανομή εφαρμογών του λειτουργικού συστήματος Android από πηγές τρίτων μπορεί να το καταστήσει πιο ευάλωτο σε απειλές ασφαλείας και κακόβουλο λογισμικό. Ωστόσο, η Google έχει εφαρμόσει διάφορα μέτρα ασφαλείας, όπως το Google Play Protect, το sandboxing εφαρμογών και τακτικές ενημερώσεις ασφαλείας για να μετριάσει αυτούς τους κινδύνους. Από την άλλη πλευρά, ο κλειστός χαρακτήρας του iOS και η αυστηρή διαδικασία ελέγχου εφαρμογών στο Apple App Store παρέχουν ένα πιο ασφαλές περιβάλλον για τους χρήστες, με λιγότερους κινδύνους ασφαλείας.

Μια άλλη σημαντική διαφορά μεταξύ του λειτουργικού συστήματος Android και του iOS είναι η ποικιλομορφία της συσκευής και το εύρος τιμών. Το λειτουργικό σύστημα Android είναι διαθέσιμο σε μια μεγάλη γκάμα συσκευών από διάφορους κατασκευαστές, προσφέροντας διαφορετικές δυνατότητες, προδιαγραφές και τιμές. Αυτό κάνει το λειτουργικό σύστημα Android πιο προσιτό σε χρήστες με διαφορετικούς προϋπολογισμούς και προτιμήσεις. Από την άλλη πλευρά, το iOS είναι αποκλειστικά διαθέσιμο σε συσκευές Apple, οι οποίες είναι γενικά πιο ακριβές από ορισμένες συσκευές Android, καθιστώντας το λιγότερο προσβάσιμο σε χρήστες με περιορισμούς προϋπολογισμού.

Τόσο το λειτουργικό σύστημα Android όσο και το iOS ενσωματώνονται με τα αντίστοιχα οικοσυστήματα και υπηρεσίες τους. Το λειτουργικό σύστημα Android έχει απρόσκοπτη ενοποίηση με τις υπηρεσίες της Google όπως το Gmail, το Google Drive, οι Χάρτες Google, το Google Pay και το Google Assistant, καθιστώντας το μια προτιμώμενη επιλογή για χρήστες που βασίζονται σε μεγάλο βαθμό στο οικοσύστημα της Google. Το iOS, από την άλλη πλευρά, ενσωματώνεται στενά με το οικοσύστημα της Apple, συμπεριλαμβανομένων των iCloud, Apple Music, Apple Pay και τη Siri, παρέχοντας μια απρόσκοπτη εμπειρία σε όλες τις συσκευές Apple.

Η απόδοση και η βελτιστοποίηση είναι κρίσιμοι παράγοντες στην εμπειρία χρήστη των κινητών συσκευών. Το λειτουργικό σύστημα Android εκτελείται σε ένα ευρύ φάσμα

συσκευών με ποικίλες δυνατότητες υλικού, με αποτέλεσμα ασυνέπεια στην απόδοση. Ωστόσο, οι πρόσφατες βελτιστοποιήσεις σε νεότερες εκδόσεις Android, όπως το Project Treble και το Project Mainline, έχουν ως στόχο τη βελτίωση της απόδοσης. Από την άλλη πλευρά, το iOS είναι βελτιστοποιημένο για να λειτουργεί με το δικό του υλικό της Apple, παρέχοντας ομαλή και σταθερή απόδοση σε όλες τις συσκευές Apple.

Τόσο το λειτουργικό σύστημα Android όσο και το iOS παρέχουν στους προγραμματιστές ισχυρά εργαλεία και οικοσυστήματα για τη δημιουργία εφαρμογών για κινητά. Το λειτουργικό σύστημα Android χρησιμοποιεί Java, Kotlin και C++ ως τις κύριες γλώσσες προγραμματισμού του και προσφέρει ένα ευρύ φάσμα εργαλείων ανάπτυξης και βιβλιοθηκών μέσω του Android Studio IDE. Το iOS, από την άλλη πλευρά, χρησιμοποιεί τη Swift και την Objective-C ως τις κύριες γλώσσες προγραμματισμού του και παρέχει εργαλεία ανάπτυξης όπως το Xcode IDE για την ανάπτυξη εφαρμογών.

Όσον αφορά το μερίδιο αγοράς και τους χρήστες, το λειτουργικό σύστημα Android κυριαρχεί στην παγκόσμια αγορά smartphones με υψηλότερο μερίδιο αγοράς σε σύγκριση με το iOS, σε αυτό οφείλετε η ευρύτερη διαθεσιμότητα του λειτουργικού συστήματος Android σε διάφορες συσκευές από διαφορετικούς κατασκευαστές και οι προσιτές τιμές. Ωστόσο, το iOS έχει μια πιστή και εύπορη βάση χρηστών, με την ισχυρή εικόνα της επωνυμίας και το οικοσύστημα της Apple να προσελκύει ένα premium τμήμα πελατών.

Τόσο το λειτουργικό σύστημα Android όσο και το iOS εξελίσσονται συνεχώς για να προσαρμόζονται στις μεταβαλλόμενες τεχνολογικές τάσεις και καινοτομίες. Και τα δύο λειτουργικά συστήματα ενσωματώνουν αναδυόμενες τεχνολογίες όπως η επαυξημένη πραγματικότητα (AR), η εικονική πραγματικότητα (VR), η μηχανική μάθηση και η τεχνητή νοημοσύνη (AI) στα οικοσυστήματά τους, παρέχοντας νέες ευκαιρίες για προγραμματιστές εφαρμογών και χρήστες. Καθώς η τεχνολογία προχωρά, το μέλλον του λειτουργικού συστήματος Android και του iOS πιθανότατα θα διαμορφωθεί από νέες τάσεις και καινοτομίες, οδηγώντας σε περαιτέρω εξελίξεις στον τομέα των κινητών συσκευών.<sup>24,25,26,27,28,29,30</sup>

## **1.8 Ασφάλεια του Android**

Το λειτουργικό σύστημα Android περιλαμβάνει πολλές λειτουργίες ασφαλείας για την προστασία των δεδομένων χρήστη και την αποτροπή μη εξουσιοδοτημένης πρόσβασης. Ένα τέτοιο χαρακτηριστικό είναι το σύστημα αδειών εφαρμογής, το οποίο επιτρέπει στους

χρήστες να ελέγχουν σε ποιες άδειες έχει πρόσβαση μια εφαρμογή. Για παράδειγμα, εάν μια εφαρμογή ζητήσει πρόσβαση στην κάμερα ή το μικρόφωνο της συσκευής, ο χρήστης μπορεί να επιλέξει να επιτρέψει ή να αρνηθεί αυτήν την πρόσβαση. Το λειτουργικό σύστημα Android περιλαμβάνει επίσης μια ασφαλή διαδικασία εκκίνησης, η οποία διασφαλίζει ότι μόνο αξιόπιστο λογισμικό φορτώνεται κατά την εκκίνηση. Επιπλέον, το λειτουργικό σύστημα Android υποστηρίζει πλήρως κρυπτογράφηση δίσκου, η οποία κρυπτογραφεί όλα τα δεδομένα που είναι αποθηκευμένα στη συσκευή για να την προστατεύσει από μη εξουσιοδοτημένη πρόσβαση σε περίπτωση απώλειας ή κλοπής της συσκευής.<sup>31,32,33,34</sup>

## **1.9 Το Πρόβλημα που επιλύει η εφαρμογή**

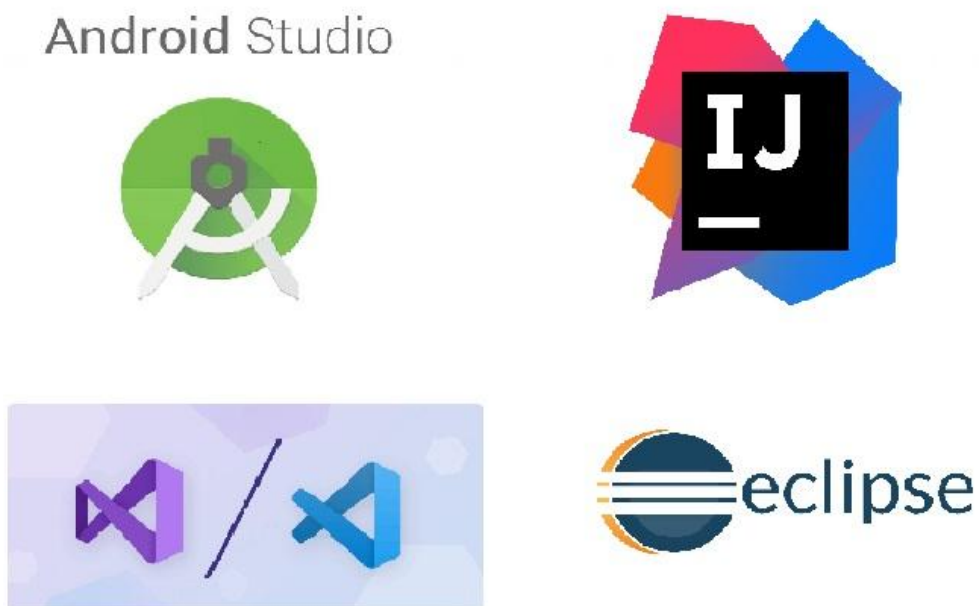
Η εφαρμογή «Επίλυσης του προβλήματος του Μέλανος Σώματος (Black Body)» έχει γίνει με σκοπό την προσομοίωσης και την απεικόνιση της καμπύλης της ακτινοβολίας του Μέλανος Σώματος, την εξάρτηση της από τη θερμοκρασία και τη μετατόπιση της κορυφής του σχήματος με τη θερμοκρασία, τα οποία περιγράφονται αντίστοιχα από τρεις νόμους, το νόμο του Planck, το νόμος του Stefan-Boltzmann και ο νόμος του Wien. Παρουσιάζει ακόμα κάποιες ιδιότητες ενός αστεριού όπως η θερμοκρασία, η φωτεινότητα, η ισχύς εκπομπής, η σχέση θερμοκρασίας-χρώματος και το φασματικό τύπο στο πλαίσιο ενός μοντέλου ακτινοβολίας μέλανος σώματος

Η εφαρμογή μπορεί να χρησιμοποιεί με σκοπό τον έλεγχο της εγκυρότητας του νόμου του Planck σχεδιάζοντας την καμπύλη του μέλανος σώματος ενός αστεριού σε μια συγκεκριμένη θερμοκρασία, καθώς και την επαλήθευση του νόμου του Wien δοκιμάζοντας τη σχέση μεταξύ της μετρούμενης θερμοκρασίας του Αστέρα και του προκύπτοντος χρώματος του, ακόμα μπορεί να βοηθήσει στην επαλήθευση του φασματικού τύπου ενός αστεριού και στον προσδιορισμό της συνολικής εκπεμπόμενης έντασης (ισχύς) και φωτεινότητας (L) ενός αστέρα από την εφαρμογή του νόμου Stefan-Boltzmann.

## **2. Περιβάλλον Ανάπτυξης**

Τα τελευταία χρόνια, η ευρεία υιοθέτηση κινητών συσκευών οδήγησε σε μια έκρηξη στην ανάπτυξη εφαρμογών για κινητές συσκευές, ιδιαίτερα εκείνων που στοχεύουν στην πλατφόρμα Android, ως αποτέλεσμα οι προγραμματιστές απαιτούν ισχυρά και αποτελεσματικά εργαλεία για τον εξορθολογισμό της διαδικασίας ανάπτυξης εφαρμογών. Μια τέτοια κατηγορία εργαλείων που έχει κερδίσει εξέχουσα θέση στον χώρο ανάπτυξης

εφαρμογών για κινητά είναι τα Ενσωματωμένα Περιβάλλοντα Ανάπτυξης Android (IDE). Τα IDE παρέχουν ένα ολοκληρωμένο σύνολο εργαλείων και λειτουργιών για την ανάπτυξη εφαρμογών Android όπου οι προγραμματιστές μπορούν να γράφουν, να μεταγλωττίζουν, να διορθώνουν και να ελέγχουν τον κώδικά τους, όλα μέσα σε μια ενιαία διεπαφή. Τα IDE απλοποιούν τη διαδικασία ανάπτυξης παρέχοντας διάφορες δυνατότητες, όπως συμπλήρωση κώδικα, επισήμανση σύνταξης, εργαλεία εντοπισμού σφαλμάτων και προγράμματα επεξεργασίας οπτικής διάταξης, τα οποία ενισχύουν την παραγωγικότητα και την αποτελεσματικότητα. Επιπλέον, τα IDE διευκολύνουν τη συνεργασία μεταξύ των προγραμματιστών, προσφέροντας έλεγχο εκδόσεων, κοινή χρήση κώδικα και δυνατότητες διαχείρισης έργου.



Εικόνα 2 Android IDEs

## 2.1 Android Studio



*Εικόνα 2 Android IDEs*

Το Android Studio είναι ένα επίσημο IDE που αναπτύχθηκε από την Google για τη δημιουργία εφαρμογών Android. Είναι χτισμένο πάνω στο IntelliJ IDEA της JetBrains, ένα δημοφιλές Java IDE, και έχει σχεδιαστεί ειδικά για την ανάπτυξη εφαρμογών Android. Το Android Studio παρέχει ένα ολοκληρωμένο σύνολο εργαλείων και λειτουργιών που απλοποιούν τη διαδικασία ανάπτυξης, αυξάνουν την παραγωγικότητα και βελτιώνουν την ποιότητα των εφαρμογών Android.

Το Android Studio παρέχει ένα πλήρες περιβάλλον ανάπτυξης με ένα πλούσιο σύνολο εργαλείων για το σχεδιασμό, την κωδικοποίηση, τη δοκιμή και τον εντοπισμό σφαλμάτων εφαρμογών Android. Περιλαμβάνει μια διαισθητική και φιλική προς το χρήστη διεπαφή, που διευκολύνει τους προγραμματιστές να πλοηγηθούν και να εργαστούν στα έργα τους. Το Android Studio παρέχει προηγμένες λειτουργίες επεξεργασίας κώδικα, όπως συμπλήρωση κώδικα, επισήμανση σύνταξης και ανακατασκευή, που βοηθούν τους προγραμματιστές να γράφουν καθαρό και αποτελεσματικό κώδικα και υποστηρίζει γλώσσες προγραμματισμού Java, Kotlin και C++. Το Android Studio περιλαμβάνει επίσης ένα πρόγραμμα επεξεργασίας οπτικής διάταξης που επιτρέπει στους προγραμματιστές να δημιουργούν διατάξεις εφαρμογών χρησιμοποιώντας στοιχεία μεταφοράς και απόθεσης, ακόμα παρέχει μια προεπισκόπηση της διάταξης σε πραγματικό χρόνο και υποστηρίζει διαφορετικά μεγέθη και προσανατολισμούς οθόνης, καθιστώντας εύκολη τη δημιουργία διεπαφών χρήστη. Το Android Studio χρησιμοποιεί το σύστημα δημιουργίας Gradle, το οποίο είναι ένα ευέλικτο και επεκτάσιμο σύστημα κατασκευής για εφαρμογές Android που επιτρέπει στους προγραμματιστές να ορίζουν εξαρτήσεις, να διαχειρίζονται πόρους και να διαμορφώνουν



παραλλαγές κατασκευής, καθιστώντας εύκολη τη δημιουργία και τη διανομή εφαρμογών για διαφορετικές συσκευές και διαμορφώσεις.

Το Android Studio περιλαμβάνει έναν ενσωματωμένο εξομοιωτή που επιτρέπει στους προγραμματιστές να δοκιμάσουν τις εφαρμογές τους σε εικονικές συσκευές με διαφορετικές διαμορφώσεις. Υποστηρίζει διάφορες εκδόσεις Android, μεγέθη οθόνης και διαμορφώσεις υλικού, καθιστώντας εύκολο τον εντοπισμό και την επίλυση προβλημάτων κατά την ανάπτυξη. Παρέχει εκτεταμένη υποστήριξη για τη δοκιμή των εφαρμογών Android, συμπεριλαμβανομένης της δοκιμής μονάδας (unit testing), της δοκιμής ενοποίησης (integration testing) και της δοκιμής διεπαφής χρήστη (UI testing). Περιλαμβάνει ένα πλαίσιο δοκιμών και εργαλεία που βοηθούν τους προγραμματιστές να γράφουν, να εκτελούν και να αναλύουν δοκιμές, διασφαλίζοντας την ποιότητα και την αξιοπιστία των εφαρμογών τους.

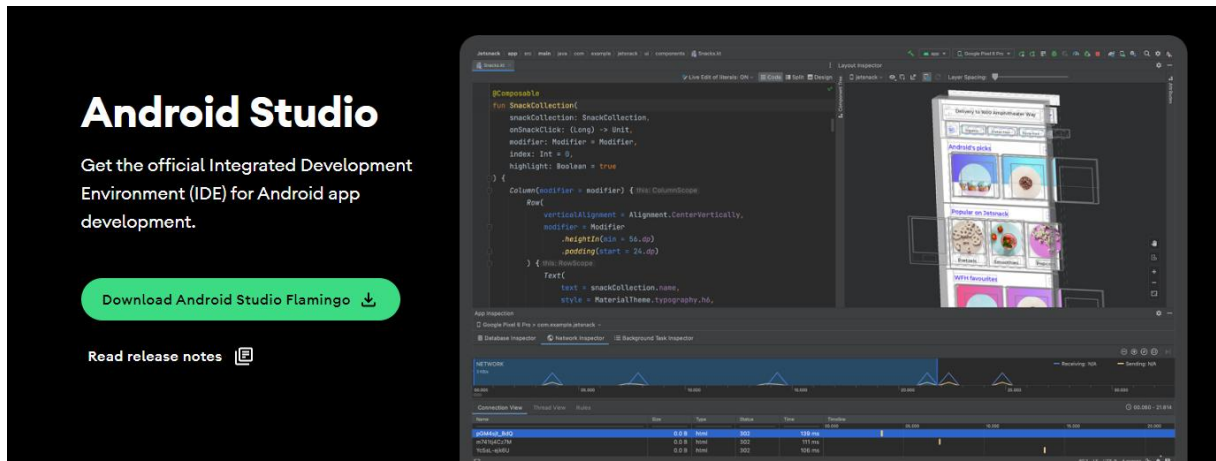
Το Android Studio υποστηρίζει την ανάπτυξη άμεσης εφαρμογής, η οποία επιτρέπει στους προγραμματιστές να δημιουργούν αρθρωτές εφαρμογές που μπορούν να ληφθούν και να εκτελεστούν χωρίς εγκατάσταση. Αυτή η δυνατότητα βοηθά στη μείωση του μεγέθους της εφαρμογής, στη βελτίωση της απόδοσης της εφαρμογής και στη βελτίωση της εμπειρίας χρήστη.

Το Android Studio υποστηρίζει ένα ευρύ φάσμα προσθηκών και επεκτάσεων που ενισχύουν τη λειτουργικότητά του και επιτρέπουν στους προγραμματιστές να προσαρμόσουν το περιβάλλον ανάπτυξής τους. Υπάρχει μια πληθώρα από πρόσθετα όπως για ανάλυση κώδικα, έλεγχο έκδοσης και ενσωμάτωση cloud, καθιστώντας το Android Studio ένα εξαιρετικά επεκτάσιμο IDE.<sup>35,36,37,38</sup>

## 2.2 Εγκατάσταση

Η διαδικασία εγκατάστασης του Android Studio περιλαμβάνει διάφορα βήματα, τα οποία περιγράφονται παρακάτω:

**Λήψη του Android Studio:** Το αρχικό βήμα συνεπάγεται την απόκτηση του λογισμικού Android Studio από τον επίσημο ιστότοπο της Google. Ο ιστότοπος παρέχει έναν σύνδεσμο λήψης για την πιο πρόσφατη έκδοση του Android Studio, η οποία είναι συμβατή με διάφορα λειτουργικά συστήματα όπως Windows, macOS ή Linux.



Εικόνα 2.2.1 Λήψη Android Studio

**Εκτέλεση του προγράμματος εγκατάστασης:** Μόλις ολοκληρωθεί η λήψη, το ληφθέν αρχείο εγκατάστασης πρέπει να εκτελεστεί για να ξεκινήσει η διαδικασία εγκατάστασης. Η διαδικασία εγκατάστασης μπορεί να διαφέρει ανάλογα με το συγκεκριμένο λειτουργικό σύστημα που χρησιμοποιείται.



Εικόνα 2.2.2 Εκτέλεση Προγράμματος Εγκατάσταση Android Studio

**Διαμόρφωση Android Studio:** Μετά την εκκίνηση του προγράμματος εγκατάστασης, θα ζητηθεί από τον χρήστη να επιλέξει τα στοιχεία που θα εγκατασταθούν, όπως το Android SDK, τον εξομοιωτή και άλλα εργαλεία. Συνιστάται η εγκατάσταση όλων των στοιχείων για να διασφαλιστεί η ομαλή λειτουργία του Android Studio. Μετά την επιλογή των στοιχείων προς εγκατάσταση, ο χρήστης θα κληθεί να αποδεχτεί τους όρους και τις προϋποθέσεις χρήσης πριν προχωρήσει στην εγκατάσταση. Ο χρήστης πρέπει να διαβάσει προσεκτικά και να συμφωνήσει με τους όρους και τις προϋποθέσεις που ορίζει η Google, οι οποίοι περιγράφουν τις νομικές συμφωνίες και τις ευθύνες που σχετίζονται με τη χρήση του Android Studio.

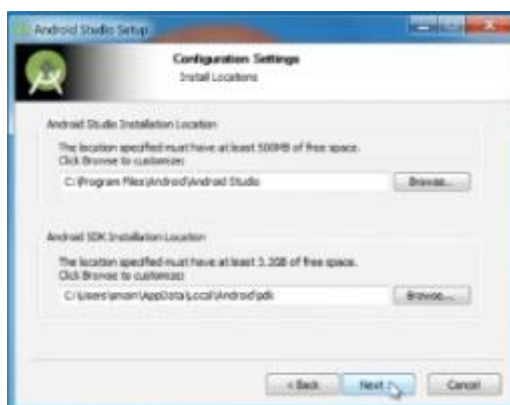


Εικόνα 2.2.3 Διαμόρφωση Android Studio



Εικόνα 2.2.4 Όροι και Προϋποθέσεις Android Studio

**Καθορισμός τοποθεσίας εγκατάστασης:** Στη συνέχεια, θα ζητηθεί από τον χρήστη να καθορίσει την επιθυμητή τοποθεσία εγκατάστασης για το Android Studio. Ο χρήστης μπορεί είτε να επιλέξει την προεπιλεγμένη τοποθεσία είτε να καθορίσει μια προσαρμοσμένη τοποθεσία σύμφωνα με τις προτιμήσεις του.

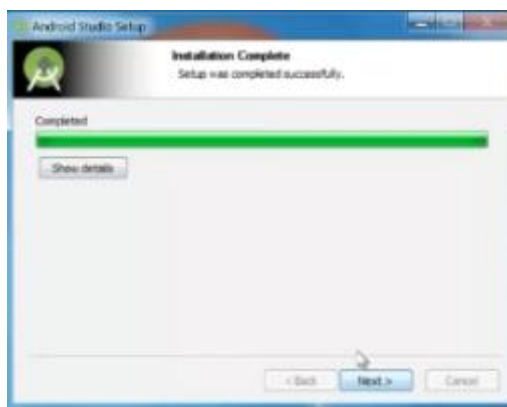


Εικόνα 2.2.5 Τοποθεσία Εγκατάστασης Android Studio

**Εγκατάσταση Android Studio:** Αφού επιλεγούν τα στοιχεία και η τοποθεσία εγκατάστασης, η διαδικασία εγκατάστασης μπορεί να ξεκινήσει κάνοντας κλικ στο κουμπί «Εγκατάσταση». Αυτό θα εγκαταστήσει το Android Studio στον υπολογιστή.



Εικόνα 2.2.6 Start Menu Android Studio

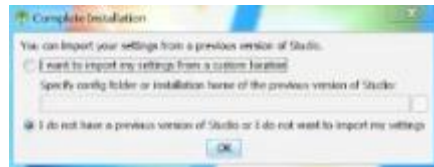


Εικόνα 2.2.7 Εγκατάσταση Android Studio

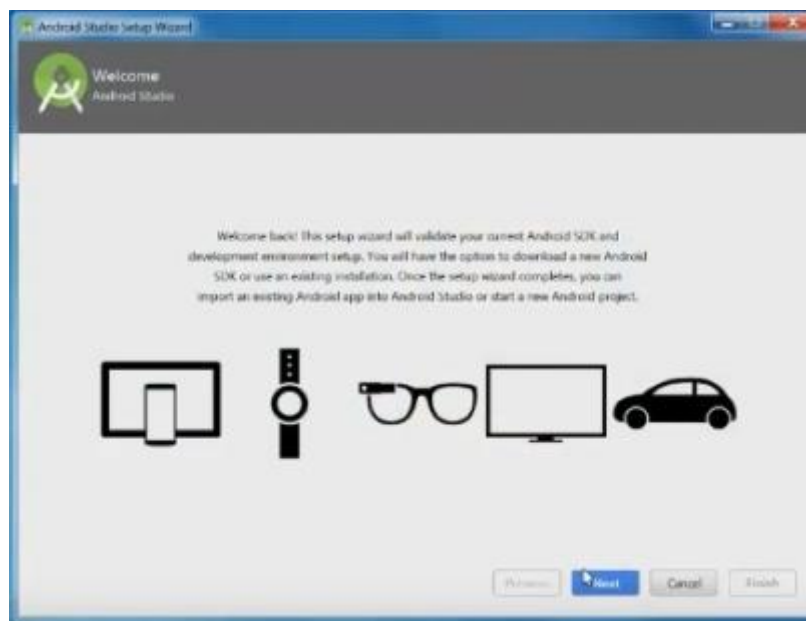


Εικόνα 2.2.8 Ολοκλήρωση Εγκατάσταση Android Studio

**Ολοκλήρωση της εγκατάστασης:** Μετά την ολοκλήρωση της εγκατάστασης, ο χρήστης θα κληθεί να επιλέξει εάν θα εισαγάγει προηγούμενες ρυθμίσεις ή όχι. Ο χρήστης μπορεί να επιλέξει την επιθυμητή επιλογή και να κάνει κλικ στο κουμπί «ΟΚ» για να ολοκληρώσει τη διαδικασία εγκατάστασης.

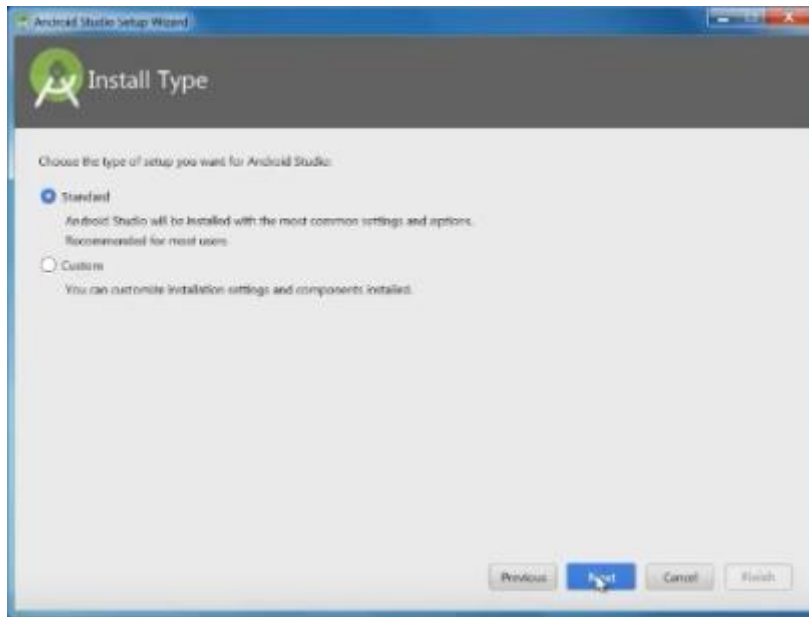


Εικόνα 2.2.9 Προηγούμενες Ρυθμίσεις Android Studio

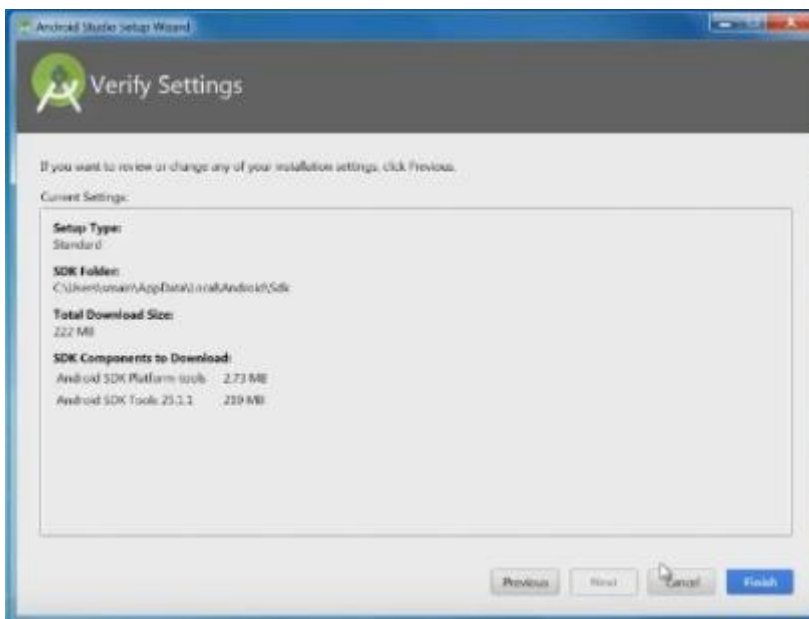


Εικόνα 2.2.10 Οθόνη Καλωσορίσματος Android Studio

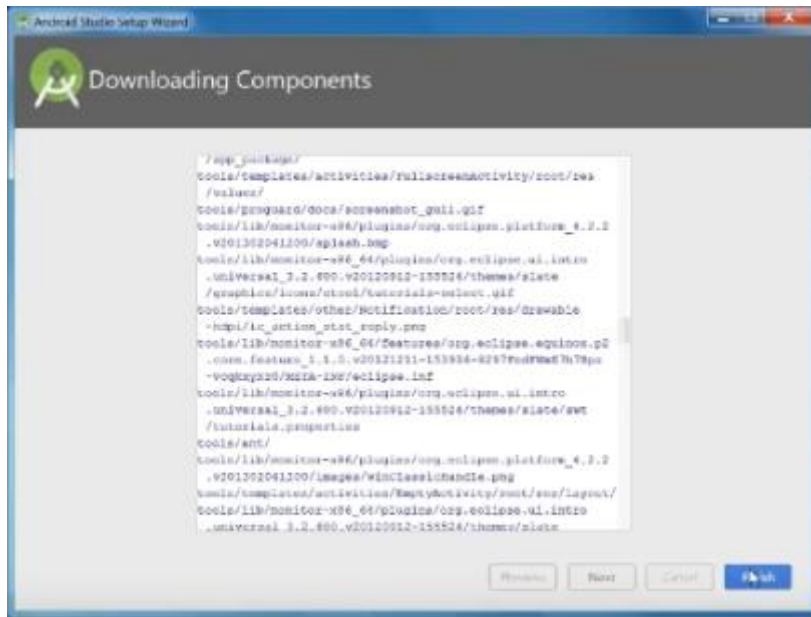
**Διαμόρφωση Android SDK:** Κατά την εκκίνηση του Android Studio, πρέπει να διαμορφωθεί το Android SDK. Το SDK αποτελείται από διάφορα εργαλεία, βιβλιοθήκες και πόρους που απαιτούνται για την ανάπτυξη εφαρμογών Android.



Εικόνα 2.2.11 Επιλογή Τύπος Εγκατάσταση Android Studio



Εικόνα 2.2.12 Επιβεβαίωση Ρυθμίσεων Android Studio SDK



Εικόνα 2.2.13 Λήψη Android Studio SDK

**Εκκίνηση Android Studio:** Με την ολοκλήρωση, το Android Studio μπορεί να εκκινηθεί κάνοντας κλικ στο κουμπί «Τέλος». Αυτό θα ανοίξει το Android Studio και ο χρήστης μπορεί να ξεκινήσει τη δημιουργία εφαρμογών Android.<sup>39</sup>



Εικόνα 2.2.14 Δημιουργία Έργου Android Studio

## 2.3 Δημιουργία νέου Project

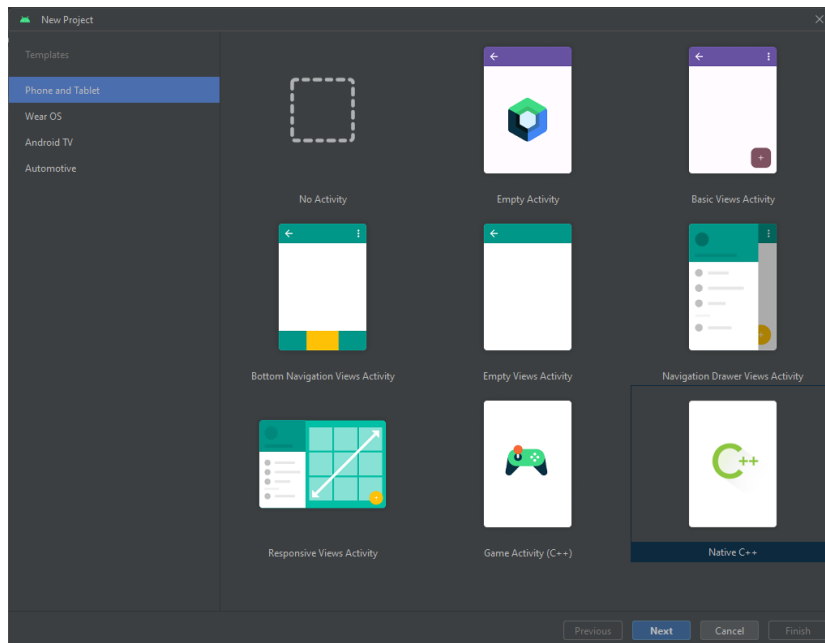
Το Android Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που παρέχει ένα ολοκληρωμένο σύνολο εργαλείων για τη δημιουργία εφαρμογών Android. Η δημιουργία

ενός νέου έργου στο Android Studio είναι ένα κρίσιμο βήμα για κάθε προγραμματιστή που θέλει να δημιουργήσει μια εφαρμογή Android.

Αρχικά, το Android Studio θα πρέπει να εκκινήσει στον υπολογιστή. Εάν δεν υπάρχει εγκαταστημένο, θα πρέπει να γίνει λήψη και εγκατάσταση από τον επίσημο ιστότοπο του Android Studio. Μόλις ανοίξει το Android Studio, εμφανίζεται η οθόνη καλωσορίσματος. Από εκεί η επιλογή "Δημιουργία νέου έργου", θα ανοίξει έναν νέο οδηγό έργου, σε περίπτωση που το Android Studio ήταν ήδη ανοιχτό σε κάποιο άλλο έργο η δημιουργία νέου έργου μπορεί να επιτευχθεί μεταβαίνοντας στο "Αρχείο" > "Νέο" > "Νέο έργο" στη γραμμή μενού.

Το Android Studio παρέχει ένα σύνολο προτύπων από τα οποία μπορούν να επιλεγθούν κατά τη δημιουργία ενός νέου έργου. Αυτά τα πρότυπα παρέχουν ένα σημείο εκκίνησης για την ανάπτυξη της εφαρμογής. Η Επιλογή το πρότυπο γίνεται αναλόγως τις απαιτήσεις της εφαρμογής. Τα πρότυπα αυτά κατανέμονται σε τέσσερες βασικές κατηγορίες ανάλλαγα με τον τύπο συσκευής προορισμού της εφαρμογής.

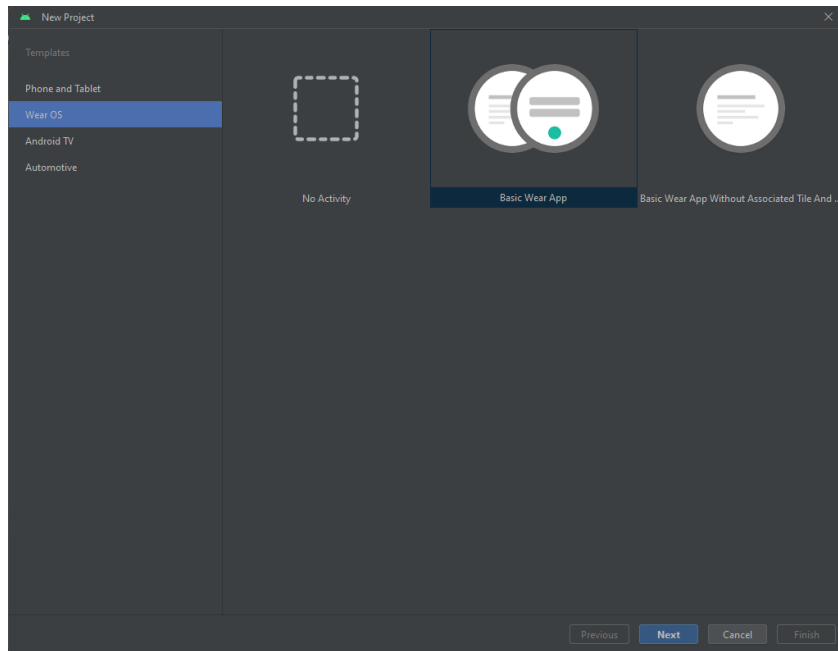
### **Phone and Tablet (Τηλέφωνο και Ταμπλέτα)**



Εικόνα 2.3.1 Πρότυπα Έργου Phone and Tablet

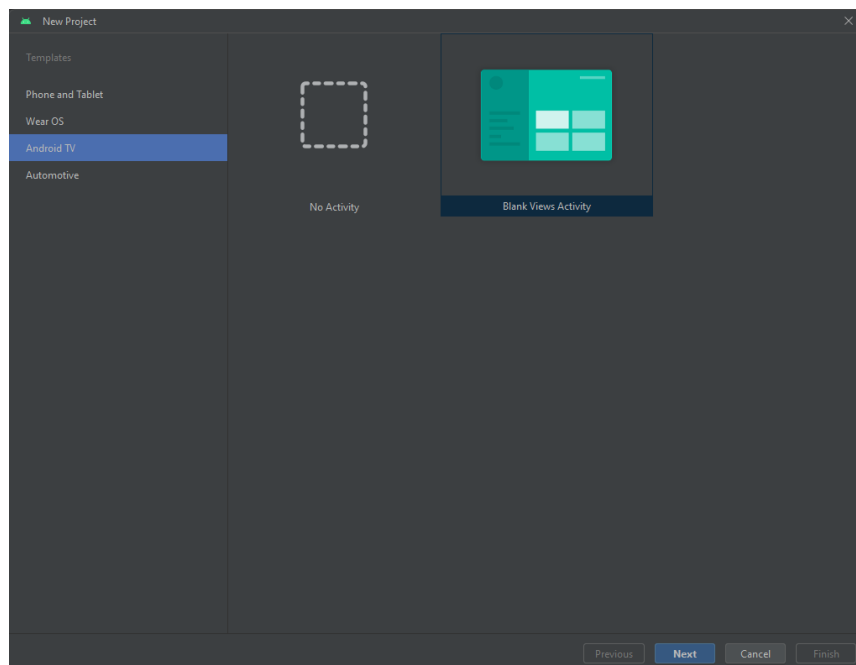
### **Wear OS**





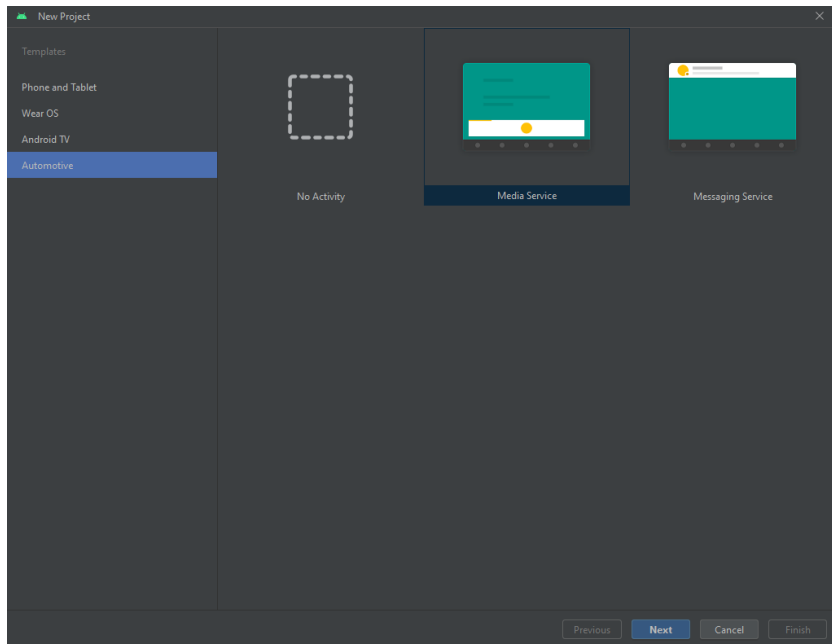
*Εικόνα 2.3.2 Πρότυπα Έργου Wear OS*

## **Android TV**



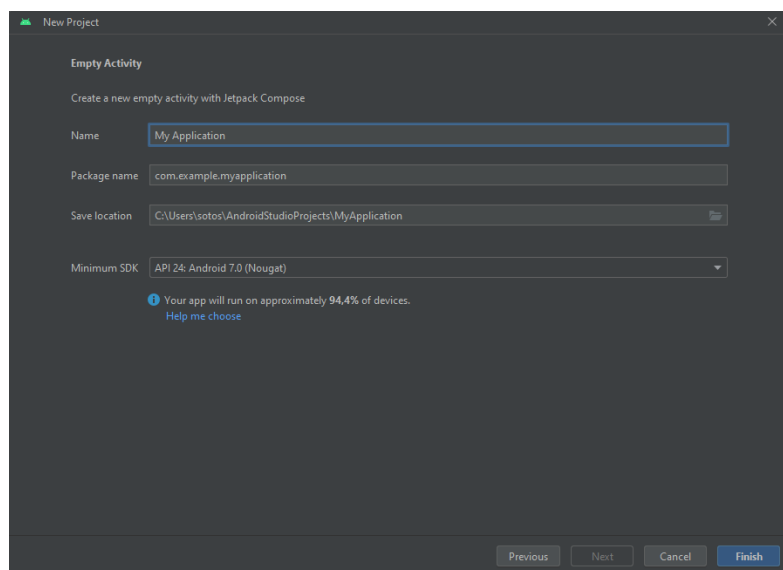
*Εικόνα 2.3.3 Πρότυπα Έργου Android TV*

## **Automotive**



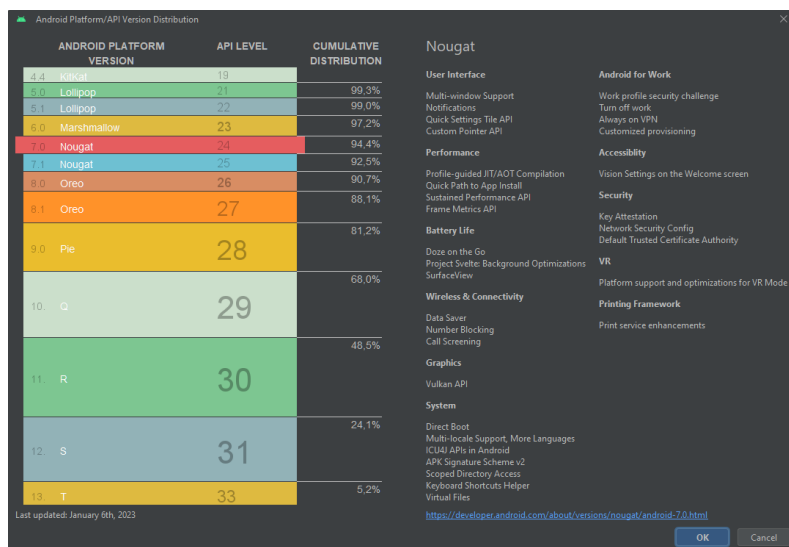
Εικόνα 2.3.4 Πρότυπα Έργου Automotive

Αφού επιλεγθεί το πρότυπο του έργου, θα χρειαστεί να διαμορφωθούν κάποιες λεπτομέρειες σχετικά με το έργο. Θα χρειαστεί να εισαχθεί ένα όνομα για το έργο, να επιλέγει μια γλώσσα προγραμματισμού Java ή Kotlin, να επιλεγεί μια τοποθεσία αποθήκευσης του έργου και να ορισθεί η ελάχιστη έκδοση SDK για την εφαρμογή για τον καθορισμό της χαμηλότερης έκδοσης Android στην οποία μπορεί να εκτελεστεί η εφαρμογή.



Εικόνα 2.3.5 Πρότυπα Έργου Automotive

Συνιστάται να επιλεγεί ένα ελάχιστο SDK που χρησιμοποιείται ευρέως για τη διασφάλιση της μέγιστης συμβατότητας με ένα ευρύ φάσμα συσκευών.



Εικόνα 2.3.6 Διανομή πλατφόρμας Android

Αφού διαμορφωθούν οι λεπτομέρειες του έργου, κάνοντας κλικ στο κουμπί "Τέλος" το Android Studio θα δημιουργήσει τα απαραίτητα αρχεία και φακέλους για το έργο. Μόλις κατασκευαστεί το έργο, μπορείτε να γίνει η εκτέλεση του σε έναν εξομοιωτή ή σε μια φυσική συσκευή κάνοντας κλικ στο κουμπί "Εκτέλεση" στη γραμμή εργαλείων.<sup>40</sup>

## 2.4 Πλοήγηση στο Project

Η ανάπτυξη εφαρμογών για κινητά έχει γίνει μια ολοένα και πιο σημαντική πτυχή της ανάπτυξης λογισμικού λόγω της ευρείας χρήσης των έξυπνων κινητών συσκευών. Το Android Studio είναι το πιο δημοφιλές περιβάλλον ολοκληρωμένης ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android και είναι εξοπλισμένο με πολλές δυνατότητες για τη βελτίωση της διαδικασίας ανάπτυξης. Ένα από αυτά τα χαρακτηριστικά είναι η ικανότητα αποτελεσματικής πλοήγησης στα έργα, εξοικονομώντας πολύτιμο χρόνο και αυξάνοντας την παραγωγικότητα.

Η γραμμή πλοήγησης είναι ένα χρήσιμο εργαλείο που επιτρέπει γρήγορη και εύκολη πλοήγηση στα αρχεία και τους καταλόγους ενός έργου. Μπορεί να βρεθεί στο επάνω μέρος του παραθύρου του Android Studio και εμφανίζει τη θέση του τρέχοντος αρχείου

στη δομή καταλόγου του έργου. Κάνοντας κλικ σε οποιοδήποτε όνομα καταλόγου στη γραμμή πλοήγησης θα ανοίξει τα περιεχόμενα αυτού του καταλόγου στην Προβολή έργου.

Η Προβολή έργου είναι ένα άλλο απαραίτητο εργαλείο για την πλοήγηση σε έργα Android Studio. Εμφανίζει ολόκληρη τη δομή καταλόγου του έργου με ιεραρχικό τρόπο, επιτρέποντας στους προγραμματιστές να περιηγηθούν γρήγορα στα αρχεία και τους φακέλους του έργου. Μπορείτε να αποκτήσετε πρόσβαση στην Προβολή έργου κάνοντας κλικ στην καρτέλα Έργο στην αριστερή πλευρά του παραθύρου του Android Studio.

Η δυνατότητα αναζήτησης από οποιοδήποτε σημείο (Search Everywhere) είναι ένα ισχυρό εργαλείο που επιτρέπει στους προγραμματιστές να αναζητούν αρχεία, κλάσεις, μεθόδους και σύμβολα σε ολόκληρο το έργο. Η πρόσβαση σε αυτήν τη δυνατότητα μπορείς να αποκτηθεί πατώντας δύο φορές το πλήκτρο Shift, το οποίο εμφανίζει μια γραμμή αναζήτησης που πραγματοποιεί αναζήτηση σε ολόκληρο το έργο. Τα αποτελέσματα εμφανίζονται σε ένα αναπτυσσόμενο μενού, επιτρέποντας στους προγραμματιστές να προηγηθούν γρήγορα στο επιθυμητό αρχείο ή σύμβολο.

Η προβολή δομής (Structure View) είναι ένα εργαλείο που εμφανίζει τη δομή του τρέχοντος αρχείου, συμπεριλαμβανομένων των κλάσεων, των μεθόδων και των πεδίων του. Αυτή η δυνατότητα είναι ιδιαίτερα χρήσιμη όταν εργάζεστε με μεγάλα αρχεία που έχουν πολλές μεθόδους και κλάσεις. Η πρόσβαση στην Προβολή δομής επιτυγχάνετε κάνοντας κλικ στην καρτέλα Δομή στην αριστερή πλευρά του παραθύρου του Android Studio.

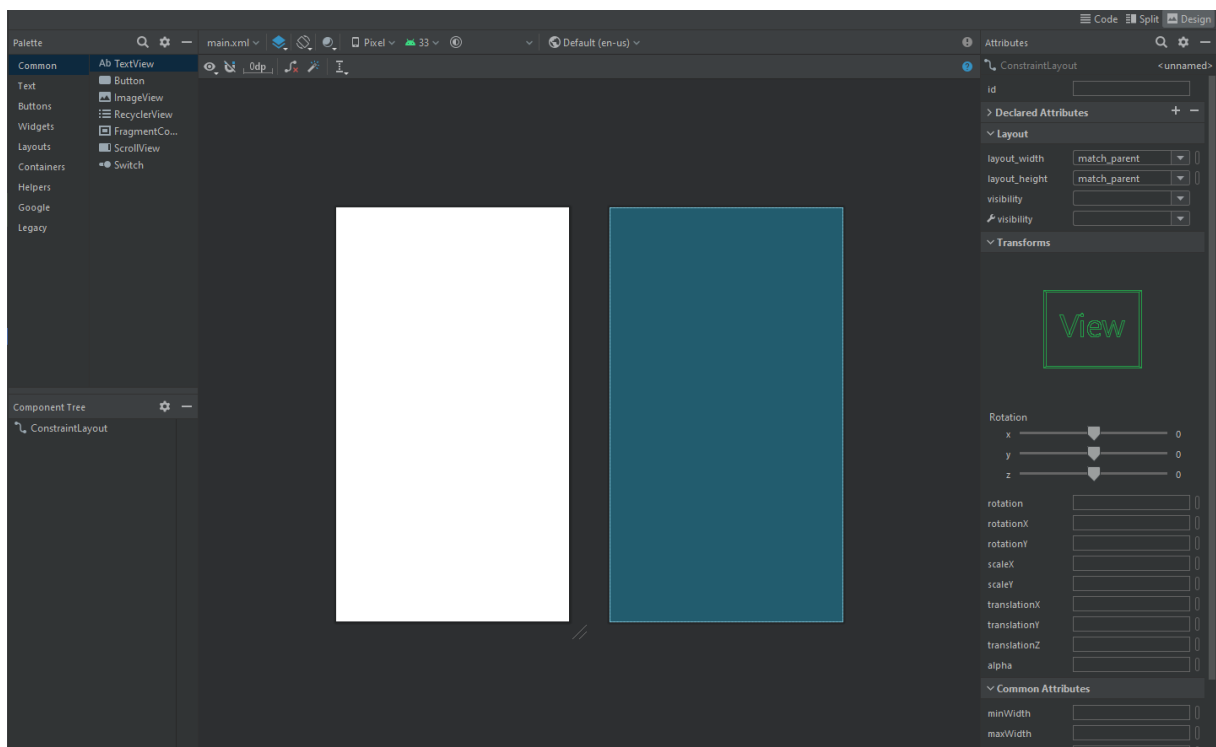
Η λειτουργία Πρόσφατα αρχεία είναι ένας γρήγορος τρόπος πλοήγησης σε αρχεία στα οποία υπήρξε πρόσφατα πρόσβαση. Αυτό μπορεί να επιτευχθεί κάνοντας κλικ στην καρτέλα Πρόσφατα αρχεία στην αριστερή πλευρά του παραθύρου του Android Studio, το οποίο εμφανίζει μια λίστα με αρχεία στα οποία υπήρξε πρόσφατα πρόσβαση.<sup>41</sup>

## **2.5 Δημιουργία User Interface**

Η διεπαφή χρήστη (UI) είναι βασικό στοιχείο οποιασδήποτε εφαρμογής για κινητά. Είναι το πρώτο πράγμα που βλέπουν και αλληλοεπιδρούν οι χρήστες και παίζει καθοριστικό ρόλο στον καθορισμό του πόσο εύκολη είναι η χρήση της εφαρμογής. Ένα καλό UI πρέπει να είναι απλό, διαισθητικό και εύκολο στη χρήση. Θα πρέπει επίσης να είναι συνεπής σε όλη την εφαρμογή, με ομοιόμορφο στυλ και διάταξη. Επιπλέον, θα πρέπει να είναι οπτικά ελκυστικό, με κατάλληλα χρώματα, γραμματοσειρές και εικόνες.

Το Android Studio παρέχει μια μεγάλη γκάμα εργαλείων και στοιχείων για την δημιουργία μια διεπαφής χρήστη. Αυτά περιλαμβάνουν επεξεργαστές διάταξης, γραφικά στοιχεία και στυλ. Η επεξεργασία διάταξης επιτρέπει τη δημιουργία και το χειρισμό οπτικών στοιχείων διεπαφής χρήστη, ενώ τα γραφικά στοιχεία παρέχουν προκατασκευασμένα στοιχεία όπως κουμπιά, πεδία κειμένου και εικόνες, ενώ τα στυλ επιτρέπουν τον ορισμό μιας συνεπής εμφάνισης και αίσθησης για την εφαρμογή.

Για τη δημιουργία μιας διεπαφής χρήστη στο Android Studio, πρέπει πρώτα να δημιουργηθεί ένα νέο έργο και να επιλεγεί το κατάλληλο πρότυπο έργου. Αφού δημιουργηθεί το έργο, μπορεί να ξεκινήσει η σχεδίαση της διεπαφής χρήστη χρησιμοποιώντας το πρόγραμμα επεξεργασίας διάταξης. Το πρόγραμμα επεξεργασίας διάταξης επιτρέπει το σύρσιμο και την απόθεση στοιχείων διεπαφής χρήστη σε έναν καμβά και στη συνέχεια, μπορεί να τροποποιεί το μέγεθος, μετακινώντας ή προσαρμόζοντας τις ιδιότητές τους.



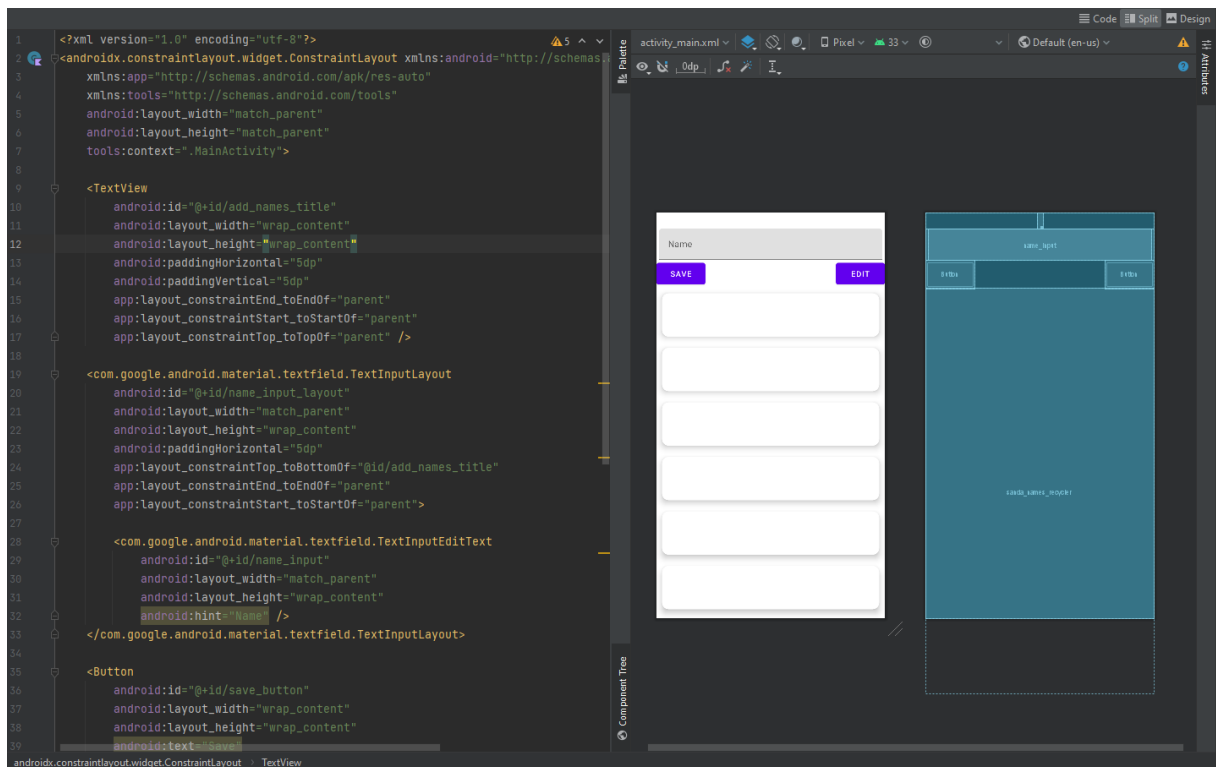
Εικόνα 2.5.1 Πρόγραμμα επεξεργασίας διάταξης Android Studio

Κατά το σχεδιασμό της διεπαφής χρήστη, είναι σημαντικό να λαμβάνεται υπόψη τα διαφορετικά μεγέθη οθόνης και τους προσανατολισμούς στους οποίους θα χρησιμοποιηθεί η εφαρμογή. Το Android Studio παρέχει διαφορετικά αρχεία διάταξης για διαφορετικά

μεγέθη οθόνης και προσανατολισμούς και μπορεί να προσαρμόσει τη διεπαφή χρήστη για καθένα από αυτά.

Αφού δημιουργηθεί η διεπαφή χρήστη, είναι σημαντικό να δοκιμάζετε διεξοδικά για να επιβεβαιωθεί ότι λειτουργεί όπως αναμένεται. Το Android Studio παρέχει μια σειρά εργαλείων για δοκιμή, συμπεριλαμβανομένου του εξομοιωτή Android και της διαχείρισης συσκευών. Η διεπαφή χρήστη θα πρέπει να δοκιμαστεί σε μια σειρά συσκευών για να επιβεβαιωθεί ότι λειτουργεί καλά σε όλες.

Το πρόγραμμα επεξεργασίας διάταξης δημιουργεί αρχεία XML με όλες τις πληροφορίες και τα στοιχεία που εισήχθησαν και τροποποιήθηκαν για την αποθήκευση της διεπαφής χρήστη. Το αρχείο διάταξης XML μπορεί να περιέχει μια ιεραρχία στοιχείων διεπαφής χρήστη και κάθε στοιχείο μπορεί να έχει χαρακτηριστικά που καθορίζουν την εμφάνιση και τη συμπεριφορά του. Μόλις δημιουργηθεί η διάταξη, οι προγραμματιστές μπορούν να τροποποιήσουν τον κώδικα XML απευθείας εάν χρειάζεται. Αυτό μπορεί να είναι χρήσιμο για την τελειοποίηση της διάταξης ή την προσθήκη πιο σύνθετης συμπεριφοράς στα στοιχεία διεπαφής χρήστη. Ενώ η ανάπτυξη διεπαφής χρήστη που βασίζεται σε XML είναι η τυπική προσέγγιση για πολλά χρόνια, μπορεί να είναι πιο περιεκτική και επιρρεπής σε σφάλματα και απαιτεί περισσότερη γνώση του πλαισίου Android, καθώς οι προγραμματιστές πρέπει να κατανοήσουν πώς αναπαράγονται τα στοιχεία διεπαφής χρήστη και πώς αλληλεπιδρούν με την υπόλοιπη εφαρμογή.

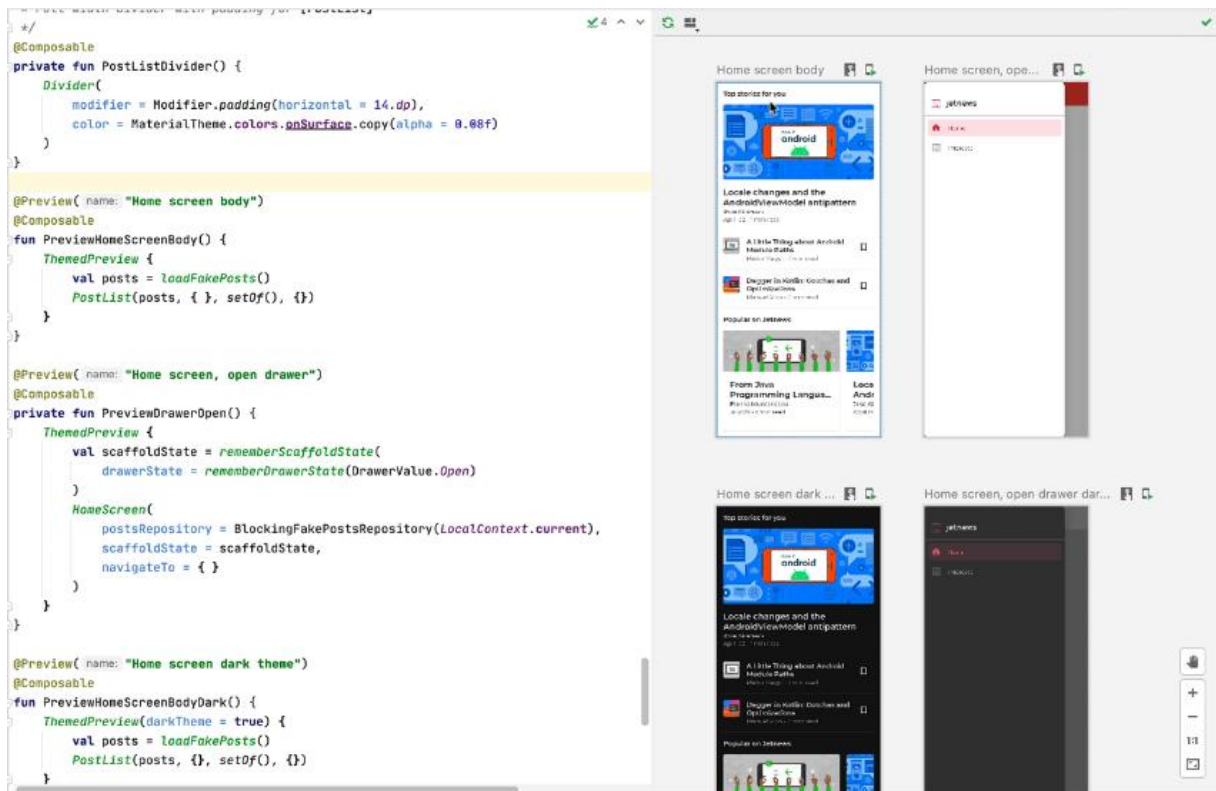


Εικόνα 2.5.2 Πρόγραμμα επεξεργασίας διάταξης XML Αρχείο Android Studio

Η Google το 2020 εισήγαγε το Jetpack Compose που είναι μια σύγχρονη εργαλειοθήκη διεπαφής χρήστη για Android. Το Jetpack Compose επιτρέπει στους προγραμματιστές να δημιουργούν UI χρησιμοποιώντας μια δηλωτική σύνταξη (declarative syntax), όπου η διεπαφή χρήστη περιγράφεται ως μια ιεραρχία συνθετικών συναρτήσεων (Composable functions). Οι συναρτήσεις αυτές επιστρέφουν στοιχεία διεπαφής χρήστη και μπορούν να συνδυαστούν μαζί για να δημιουργήσουν σύνθετες διεπαφές χρήστη. Με το Jetpack Compose, οι προγραμματιστές μπορούν να δημιουργήσουν UI γρηγορότερα και με λιγότερο κώδικα σε σχέση με την παραδοσιακή ανάπτυξη διεπαφής χρήστη που βασίζεται σε XML. Επιτρέπει επίσης τον καλύτερο διαχωρισμό, καθώς η λογική για τη διεπαφή χρήστη και η πραγματική διάταξη της διεπαφής χρήστη μπορούν να διατηρηθούν ξεχωριστά.

Για να δημιουργηθεί μια διεπαφή χρήστη με το Jetpack Compose, πρέπει πρώτα να προστεθούν οι απαραίτητες εξαρτήσεις στο έργο. Στη συνέχεια, μπορεί να ξεκινήσει η δημιουργία της διεπαφής χρήστη χρησιμοποιώντας οι συναρτήσεις του Jetpack Compose. Οι συνθετικές συναρτήσεις μπορούν να προσαρμοστούν με παραμέτρους και μπορούν να συνδυαστούν μαζί χρησιμοποιώντας τροποποιητές (modifiers) για τη δημιουργία σύνθετων διατάξεων.

Ένα ακόμα πλεονέκτημα του Jetpack Compose είναι ότι παρέχει προεπισκόπηση σε πραγματικό χρόνο και επανάληψη φόρτωσης, γεγονός που καθιστά ευκολότερο την παρακολούθηση αλλαγών στη διεπαφή χρήστη κατά την τροποποίηση του κώδικα. Υποστηρίζει επίσης κινούμενα σχέδια και μεταβάσεις, που μπορούν να κάνουν τη διεπαφή χρήστη πιο ελκυστική για τους χρήστες.



Εικόνα 2.5.3 Android Studio προεπισκόπηση Jetpack Compose

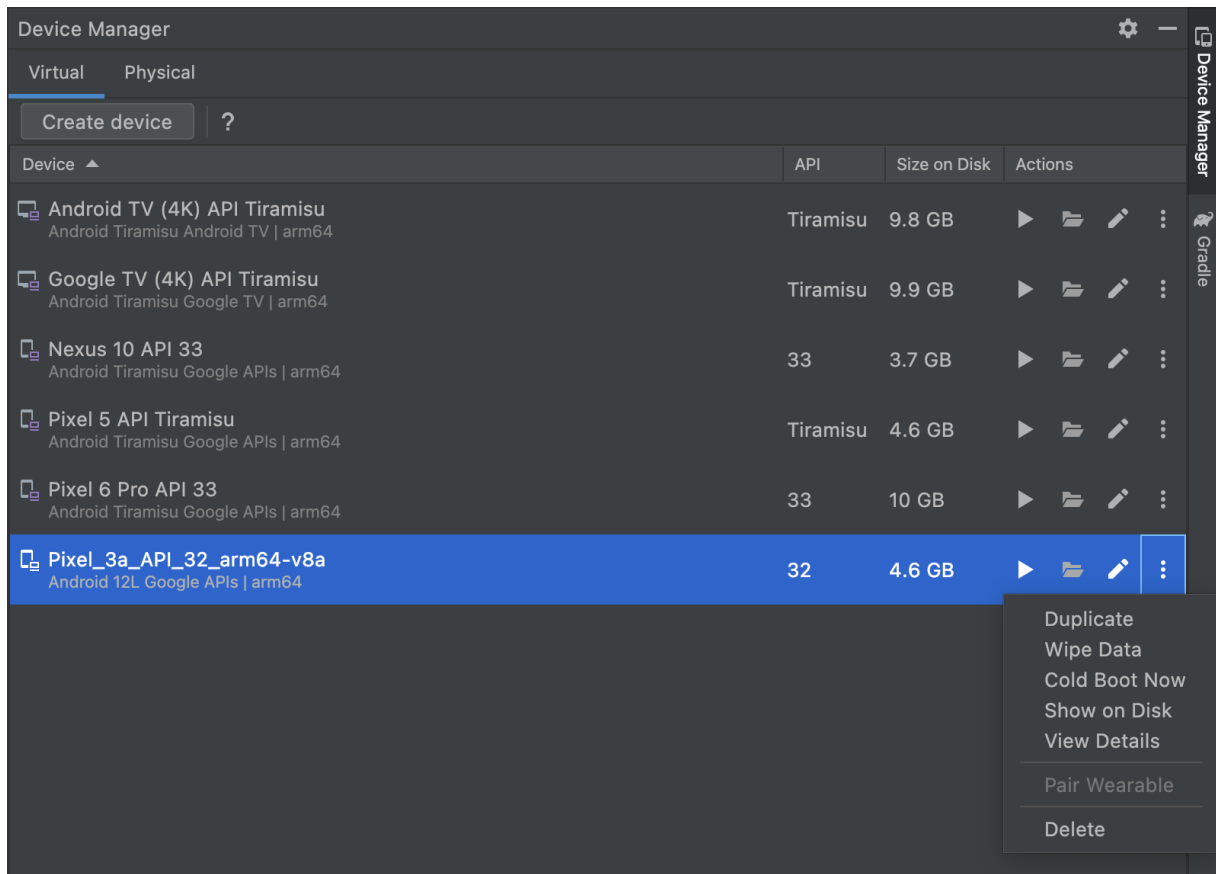
Τόσο το Jetpack Compose όσο και η ανάπτυξη διεπαφής χρήστη που βασίζεται σε XML είναι έγκυρες προσεγγίσεις για τη δημιουργία UI στο Android. Το Jetpack Compose προσφέρει μια πιο σύγχρονη και αποτελεσματική προσέγγιση, ενώ η ανάπτυξη διεπαφής χρήστη που βασίζεται σε XML είναι το πρότυπο για πολλά χρόνια και εξακολουθεί να χρησιμοποιείται ευρέως.

Κατανοώντας τα πλεονεκτήματα και τα μειονεκτήματα κάθε προσέγγισης, οι προγραμματιστές μπορούν να επιλέξουν την καλύτερη προσέγγιση για τις συγκεκριμένες ανάγκες και προτιμήσεις τους καθώς και τον συνδυασμό και τον δυο για τη δημιουργία ενός καλού UI που είναι απαραίτητη για τη δημιουργία μιας επιτυχημένης εφαρμογής.<sup>42,43</sup>



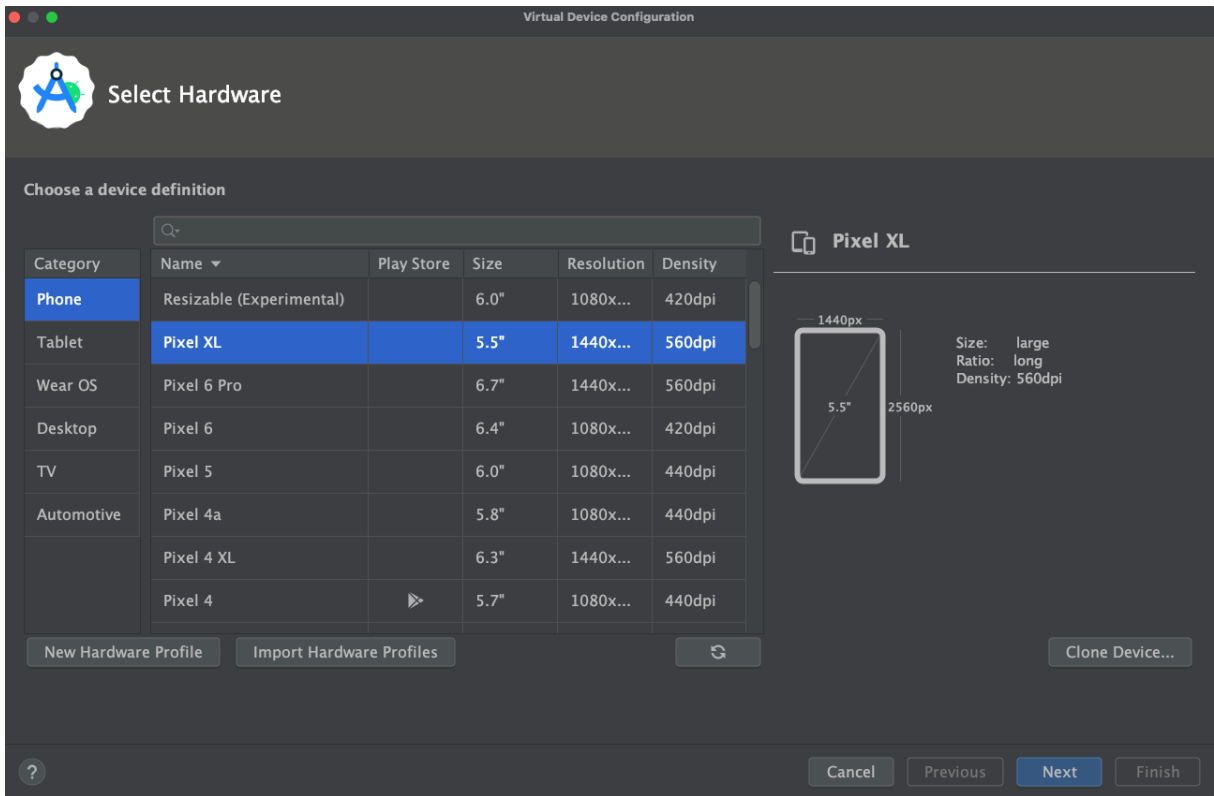
## 2.6 Android Virtual Device (AVD) Manger

Το Android Virtual Device Manager (AVD) είναι ένα εργαλείο που παρέχεται από το Android Studio για τη δημιουργία και τη διαχείριση εικονικών συσκευών για τη δοκιμή εφαρμογών Android. Επιτρέπει στους προγραμματιστές να δοκιμάσουν τις εφαρμογές τους σε ένα ευρύ φάσμα εικονικών συσκευών χωρίς να χρειάζεται να αγοράζουν και να συντηρούν φυσικές συσκευές.

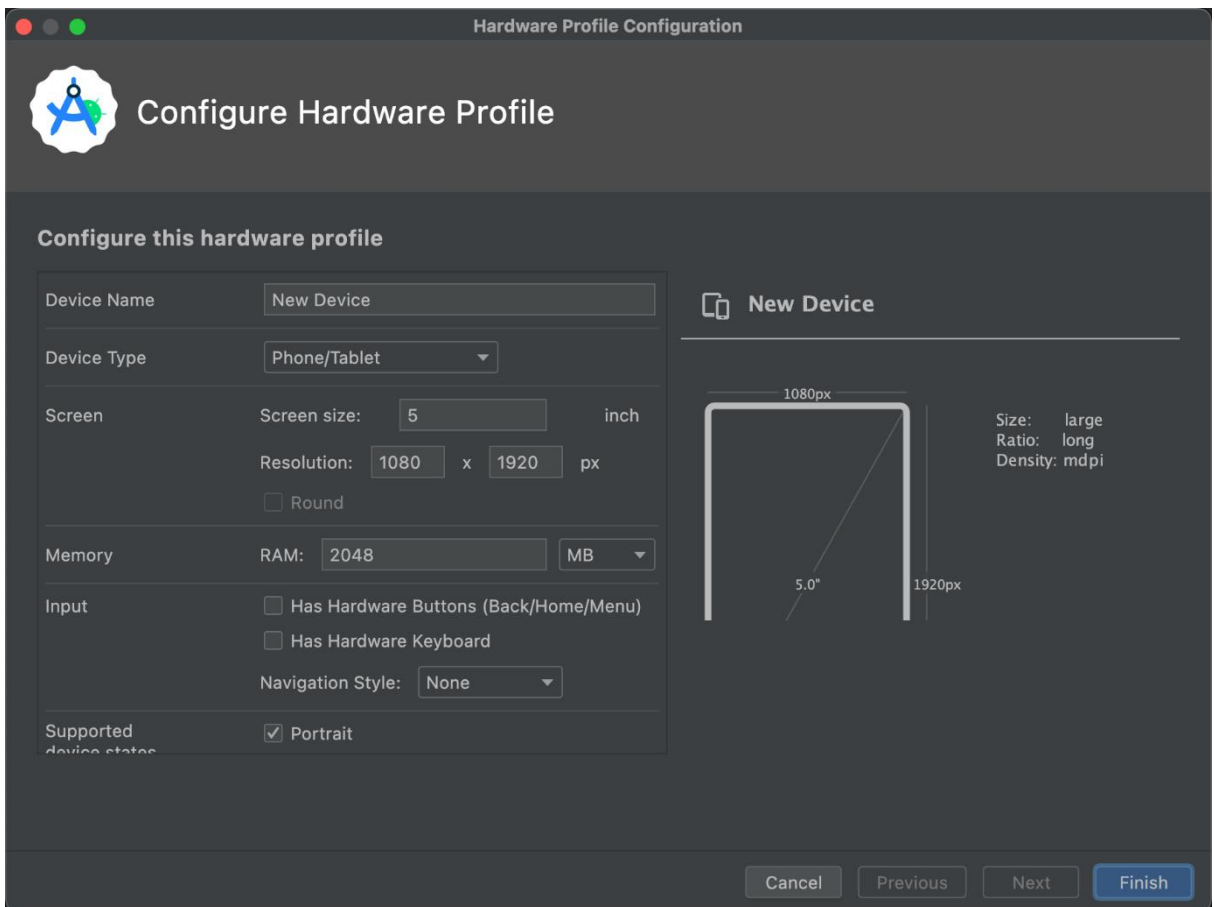


Εικόνα 2.6.1 Διαχειριστής AVD Android Studio

Ο Διαχειριστής AVD επιτρέπει στους προγραμματιστές να διαμορφώνουν εικονικές συσκευές με διαφορετικές προδιαγραφές, όπως μέγεθος οθόνης, ανάλυση και δυνατότητες υλικού. Αυτό επιτρέπει στους προγραμματιστές να δοκιμάσουν τις εφαρμογές τους σε μια σειρά εικονικών συσκευών, προσομοιώνοντας σενάρια πραγματικού κόσμου.

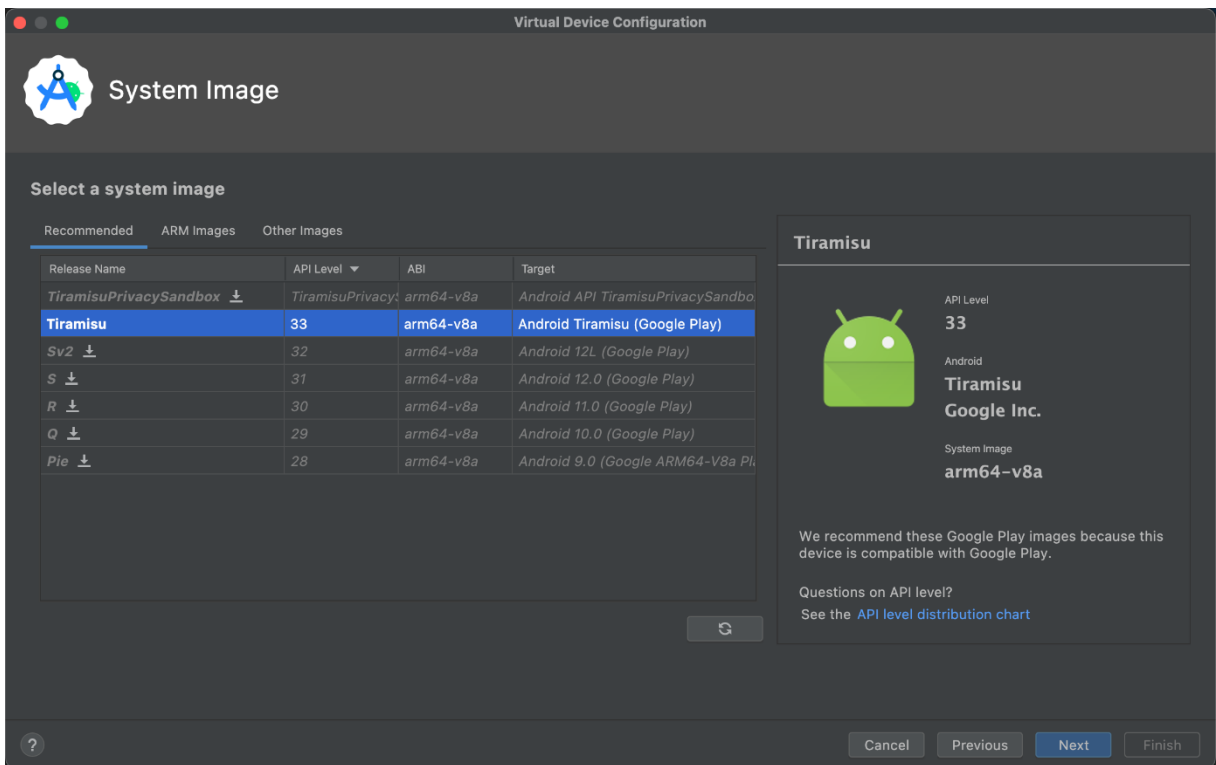


Εικόνα 2.6.2 Google Android Virtual Devices



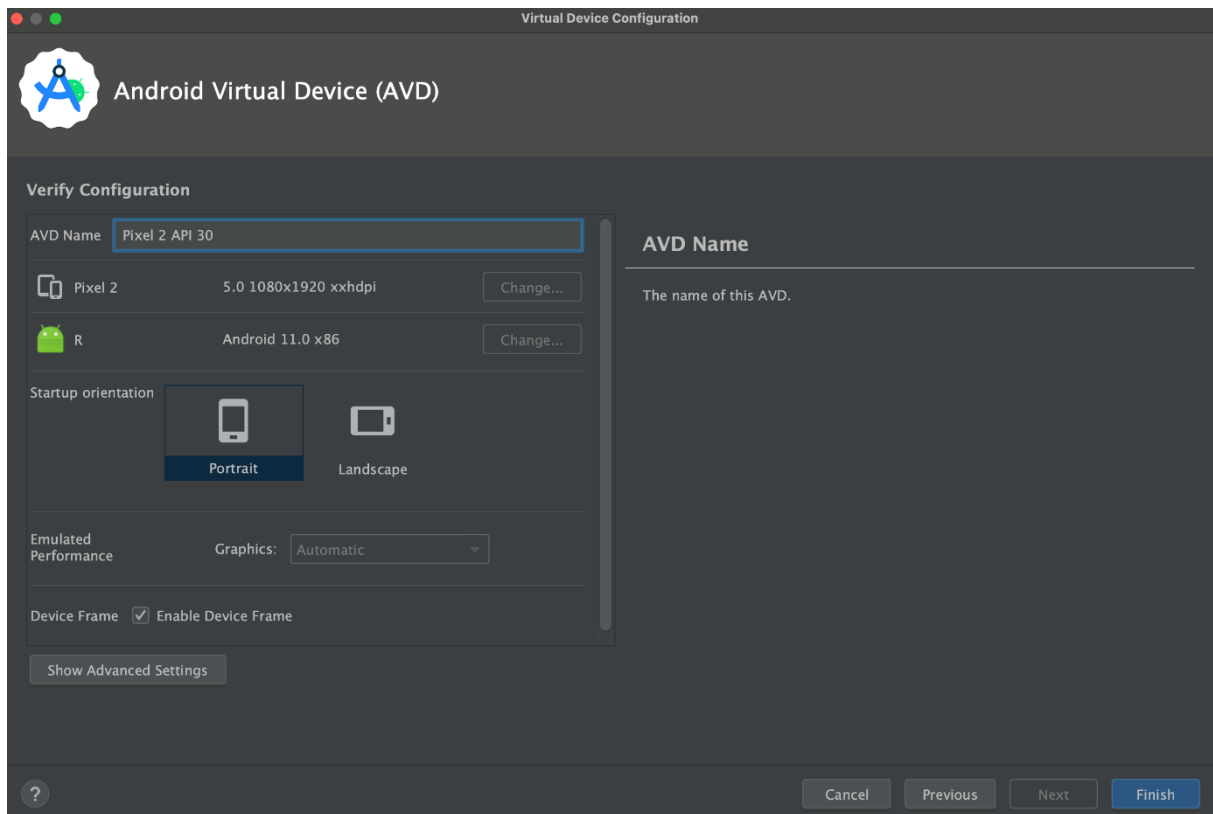
Εικόνα 2.6.3 Διαμόρφωση Προφίλ Υλικού AVD

Ο Διαχειριστής AVD παρέχει μια σειρά εικόνων συστήματος, οι οποίες είναι προρυθμισμένες εκδόσεις του Android που μπορούν να χρησιμοποιηθούν για τη δημιουργία εικονικών συσκευών. Οι προγραμματιστές μπορούν να επιλέξουν από μια σειρά επιπέδων API, τύπων συσκευών και άλλες προδιαγραφές για να δημιουργήσουν εικονικές συσκευές που ταιριάζουν με τις απαιτήσεις δοκιμών τους.



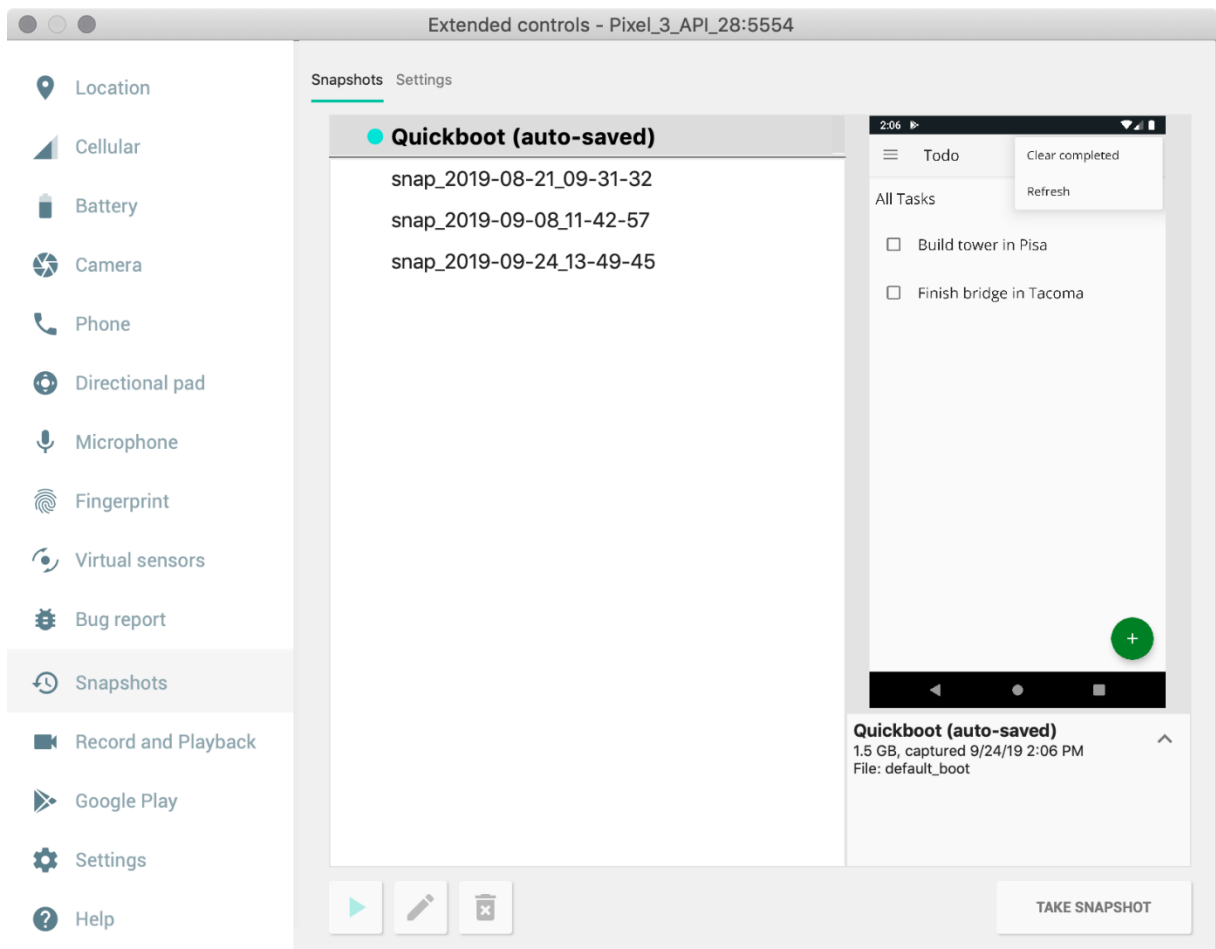
Εικόνα 2.6.4 Εικόνες συστήματος AVD

Ο Διαχειριστής AVD παρέχει εξομοίωση συσκευής, η οποία προσομοιώνει τη συμπεριφορά πραγματικών συσκευών. Αυτό επιτρέπει στους προγραμματιστές να δοκιμάσουν τις εφαρμογές τους σε διαφορετικά σενάρια, όπως χαμηλή μπαταρία, κακή συνδεσιμότητα δικτύου ή διαφορετικούς προσανατολισμούς οθόνης.



Εικόνα 2.6.5 AVD

Ο Διαχειριστής AVD παρέχει υποστήριξη στιγμιότυπων, η οποία επιτρέπει στους προγραμματιστές να αποθηκεύουν και να επαναφέρουν την κατάσταση των εικονικών συσκευών. Αυτό είναι χρήσιμο για δοκιμή και εντοπισμό σφαλμάτων, καθώς οι προγραμματιστές μπορούν να αποθηκεύσουν την κατάσταση μιας εικονικής συσκευής σε μια συγκεκριμένη χρονική στιγμή και στη συνέχεια να την επαναφέρουν σε αυτήν την κατάσταση αργότερα για περαιτέρω δοκιμή.<sup>44</sup>



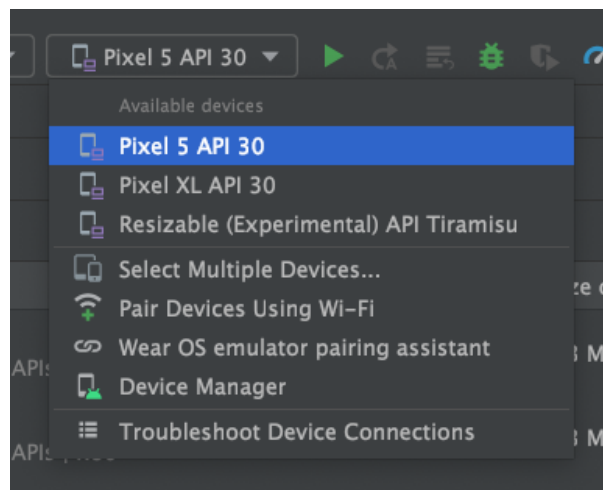
Εικόνα 2.6.6 Στιγμιότυπα AVD

## 2.7 Εκτέλεση και Debugging

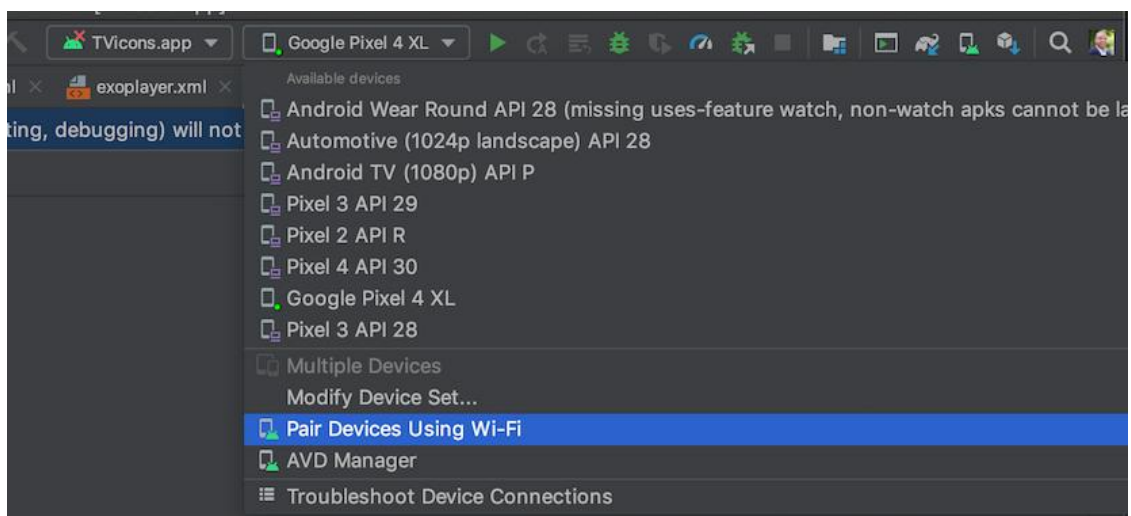
Το Android Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που έχει σχεδιαστεί για τη δημιουργία εφαρμογών Android. Παρέχει διάφορες δυνατότητες που βοηθούν τους προγραμματιστές δημιουργήσουν, να δοκιμάσουν και να αναπτύξουν τις εφαρμογές Android τους.

Για την εκτελεστεί μια εφαρμογή στο Android Studio θα πρέπει να επιλέγει μια συσκευή στην οποία προορίζεται να εκτελεστεί εφαρμογή. Για την εκτελεστεί της εφαρμογής μπορεί χρησιμοποιηθεί είτε έναν εξομοιωτή είτε μια φυσική συσκευή. Οι εξομοιωτές είναι συσκευές που βασίζονται σε λογισμικό που μιμούνται τη συμπεριφορά μιας φυσικής συσκευής. Η δημιουργία ενός εξομοιωτή μπορεί να γίνει μεταβαίνοντας στα Εργαλεία > Διαχείριση AVD > Δημιουργία εικονικής συσκευής. Εναλλακτικά, μπορεί να χρησιμοποιηθεί μια φυσική συσκευή συνδέοντάς την στον υπολογιστή σας χρησιμοποιώντας ένα καλώδιο USB. Η συσκευή θα πρέπει να έχει ενεργοποιημένο το

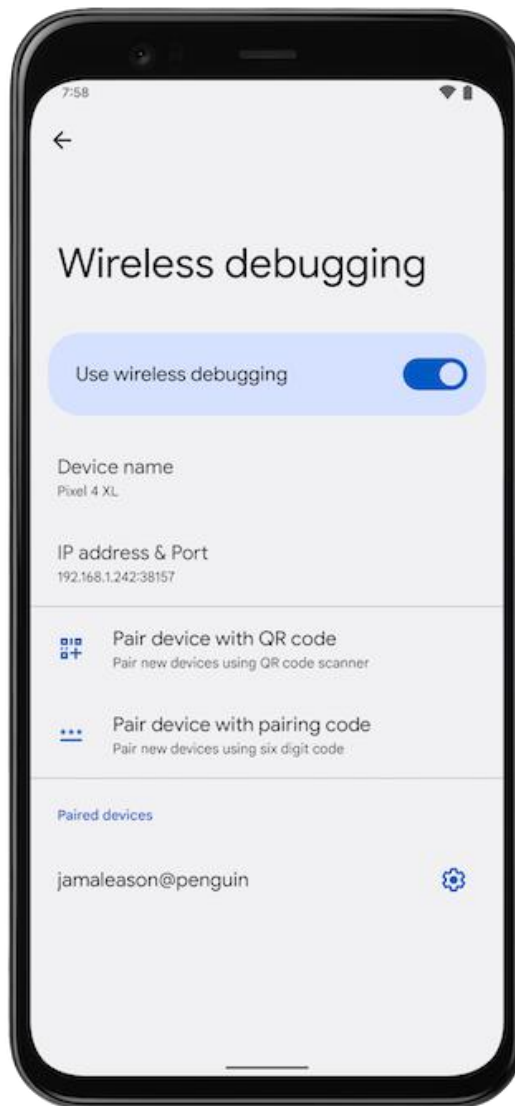
εντοπισμό σφαλμάτων USB. Η ενεργοποίηση του εντοπισμού σφαλμάτων USB σε περίπτωση που δεν είναι ενεργός βρίσκετε στις Ρυθμίσεις > Επιλογές προγραμματιστή > Εντοπισμός σφαλμάτων USB. Από την έκδοση Android 11 και στις νεότερες εκδόσεις του λειτουργικού σύστημα Android, υπάρχει και η επιλογή της ασύρματης εκτέλεσης της εφαρμογής σε μια πραγματική συσκευή μέσω δικτύου Wi-Fi. Για την χρήση της ασύρματης εκτέλεσης θα πρέπει να ενεργοποιεί η επιλογή του ασύρματου εντοπισμού σφαλμάτων που βρίσκετε στις Ρυθμίσεις > Επιλογές προγραμματιστή > Χρήση του Ασύρματου Εντοπισμού Σφαλμάτων. Μετά την ενεργοποίηση της επιλογής η σύνδεση της συσκευής μπορεί να επιτευχθεί με δυο τρόπου σαρώνοντας ένα QR κωδικό ή τοποθετώντας έναν εξαπήφιο μοναδικό κωδικό που εμφανίζεται στο αναδιδόμενο παράθυρο του Android Studio.



Εικόνα 2.7.1 Λίστα Επιλογής Συσκευής για Εκτέλεση Εφαρμογής



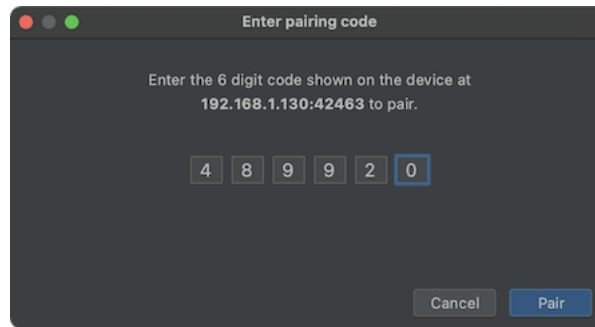
Εικόνα 2.7.2 Ασύρματη Σύνδεση Συσκευής μέσω Wi-Fi



Εικόνα 2.7.3 Μενού Ασύρματου Εντοπισμού Σφαλμάτων



Εικόνα 2.7.4 QR Κωδικός για Ασύρματη Σύνδεση Συσκευής μέσω Wi-Fi



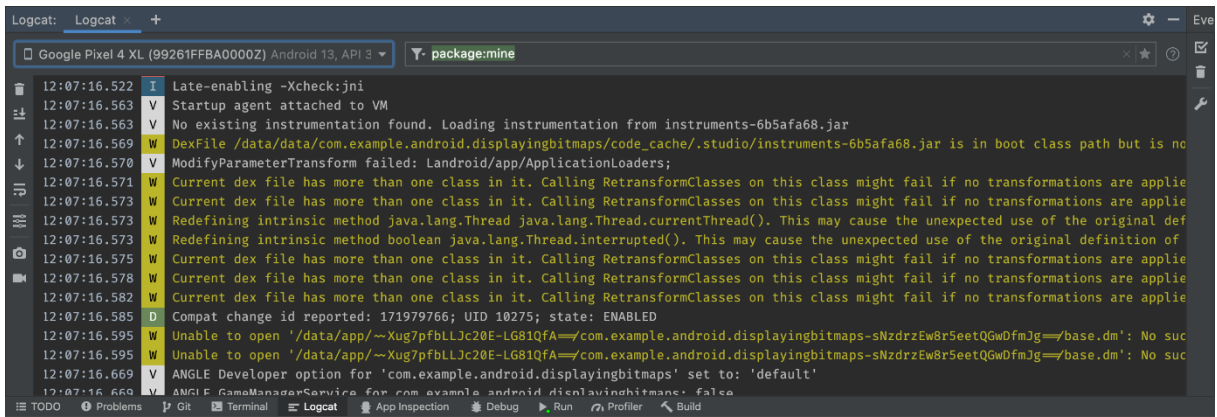
Εικόνα 2.7.5 Εξαψήφιος Κωδικός για Ασύρματη Σύνδεση Συσκευής μέσω Wi-Fi

Αφού εκκινήσει ο εξομοιωτής ή συνδεθεί η συσκευή, μπορεί να γίνει η εκτέλεση της εφαρμογής. Για να εκτελεστεί η εφαρμογή, θα πρέπει να γίνει κλικ στο κουμπί Εκτέλεση στη γραμμή εργαλείων ή επιλέγοντας Εκτέλεση > Εκτέλεση «εφαρμογής» από το μενού. Αυτό θα δημιουργήσει την εφαρμογή και θα την εγκαταστήσει στην επιλεγμένη συσκευή. Στη συνέχεια, μπορεί να δοκιμαστεί η εφαρμογή και να γίνουν οι απαραίτητες αλλαγές.<sup>45,46,47</sup>

Ο εντοπισμός σφαλμάτων είναι ένα ουσιαστικό μέρος της διαδικασίας ανάπτυξης μιας εφαρμογής. Περιλαμβάνει τον εντοπισμό και τη διόρθωση σφαλμάτων που εμφανίζονται κατά την εκτέλεση του προγράμματος. Για να διευκολυνθεί αυτή η διαδικασία, το Android Studio προσφέρει πολλά εργαλεία και τεχνικές που μπορούν να χρησιμοποιηθούν για τον αποτελεσματικό εντοπισμό σφαλμάτων μιας εφαρμογής.

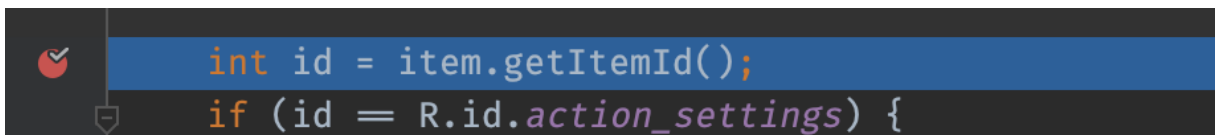
Ένα από τα κύρια εργαλεία για τον εντοπισμό σφαλμάτων μιας εφαρμογής στο Android Studio είναι το Logcat. Αυτό το εργαλείο επιτρέπει την παρακολούθηση αρχείων καταγραφής συστήματος που δημιουργούνται από το λειτουργικό σύστημα Android και την εφαρμογή, βοηθώντας στον εντοπισμό σφαλμάτων και στη παρακολούθηση της ροής της εφαρμογής. Για την απόκτηση πρόσβαση στο Logcat, πρέπει πρώτα η εφαρμογή αν εκκινήσει σε λειτουργία εντοπισμού σφαλμάτων, μετά την οποία μπορεί να γίνει κλικ στην καρτέλα Android Monitor και, στη συνέχεια, να επιλεγεί η καρτέλα Logcat για τη προβολή των αρχείων καταγραφής. Μπορεί να φιλτραριστούν τα αρχεία καταγραφής χρησιμοποιώντας το όνομα πακέτου της εφαρμογής για να γίνει εστίαση στα σχετικά αρχεία καταγραφής.





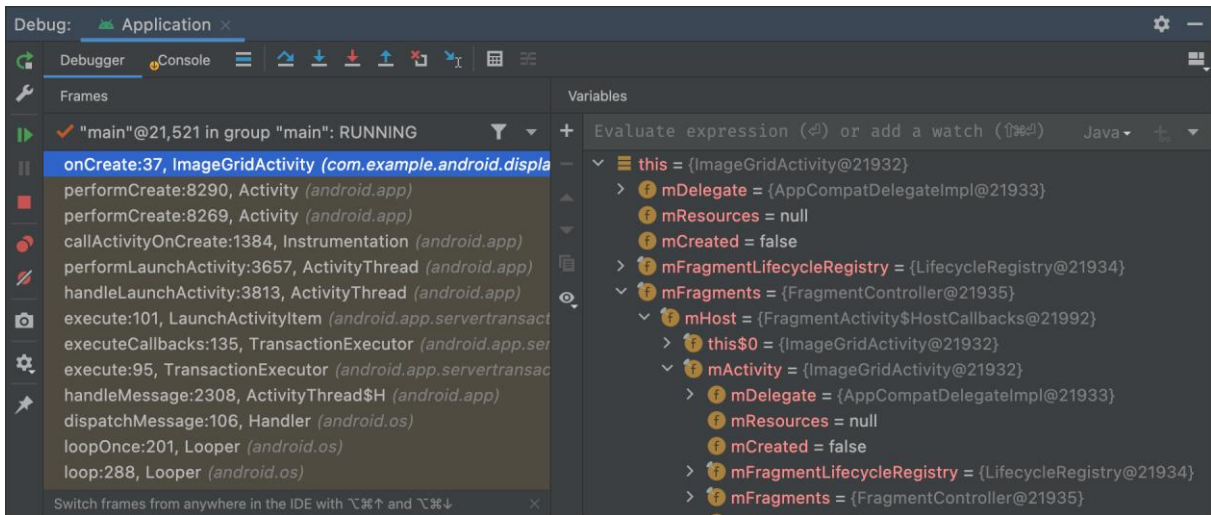
Εικόνα 2.7.6 Logcat

Μια άλλη δημοφιλής μέθοδος εντοπισμού σφαλμάτων μιας εφαρμογής είναι η χρήση σημείων διακοπής (Breakpoints). Τα σημεία διακοπής είναι δείκτες που μπορούν να τοποθετηθούν στον κώδικα και διακόπτουν την εκτέλεση της εφαρμογής όταν φτάσει στο σημείο διακοπής, επιτρέποντάς την επιθεώρηση της κατάστασης της εφαρμογής σε αυτό το σημείο και τον εντοπισμό τυχόν σφαλμάτων. Για να χρησιμοποιηθούν τα σημεία διακοπής, πρέπει να επιλεγεί η γραμμή κώδικα του επιθυμητού σημείου διακοπής και, στη συνέχεια, να γίνει κλικ στο αυλάκι δίπλα στον αριθμό γραμμής. Στη συνέχεια, μπορεί να εκτελεστεί η εφαρμογή σε λειτουργία εντοπισμού σφαλμάτων και η εκτέλεση θα σταματήσει στο σημείο διακοπής.

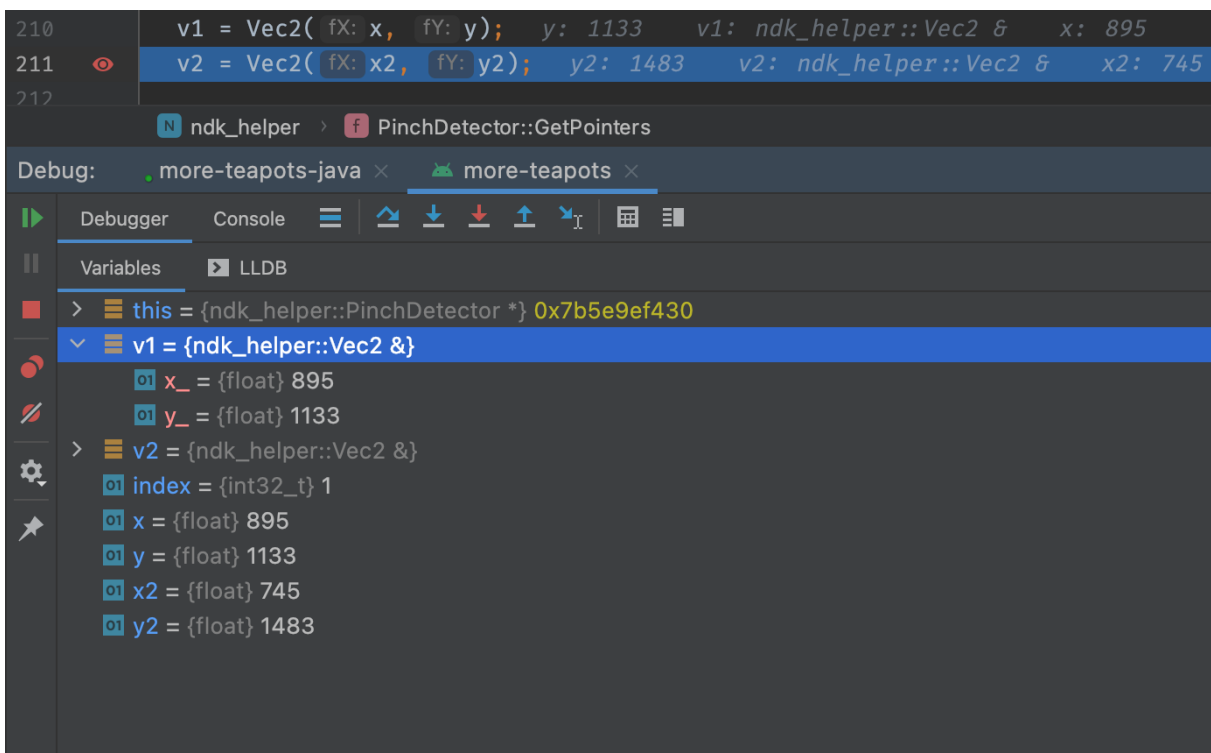


Εικόνα 2.7.7 Breakpoint

Τέλος, το Android Studio παρέχει ένα ισχυρό εργαλείο για τον εντοπισμό σφαλμάτων που ονομάζεται Debugger. Αυτό το εργαλείο επιτρέπει να επιθεωρηθεί η κατάσταση της εφαρμογής κατά το χρόνο εκτέλεσης αξιολογώντας εκφράσεις, περνώντας τον κώδικα και επιθεωρώντας μεταβλητές. Για να χρησιμοποιηθεί το πρόγραμμα εντοπισμού σφαλμάτων, πρέπει να γίνει εκκίνηση της εφαρμογής σε λειτουργία εντοπισμού σφαλμάτων και να γίνει κλικ στο κουμπί Εντοπισμός σφαλμάτων στη γραμμή εργαλείων για τη χρήση της λειτουργίας εντοπισμού σφαλμάτων. Στη συνέχεια, μπορεί να χρησιμοποιηθούν τα διάφορα εργαλεία εντοπισμού σφαλμάτων, όπως το περπάτημα, το βήμα μέσα και το βήμα έξω, για τη περιήγηση στον κώδικα και τη επιθεώρηση της κατάστασης της εφαρμογής.<sup>48</sup>



Εικόνα 2.7.8 Debugger Επιθεώρηση Μεταβλητών



Εικόνα 2.7.9 Debugger Σημείο Παρακολούθησης

## 2.8 Προετοιμασία της Εφαρμογής για Κυκλοφορία

Τα καταστήματα εφαρμογών για κινητά είναι τα κύρια κανάλια διανομής για εφαρμογές για κινητές συσκευές. Αυτά τα καταστήματα παρέχουν μια πλατφόρμα για τους προγραμματιστές για να παρουσιάσουν τις εφαρμογές τους σε ένα παγκόσμιο κοινό, διευκολύνοντας τους χρήστες να τις ανακαλύψουν και να τις κατεβάσουν Το Android Play

Store είναι το μεγαλύτερο κατάστημα εφαρμογών στον κόσμο με εκατομμύρια εφαρμογές διαθέσιμες για λήψη.

Για την δημοσίευση μιας εφαρμογής στο Android Play Store, θα πρέπει να γίνει εγγραφή ως προγραμματιστής. Για να επιτευχθεί αυτό, μεταβαίνουμε στον ιστότοπο του Google Play Console και συνδεόμαστε με τον Google λογαριασμό μας. Εάν δεν υπάρχει λογαριασμό Google, δημιουργούμε έναν. Μόλις συνδεθούμε, κάνουμε κλικ στο «Δημιουργία λογαριασμού προγραμματιστή» και ακολουθούμε τις οδηγίες για να ολοκληρώσουμε τη διαδικασία εγγραφής. Υπάρχει μια εφάπαξ χρέωση 25 \$ για να εγγραφεί ως προγραμματιστής. Μόλις γίνει η εγγραφή, θα έχουμε πρόσβαση στην πύλη προγραμματιστών όπου μπορούμε να διαχειριστούμε τις εφαρμογές μας, να παρακολουθούμε την απόδοση μιας εφαρμογής και να προβάλλουμε κριτικές και αξιολογήσεις χρηστών.

Αφού γίνει η εγγραφή ως προγραμματιστής, μπορεί να δημιουργηθεί μια νέα εφαρμογή κάνοντας κλικ στο "Δημιουργία εφαρμογής" στο Play Console. Θα χρειαστεί να παρέχουμε ορισμένες βασικές πληροφορίες σχετικά με την εφαρμογή, όπως το όνομα, την περιγραφή και την κατηγορία της. Θα χρειαστεί επίσης να ανεβάσουμε ένα εικονίδιο και στιγμιότυπα οθόνης που δείχνουν πώς μοιάζει η εφαρμογή.

Μπορεί να επιλεγεί μια εφαρμογή να είναι δωρεάν ή να έχει μια τιμή. Μπορεί επίσης να επιλεγεί να είναι διαθέσιμη σε όλους τους χρήστες ή να το περιοριστεί σε συγκεκριμένες χώρες ή περιοχές. Μπορεί επίσης να επιλεγεί να διανέμετε η εφαρμογή σε συγκεκριμένες συσκευές, όπως τηλέφωνα ή tablet ή τηλεοράσεις Android.

Για να μπορέσει να δημοσιευτεί μια εφαρμογή στο Play Store, πρέπει να προετοιμαστεί για κυκλοφορία. Αυτό περιλαμβάνει τη δοκιμή της εφαρμογής για να επιβεβαιωθεί ότι λειτουργεί σωστά και για τη διόρθωση τυχόν σφαλμάτων ή προβλημάτων. Θα πρέπει επίσης να βελτιστοποιηθεί η απόδοση και η εμπειρία χρήστη της εφαρμογής.

Αφού προετοιμαστεί η εφαρμογή για κυκλοφορία, μπορεί να ανεβεί στο Play Store. Αυτό επιτυγχάνεται κάνοντας κλικ στο "Μεταφόρτωση APK" στο Play Console για να ανεβεί το αρχείο APK της εφαρμογής. Μπορούν επίσης να ανεβάσουν επιπλέον αρχεία, όπως αρχεία επέκτασης ή πακέτα εφαρμογών.

Αφού ανέβει η εφαρμογή, μπορεί να δημοσιευτεί στο Play Store κάνοντας κλικ στο κουμπί «Δημοσίευση εφαρμογής». Στη συνέχεια, η εφαρμογή θα ελεγχθεί από την ομάδα του Google Play για να διασφαλιστεί ότι πληροί τις οδηγίες και τις πολιτικές του Play Store. Εάν η εφαρμογή εγκριθεί, θα είναι διαθέσιμη για λήψη στο Android Play Store.

Η δημοσίευση μιας εφαρμογής στο Android Google Play Store μπορεί να είναι μια περίπλοκη διαδικασία, αλλά είναι ένα ουσιαστικό βήμα για να γίνει μια εφαρμογή διαθέσιμη σε ένα ευρύ κοινό.<sup>49,50,51,52,53</sup>

## 2.9 Παράδειγμα κώδικα σε Java και Kotlin

Για την ανάπτυξη εφαρμογών για Android, οι προγραμματιστές μπορούν να επιλέξουν τη χρήση Java ή Kotlin ως γλώσσα προγραμματισμού. Η εφαρμογή «Επίλυσης του προβλήματος του Μέλανος Σώματος (Black Body)» η αρχική υλοποίηση έγινε σε Java ενώ αργότερα μεταφέρθηκε σε Kotlin. Παρακάτω παρουσιάζονται κάποια κομμάτια κώδικα της εφαρμογής υλοποιημένα σε Java και Kotlin.

Java	Kotlin
<pre>private void drawRadiusCircle () {     int radius = (int)(RD*11);     paint = new Paint();     if (T&gt;=1000 &amp;&amp; T&lt;3500){         paint.setColor(Color.RED);     }else if(T&gt;=3500 &amp;&amp; T&lt;5000){         paint.setColor(Color.rgb(255, 79, 0));     }else if(T&gt;=5000 &amp;&amp; T&lt;6000){         paint.setColor(Color.YELLOW);     }else if(T&gt;=6000 &amp;&amp; T&lt;7500){         paint.setColor(Color.rgb(255,255,153));     }else if(T&gt;=7500 &amp;&amp; T&lt;11000) {         paint.setColor(Color.WHITE);     }else if(T&gt;=11000 &amp;&amp; T&lt;25000) {         paint.setColor(Color.rgb(65,105,225));     }else if(T&gt;=25000 &amp;&amp; T&lt;=50000) {         paint.setColor(Color.BLUE);     } }  paint.setStyle(Paint.Style.FILL_AND_STROKE); bmp = Bitmap.createBitmap(455, 455, Bitmap.Config.ARGB_8888); canvas = new Canvas(bmp); canvas.drawCircle(bmp.getWidth()/2, bmp.getHeight()/2,radius,paint); imgcircle.setImageBitmap(bmp); }</pre>	<pre>private fun drawRadiusCircle() {     val drawRadius: Float =     (_radius.value!! * 11).toFloat()     val bitmap: Bitmap =     Bitmap.createBitmap(455, 455,     Bitmap.Config.ARGB_8888)     val canvas = Canvas(bitmap)     paint.color = when     (_temperature.value) {         in 1000..3499 -&gt; Color.RED         in 3500..4999 -&gt; Color.rgb(255, 79,         0)         in 5000..5999 -&gt; Color.YELLOW         in 6000..7499 -&gt; Color.rgb(255,         255, 153)         in 7500..10999 -&gt; Color.WHITE         in 11000..24999 -&gt; Color.rgb(65,         105, 225)         in 25000..50000 -&gt; Color.BLUE         else -&gt; Color.TRANSPARENT     }      paint.style =     Paint.Style.FILL_AND_STROKE     canvas.drawCircle(     (bitmap.width / 2).toFloat(),     (bitmap.height / 2).toFloat(),     drawRadius,</pre>

	<pre> paint ) _bitmap.postValue(bitmap) } </pre>
<pre> private void updateClassType (int scl){     spcla.setTypeface(null,Typeface.BOLD);     if(scl&lt;3500){         spcla.setText("Spectral Type: M");     }else if((scl &gt;= 3500) &amp;&amp; (scl &lt; 5000)){         spcla.setText("Spectral Type: K");     }else if((scl &gt;= 5000) &amp;&amp; (scl &lt; 6000)){         spcla.setText("Spectral Type: G");     }else if((scl &gt;= 6000) &amp;&amp; (scl &lt; 7500)){         spcla.setText("Spectral Type: F");     }else if((scl &gt;= 7500) &amp;&amp; (scl &lt; 10000)) {         spcla.setText("Spectral Type: A");     }else if((scl &gt;= 10000) &amp;&amp; (scl &lt; 30000)) {         spcla.setText("Spectral Type: B");     }else if((scl &gt;= 30000) &amp;&amp; (scl &lt; 60000)) {         spcla.setText("Spectral Type: O");     } } </pre>	<pre> private fun updateClassType() {     _classType.postValue(         when (_ temperature.value) {             in 0..3499 -&gt; "Spectral Type: M"             in 3500..4999 -&gt; "Spectral Type: K"             in 5000..5999 -&gt; "Spectral Type: G"             in 6000..7499 -&gt; "Spectral Type: F"             in 7500..9999 -&gt; "Spectral Type: A"             in 10000..29999 -&gt; "Spectral Type: B"             in 30000..60000 -&gt; "Spectral Type: O"             else -&gt; "Unknown Spectral Type"         }     ) } </pre>
<pre> violetSeries = new LineGraphSeries&lt;&gt;(new DataPoint[]){     new DataPoint(0.40/0.1,0),     new DataPoint(0.40/0.1, yma),     new DataPoint(0.42/0.1, yma),     new DataPoint(0.42/0.1,0) }); violetSeries.setColor(Color.rgb(128, 0, 255)); violetSeries.setDrawBackground(true);  violetSeries.setBackgroundColor(Color.rgb(128, 0, 255)); </pre>	<pre> val spectrumColor = Color.rgb(128, 0, 255) val violetSeries = LineGraphSeries(     arrayOf(         DataPoint(4.0, 0.0),         DataPoint(4.0, yTopPoint),         DataPoint(4.2, yTopPoint),         DataPoint(4.2, 0.0)     ) ) violetSeries.apply {     color = spectrumColor     isDrawBackground = true     backgroundColor = spectrumColor } </pre>
<pre> private void graphSetup () {     viewport.setMinY(ymi);     viewport.setMaxY(yma);     viewport.setMinX(xmi);     viewport.setMaxX(xma);     viewport.setYAxisBoundsManual(true);     viewport.setXAxisBoundsManual(true);     viewport.setScrollable(true);     viewport.setScrollableY(true);     viewport.setScalable(true);     viewport.setScalableY(true); } </pre>	<pre> private fun graphSetup() {     viewport.setMinY(yMin)     viewport.setMinX(xMin)     viewport.setMaxX(xMax)     viewport.isYAxisBoundsManual = true     viewport.isXAxisBoundsManual = true     viewport.isScrollable = true     viewport.setScrollableY(true)     viewport.isScalable = true     viewport.setScalableY(false) } </pre>

<pre>private void wavelengthChangeVisibilityHandler (){     if (hidexy==true) {         inxma.setVisibility(View.VISIBLE);         //inyma.setVisibility(View.VISIBLE);         inxmi.setVisibility(View.VISIBLE);         //inymi.setVisibility(View.VISIBLE);         chvalxy.setVisibility(View.VISIBLE);         sbsun.setVisibility(View.INVISIBLE);         hidexy=false;     }else if (hidexy==false){         inxma.setVisibility(View.INVISIBLE);         inyma.setVisibility(View.INVISIBLE);         inxmi.setVisibility(View.INVISIBLE);         inymi.setVisibility(View.INVISIBLE);         chvalxy.setVisibility(View.INVISIBLE);         sbsun.setVisibility(View.VISIBLE);         hidexy=true;     } }</pre>	<pre>private fun wavelengthChangeVisibilityHandler() {     if (hideXY) {         binding.inputXMax.visibility =         GONE         binding.inputXMin.visibility = GONE          binding.applyChangesButton.visibility =         GONE         binding.sunRadiusSeekBar.visibility         = VISIBLE         binding.sunRadiusTextview.visibility         = VISIBLE     } else {         binding.inputXMax.visibility =         VISIBLE         binding.inputXMin.visibility =         VISIBLE          binding.applyChangesButton.visibility =         VISIBLE         binding.sunRadiusSeekBar.visibility         = GONE         binding.sunRadiusTextview.visibility         = GONE     }     hideXY = !hideXY }</pre>
--	---

### 3. Η Προτεινόμενη Εφαρμογή

#### 3.1 Θεωρητικό Πλαίσιο

Στις αρχές του 1900, οι φυσικοί πάλευαν να κατανοήσουν τη συμπεριφορά του φωτός που εκπέμπεται από θερμά αντικείμενα, η οποία ήταν γνωστή ως ακτινοβολία μέλανος σώματος. Παρατηρήθηκε ότι το εκπεμπόμενο φως εξαρτιόταν από τη θερμοκρασία του αντικειμένου, αλλά οι υπάρχουσες θεωρίες απέτυχαν να εξηγήσουν την ακριβή φύση αυτής της εξάρτησης.

Το 1900, ο Max Planck πρότεινε μια πρωτοποριακή θεωρία που εξηγούσε το φάσμα ακτινοβολίας του μέλανος σώματος. Ο νόμος του Planck δηλώνει ότι η ενέργεια ενός φωτονίου φωτός είναι ευθέως ανάλογη με τη συχνότητά του και ότι η συνολική ενέργεια που ακτινοβολείται από ένα μέλαν σώμα είναι συνάρτηση της θερμοκρασίας του. Ο Planck εξήγαγε μια εξίσωση που περιέγραψε με ακρίβεια την κατανομή της ακτινοβολίας που εκπέμπεται από ένα μέλαν σώμα σε οποιαδήποτε δεδομένη θερμοκρασία.

Η εξίσωση δίνεται ως εξής:

$$B(\lambda, T) = 2hc^2/\lambda^5 * 1/(e^{(hc/\lambda kT)} - 1)$$

όπου  $B(\lambda, T)$  είναι η φασματική ακτινοβολία του μέλαν σώματος σε δεδομένο μήκος κύματος  $\lambda$  και θερμοκρασία  $T$ ,  $h$  είναι η σταθερά του Planck,  $c$  είναι η ταχύτητα του φωτός,  $k$  είναι η σταθερά του Boltzmann και  $e$  η βάση των φυσικών λογαρίθμων.

Ο νόμος του Planck περιγράφει τη φασματική κατανομή της ενέργειας που εκπέμπεται από ένα μέλαν σώμα ως συνάρτηση της θερμοκρασίας του και χρησιμοποιείται ευρέως σε πεδία όπως η αστροφυσική, η κοσμολογία και η κβαντική μηχανική.

Ένας άλλος σημαντικός νόμος που σχετίζεται με την ακτινοβολία του μέλανος σώματος είναι ο νόμος του Wilhelm Wien. Αυτός ο νόμος δηλώνει ότι το μήκος κύματος στο οποίο η φασματική ακτινοβολία ενός μέλανος σώματος είναι μέγιστη είναι αντιστρόφως ανάλογο της θερμοκρασίας του. Ο νόμος αυτός εκφράζεται ως εξής:

$$\lambda_{max} = b/T$$

όπου  $\lambda_{max}$  είναι το μήκος κύματος της μέγιστης φασματικής ακτινοβολίας,  $T$  είναι η θερμοκρασία του μέλαν σώματος και  $b$  είναι η σταθερά μετατόπισης του Wien.

Ο νόμος Stefan-Boltzmann είναι ένας άλλος σημαντικός νόμος που σχετίζεται με την ακτινοβολία του μέλανος σώματος. Αυτός ο νόμος λέει ότι η συνολική ενέργεια που ακτινοβολείται ανά μονάδα επιφάνειας ενός μέλανος σώματος ανά μονάδα χρόνου είναι ανάλογη με την τέταρτη δύναμη της απόλυτης θερμοκρασίας του. Ο νόμος αυτός εκφράζεται ως εξής:

$$J = \sigma T^4$$

όπου  $J$  είναι η συνολική ενέργεια που ακτινοβολείται ανά μονάδα επιφάνειας ανά μονάδα χρόνου,  $\sigma$  είναι η σταθερά Stefan-Boltzmann και  $T$  είναι η θερμοκρασία του μέλαν σώματος.

Εκτός από τους νόμους που αναφέρθηκαν, το φάσμα ακτινοβολίας μέλανος σώματος έχει πολλές άλλες ενδιαφέρουσες ιδιότητες. Για παράδειγμα, το φάσμα είναι συνεχές και δεν εξαρτάται από το υλικό του μέλαν σώματος, αρκεί να βρίσκεται σε θερμική ισορροπία με το περιβάλλον του. Αυτό σημαίνει ότι το φάσμα είναι το ίδιο για όλα τα μέλαν σώματα στην ίδια θερμοκρασία, ανεξάρτητα από τη σύνθεσή τους.

Συμπερασματικά, το φάσμα ακτινοβολίας μέλανος σώματος περιγράφει την κατανομή της ακτινοβολίας που εκπέμπεται από ένα μέλαν σώμα ως συνάρτηση της θερμοκρασίας του. Ο νόμος του Planck, ο νόμος του Wien και ο νόμος Stefan-Boltzmann είναι τρεις σημαντικοί νόμοι που περιγράφουν τη συμπεριφορά της ακτινοβολίας του μέλαν σώματος. Αυτοί οι νόμοι έχουν σημαντικές εφαρμογές σε διάφορους τομείς της φυσικής, συμπεριλαμβανομένης της αστροφυσικής και της κοσμολογίας καθώς και στους τομείς της τεχνολογίας.<sup>54,55,56,60</sup>

### **3.2 Η Βιβλιοθήκη Graph View**

Το Graph View είναι μια βιβλιοθήκη ανοιχτού κώδικα σε Java που παρέχει έναν εύκολο και προσαρμόσιμο τρόπο σχεδίασης γραφημάτων για Android εφαρμογές. Η βιβλιοθήκη έχει αναπτυχθεί από τον Jonas Gehring και είναι διαθέσιμη στο GitHub με την άδεια Apache 2.0.

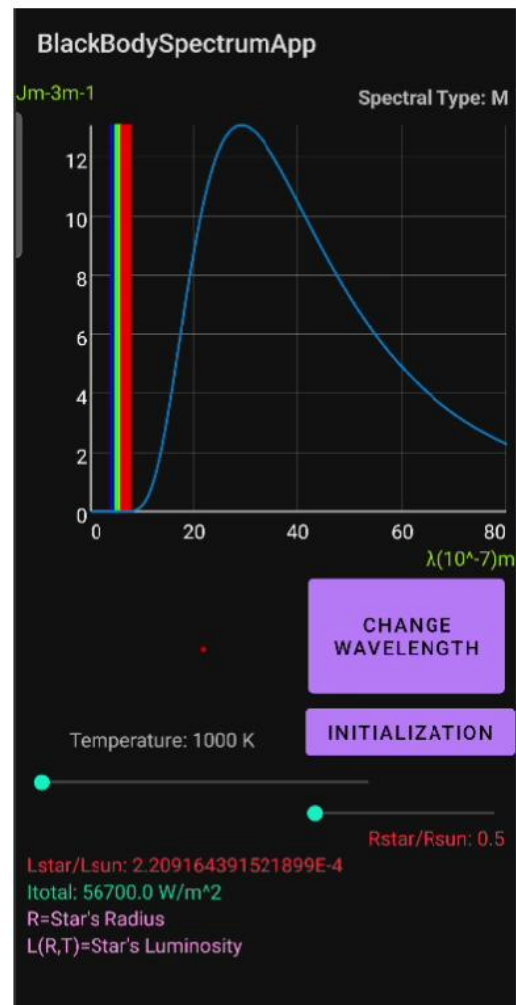
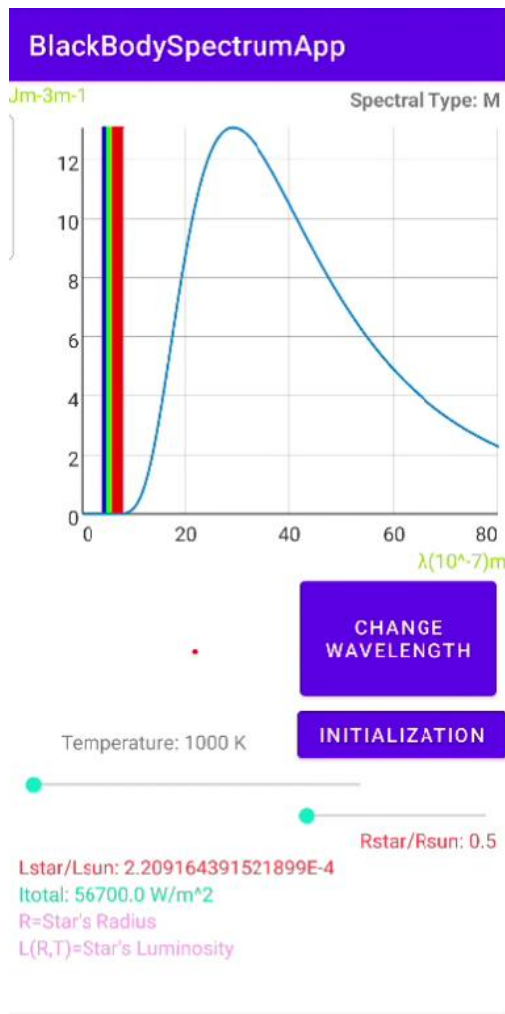
Το Graph View έχει σχεδιαστεί για να είναι ελαφρύ και εύκολο στη χρήση, καθιστώντας το ιδανικό για την προσθήκη βασικών δυνατοτήτων γραφημάτων στις εφαρμογές τους Android. Προσφέρει μια σειρά γραφημάτων, συμπεριλαμβανομένων γραμμικών γραφημάτων, γραφημάτων ράβδων και γραφημάτων πίτας και υποστηρίζει μια ποικιλία προσαρμόσιμων χαρακτηριστικών, όπως χρώματα, γραμματοσειρές και ετικέτες αξόνων.

Η βιβλιοθήκη έχει σχεδιαστεί για να είναι εξαιρετικά βελτιστοποιημένη για απόδοση, πράγμα που σημαίνει ότι μπορεί να χειριστεί μεγάλα σύνολα δεδομένων και πολύπλοκες λειτουργίες γραφημάτων χωρίς να επιβραδύνει ή να προκαλεί προβλήματα απόδοσης. Αυτό το καθιστά ιδανική επιλογή για εφαρμογές που πρέπει να εμφανίζουν μεγάλες ποσότητες δεδομένων ή που απαιτούν δυνατότητες απεικόνισης σε πραγματικό χρόνο.<sup>57,58</sup>

### **3.3 Τμήματα της Εφαρμογής**

Η εφαρμογή «Επίλυσης του προβλήματος του Μέλανος Σώματος (Black Body)» είναι μια εφαρμογή Android γραμμένη σε Android Studio με Kotlin και χρησιμοποιεί τη βιβλιοθήκη GraphView, η οποία είναι μια βιβλιοθήκη Android ανοιχτού κώδικα, για τη δημιουργία ευέλικτων διαγραμμάτων.





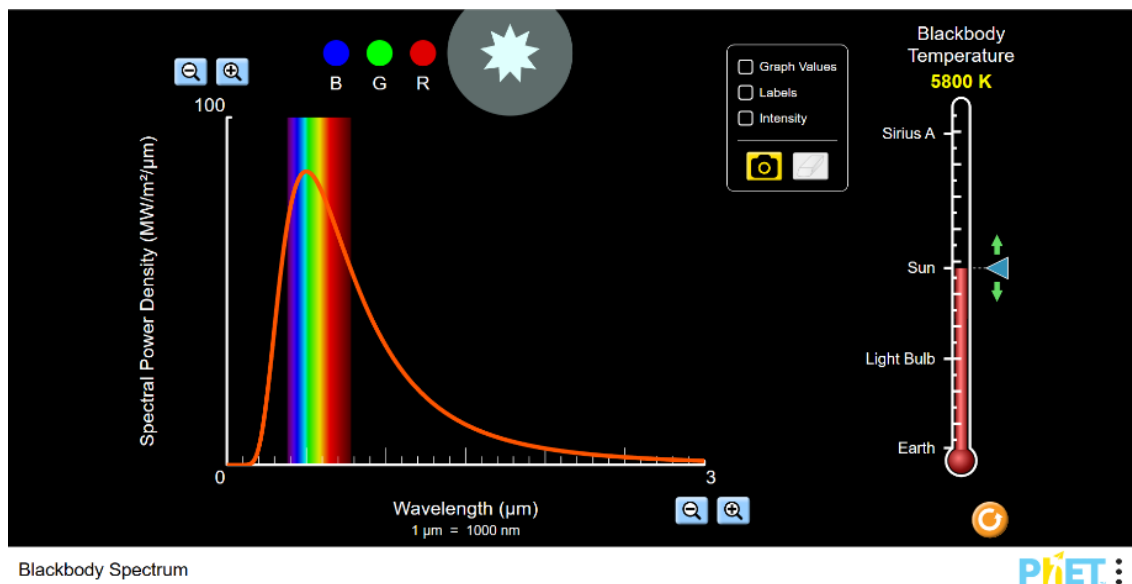
Εικόνα 3.3 Οθόνη της Εφαρμογής σε Light και Dark Θέμα

Η εφαρμογή διαθέτει δισδιάστατη κλιμακούμενη προβολή γραφήματος και δύο κουμπιά - το ένα για την προσαρμογή της κλίμακας του άξονα X και το άλλο για την επαναφορά της κλίμακας στην προεπιλογή, εάν εφαρμοστεί μεγέθυνση ή σμίκρυνση στην προβολή γραφήματος. Η εφαρμογή διαθέτει επίσης δύο μπάρες για την προσαρμογή του αστερά, μία μεγάλη για αλλαγή θερμοκρασίας από 1000K έως 21000K με βήμα 200K και η άλλη για αλλαγή της ακτίνας του αστερά από 0.5 έως 15.5. Πάνω από μπάρα της θερμοκρασίας υπάρχει μια γραφική αναπαράσταση της ακτίνας αστεριού που εξαρτάται από τη θερμοκρασία και την ακτίνα του Άστρου. Στην εφαρμογή υπάρχουν πέντε μεταβλητοί δείκτες για τη θερμοκρασία, την ακτίνα και τη φωτεινότητα του αστεριού, τη ένταση και ο φασματικός τύπος.

### 3.4 Παρόμοια Λογισμικά

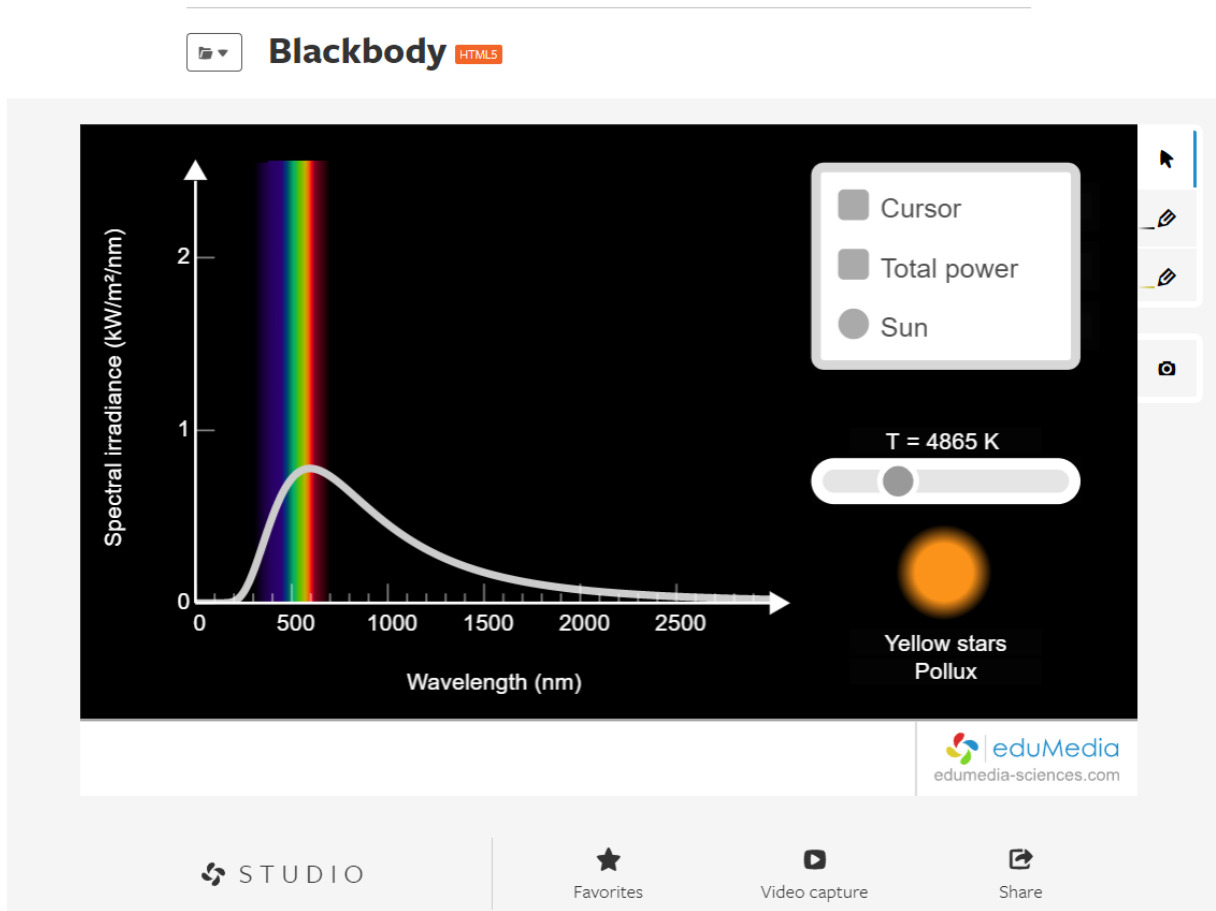
Η μελέτη του φάσματος ακτινοβολίας του μέλανος σώματος έχει συμβάλει σε πολλές σημαντικές ανακαλύψεις και έχει πολυάριθμες πρακτικές εφαρμογές. Οι νόμοι του Planck, του Wien και του Stefan-Boltzmann είναι μόνο μερικά παραδείγματα των σημαντικών αρχών που διέπουν την ακτινοβολία του μαύρου σώματος. Υπάρχουν πολλές εφαρμογές σαν και αυτήν που επιλύουν και προσομοιώνουν την καμπύλη του του φάσματος ακτινοβολίας του μέλανος σώματος.

Η εφαρμογή προσομοίωση φάσματος μέλανος σώματος από το εκπαιδευτικό ίδρυμα Phet του Colorado είναι μια διαδραστική προσομοίωση που επιτρέπει τη ρύθμιση της θερμοκρασίας ενός μέλανος σώματος και τη παρατήρηση της αλλαγής του φάσματος. Παρέχει πληροφορίες για το φάσμα μέλανος σώματος του Σείριου (Sirius) A, τον ήλιο, έναν λαμπτήρα και τη γη. Επίσης είναι εμφανές το χρώμα της κορυφής της φασματικής καμπύλης.<sup>59</sup>

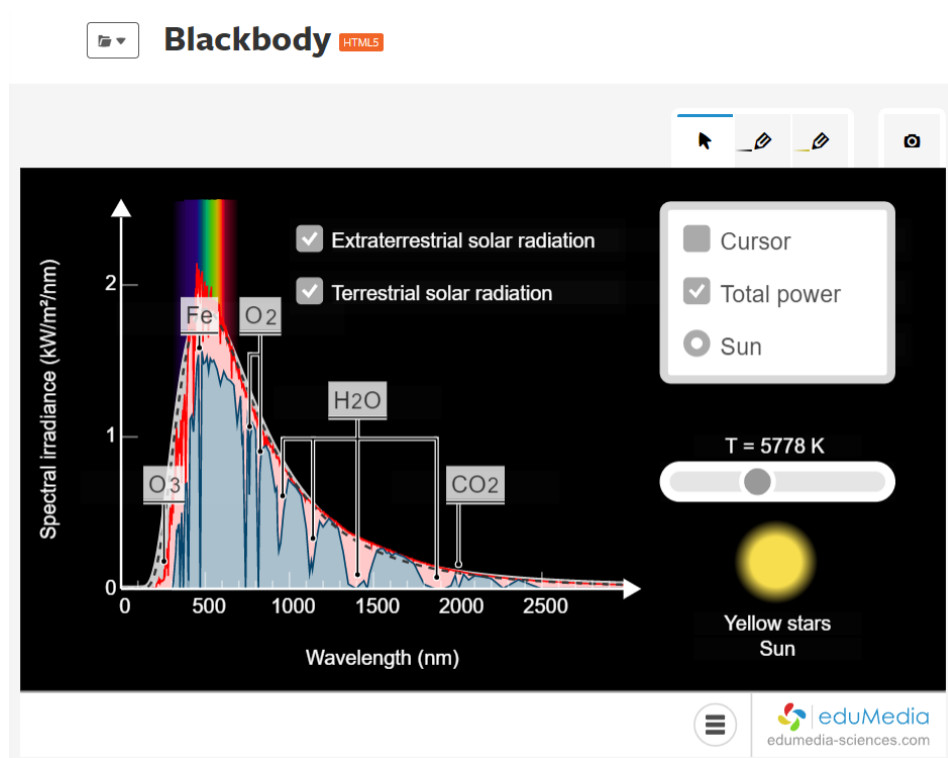


Εικόνα 3.4.1 Διαδραστική Εφαρμογή Phet

Η προσομοίωση μαύρου σώματος από το eduMedia είναι μια διαδραστική προσομοίωση που εξηγεί τι είναι ένα μαύρο σώμα. Παρέχει επίσης πληροφορίες για το τύπο του αστέρα ανάλογα με την θερμοκρασία.<sup>60</sup>

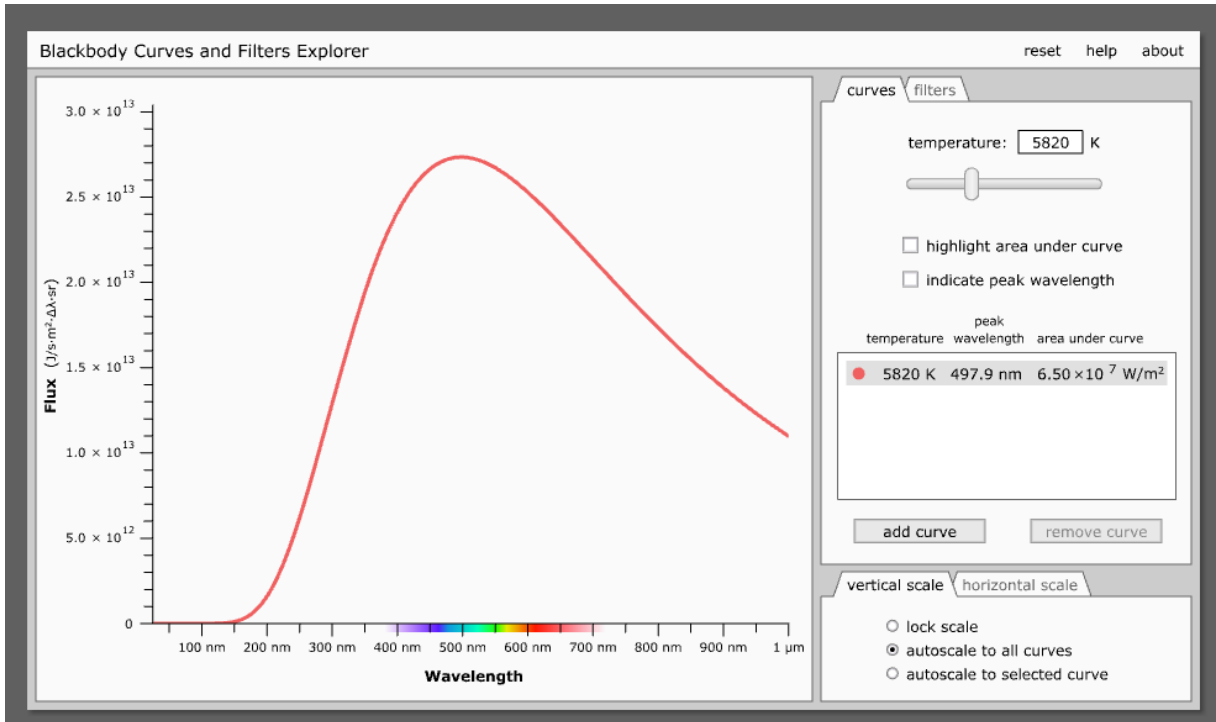


Εικόνα 3.4.2 Διαδραστική Εφαρμογή eduMedia



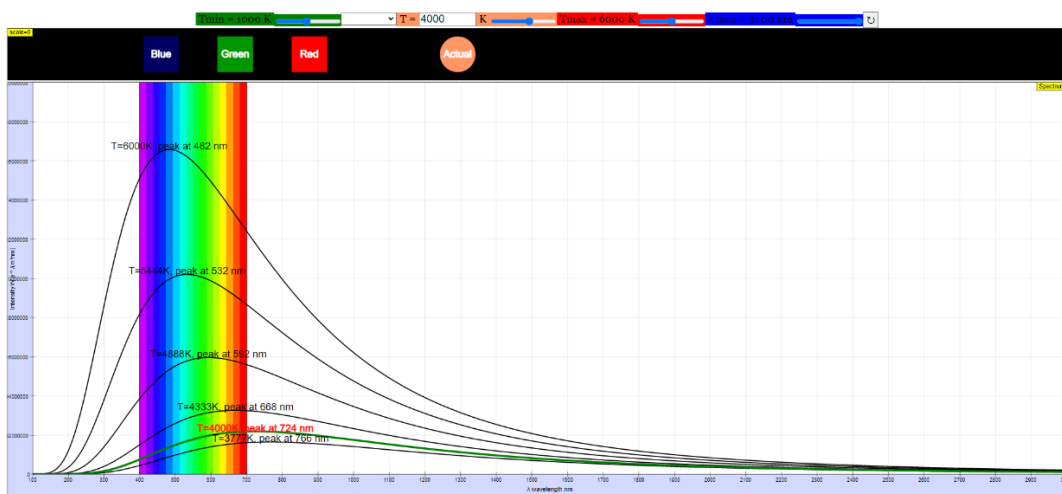
Εικόνα 3.4.3 Διαδραστική Εφαρμογή eduMedia με ακτινοβολία Ήλιου

Η Εφαρμογή του πανεπιστημίου του Lincoln της Νεμπράσκα παρείχατε διαδραστική αναπαράσταση του φάσμα του μέλαν σώματος που διαμορφώνεται ανάλογα με τη θερμοκρασία. Δείχνει επίσης πώς οι σχετικές εντάσεις που παρατηρούνται μέσω διαφορετικών φίλτρων (ένας «δείκτης χρώματος») μπορούν να δώσουν μια εκτίμηση της θερμοκρασίας.<sup>61</sup>



Εικόνα 3.4.4 Διαδραστική Εφαρμογή Nebraska-Lincoln

Η Εφαρμογή του iwant2study είναι μια απλή προσομοίωση του φάσματος ακτινοβολίας μέλαν σώματος.<sup>62</sup>



Εικόνα 3.4.5 Διαδραστική Εφαρμογή iwant2study

### **3.5 Μελλοντικές Προσθήκες**

Το πεδίο του προβλήματος που ασχολήθηκε η παρούσα πτυχιακή εργασία είναι η διαδραστική ανήκον του φάσματος της ακτινοβολίας ενός μέλανος σώματος και κάποιες μελλοντικές προσθήκες στην εφαρμογή θα μπορούσαν είναι η προσθήκη πολλαπλών απεικονίσεων κυματομορφών καθώς και προσθήκη και ρυθμίσεων ώστε να έχει ο χρήστης μεγαλύτερη παραμετροποίηση των παραγόντων της εφαρμοστής καθώς και η προσθήκη περιστέρων πληροφοριών.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. *History of Mobile Phones: From Evolution to a Revolution*. (n.d.). Retrieved April 4, 2023, from <https://blog.talkhome.co.uk/technology/history-of-mobile-phones/>
2. *Future Mobile Phones*. (2006). [www.teletronikk.com](http://www.teletronikk.com)  
[https://d1wqtxts1xzle7.cloudfront.net/30805674/T05\\_3-4-libre.pdf?1392146498=&response-content-disposition=inline%3B+filename%3DInterest+in+future+net+based+services+fo.pdf&Expires=1680471185&Signature=NtSr2ZIGGfpaoTVv1fQCEbF5gRIAiYhHiqHT2PyrfbQjtxX9A5HgoSEk-FILJ69p4~6vEJgb-cBT7xzboGL8zj5vOPTbNhHvTKrICHfUmds~5P~MS4~U4v00bjyQ-MS4JjA-6KqpszgJA9l6kvYVN-KzfZE74sTCDX7~0wpCRroCXU~i7J3rzwMmzXgkEz-tZFrUGJjvQxg-armokpUK2MQCwm~bm4aolXawFJKgRxdJ4oriO1yCIMhtuKzZImCSJZIUV28l36CEwWO3y-ZxrQd76lL1gOlf8rfEM6sWBWw4~juaxY3nVCADtVHLfBBLee4aS2F1nfKjG6jyA7iNt6g\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=24](https://d1wqtxts1xzle7.cloudfront.net/30805674/T05_3-4-libre.pdf?1392146498=&response-content-disposition=inline%3B+filename%3DInterest+in+future+net+based+services+fo.pdf&Expires=1680471185&Signature=NtSr2ZIGGfpaoTVv1fQCEbF5gRIAiYhHiqHT2PyrfbQjtxX9A5HgoSEk-FILJ69p4~6vEJgb-cBT7xzboGL8zj5vOPTbNhHvTKrICHfUmds~5P~MS4~U4v00bjyQ-MS4JjA-6KqpszgJA9l6kvYVN-KzfZE74sTCDX7~0wpCRroCXU~i7J3rzwMmzXgkEz-tZFrUGJjvQxg-armokpUK2MQCwm~bm4aolXawFJKgRxdJ4oriO1yCIMhtuKzZImCSJZIUV28l36CEwWO3y-ZxrQd76lL1gOlf8rfEM6sWBWw4~juaxY3nVCADtVHLfBBLee4aS2F1nfKjG6jyA7iNt6g_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=24)
3. Li, X., Ortiz, P. J., Browne, J., Franklin, D., Oliver, J. Y., Geyerz, R., Zhou, Y., & Chong, F. T. (2010). Smartphone evolution and reuse: Establishing a more sustainable model. *Proceedings of the International Conference on Parallel Processing Workshops*, 476–484. <https://doi.org/10.1109/ICPPW.2010.70>
4. Islam, N., & Want, R. (2014). Smartphones: Past, present, and future. *IEEE Pervasive Computing*, 13(4), 89–92. <https://doi.org/10.1109/MPRV.2014.74>
5. What is Android <https://www.android.com/what-is-android/>
6. Iggy Krajci, & Darren Cummings. (2014). *History and Evolution of the Android OS*. [https://link.springer.com/chapter/10.1007/978-1-4302-6131-5\\_1](https://link.springer.com/chapter/10.1007/978-1-4302-6131-5_1)
7. Haris, M., Jadoon, B., Yousaf, M., & Hassan Khan, F. (2017). *EVOLUTION OF ANDROID OPERATING SYSTEM: A REVIEW*. [www.apiar.org.au](http://www.apiar.org.au)
8. Shaheen, J. A., Asghar, M. A., & Hussain, A. (2017). Android OS with its Architecture and Android Application with Dalvik Virtual Machine Review. *International Journal of Multimedia and Ubiquitous Engineering*, 12(7), 19–30. <https://doi.org/10.14257/ijmue.2017.12.7.03>
9. Hall, S. P., & Anderson, E. (2009). *OPERATING SYSTEMS FOR MOBILE COMPUTING* \*.
10. Alian, M., Guang, S., & Teng, B. (n.d.). *Android Operating System: An in depth introduction CS423 Project*. Retrieved April 3, 2023, from <https://source.android.com/>
11. *Android (operating system)*. (n.d.). Retrieved April 3, 2023, from [www.android.com](http://www.android.com)
12. Android Developer Platform Releases <https://developer.android.com/about/versions>
13. Brähler, S., & Brähler, B. (2010). *Analysis of the Android Architecture Studienarbeit von*. [www.kit.edu](http://www.kit.edu)
14. Android Developer Technology Platform Architecture [Platform Architecture | Android Developers](#)
15. Android Source Core Topics Architecture Overview [Architecture overview | Android Open Source Project](#)
16. Android Source Core Topics Kernel Overview [Kernel overview | Android Open Source Project](#)
17. Android Source Core Topics Hardware Abstraction Layer Overview <https://source.android.com/docs/core/architecture/hal>

18. Android Source Core Topics Android Runtime and Dalvik  
<https://source.android.com/docs/core/runtime>
19. What is DVM, Geeks For Geeks <https://www.geeksforgeeks.org/what-is-dvmdalvik-virtual-machine>
20. Android Developer Guides Android App Manifest Overview [App manifest overview | Android Developers](#)
21. *What is the Android manifest file?* (n.d.). Retrieved April 20, 2023, from <https://emteria.com/learn/android-manifest-file>
22. *Projects overview | Android Studio | Android Developers.* (n.d.). Retrieved April 24, 2023, from <https://developer.android.com/studio/projects>
23. *Application fundamentals | Android Developers.* (n.d.). Retrieved April 24, 2023, from <https://developer.android.com/guide/components/fundamentals>
24. Pryss, R., Reichert, M., Schlee, W., Spiliopoulou, M., Langguth, B., & Probst, T. (2018). Differences between Android and iOS Users of the TrackYourTinnitus Mobile Crowdsensing mHealth Platform. *Proceedings - IEEE Symposium on Computer-Based Medical Systems, 2018-June*, 411–416. <https://doi.org/10.1109/CBMS.2018.00078>
25. Wukkadada, B., Nambiar, R., Nair, A., Professor, A., & Somaiya, K. J. (2015). Mobile Operating System: Analysis and Comparison of Android and iOS. *IJCAT-International Journal of Computing and Technology*, 2(7). [www.IJCAT.org](http://www.IJCAT.org)
26. Sheikh, A. A., Ganai, T., Jhabarmal, J., Malik, N., & Ahmad, K. (n.d.). Smartphone: Android Vs IOS Framework for Opinion Based Product Recommender System View project High Dynamic Range Imaging View project. *The SIJ Transactions on Computer Science Engineering & Its Applications (CSEA)*, 1(4). <https://doi.org/10.9756/SIJCSEA/V1I4/0104600401>
27. Ahmad, M. S., Musa, N. E., Nadarajah, R., Hassan, R., & Othman, N. E. (2013). Comparison between android and iOS Operating System in terms of security. *2013 8th International Conference on Information Technology in Asia - Smart Devices Trend: Technologising Future Lifestyle, Proceedings of CITA 2013.* <https://doi.org/10.1109/CITA.2013.6637558>
28. *Android vs. iOS: Which Mobile OS Is Best? | PCMag.* (n.d.). Retrieved April 25, 2023, from <https://www.pcmag.com/comparisons/android-vs-ios-which-mobile-os-is-best>
29. *Android vs. iOS: A Comparison Guide - TurboFuture.* (n.d.). Retrieved April 25, 2023, from <https://turbofuture.com/cell-phones/ANDRIOD-VS-IOS>
30. *iOS vs. Android: Full Comparison with Specs, History, and More - History-Computer.* (n.d.). Retrieved April 25, 2023, from <https://history-computer.com/ios-vs-android/>
31. Singh, R. (2014). An Overview of Android Operating System and Its Security Features. *Journal of Engineering Research and Applications Wwww.Ijera.Com*, 4(2), 519–521. <http://ptcoresec.eu/2013/05/02/part-1-getting>
32. Ahmed, O. M., & Sallow, A. B. (2017). Android Security: A Review. *Academic Journal of Nawroz University*, 6(3), 135–140. <https://doi.org/10.25007/AJNU.V6N3A99>
33. Karthick, S., & Binu, S. (2017). Android security issues and solutions. *IEEE International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2017 - Proceedings*, 686–689. <https://doi.org/10.1109/ICIMIA.2017.7975551>
34. Enck, W., Ongtang, M., & McDaniel, P. (2009). Understanding android security. *IEEE Security and Privacy*, 7(1), 50–57. <https://doi.org/10.1109/MSP.2009.26>
35. *Android Studio Flamingo | 2022.2.1 | Android Developers.* (n.d.). Retrieved April 25, 2023, from <https://developer.android.com/studio/releases>



36. *Meet Android Studio | Android Developers*. (n.d.). Retrieved April 25, 2023, from <https://developer.android.com/studio/intro>
37. *Android Studio 2.2 Development Essentials - Android 7 Edition - Neil Smyth - Βιβλία Google*. (n.d.). Retrieved April 25, 2023, from [https://books.google.gr/books?hl=el&lr=&id=0vTBDAAAQBAJ&oi=fnd&pg=PT34&dq=android+studios&ots=T55X4p5hcZ&sig=BKFtCWmZWnvD-sLoR7pNBI6ejt4&redir\\_esc=y#v=onepage&q=android%20studios&f=false](https://books.google.gr/books?hl=el&lr=&id=0vTBDAAAQBAJ&oi=fnd&pg=PT34&dq=android+studios&ots=T55X4p5hcZ&sig=BKFtCWmZWnvD-sLoR7pNBI6ejt4&redir_esc=y#v=onepage&q=android%20studios&f=false)
38. Thamizharasi, R. (2016). Android Mobile Application Build on Android studio. *International Journal of Modern Computer Science (IJMCS) (Online)*, 4(1). <http://ijmcs.info>
39. *Install Android Studio | Android Developers*. (n.d.). Retrieved April 26, 2023, from <https://developer.android.com/studio/install>
40. *Create a project | Android Studio | Android Developers*. (n.d.). Retrieved May 1, 2023, from <https://developer.android.com/studio/projects/create-project>
41. *Get to know the Android Studio UI | Android Developers*. (n.d.). Retrieved May 1, 2023, from <https://developer.android.com/studio/intro/user-interface>
42. *Develop a UI with Views | Android Studio | Android Developers*. (n.d.). Retrieved May 1, 2023, from <https://developer.android.com/studio/write/layout-editor>
43. *Compose previews | Android Developers*. (n.d.). Retrieved May 1, 2023, from <https://developer.android.com/jetpack/compose/tooling/previews>
44. *Create and manage virtual devices | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/run/managing-avds>
45. *Build and run your app | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/run>
46. *Run apps on the Android Emulator | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/run/emulator>
47. *Build and run your app | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/run>
48. *Debug your app | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/debug>
49. *Publish your app | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/publish>
50. *Prepare your app for release | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/publish/preparing>
51. *Version your app | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/publish/versioning>
52. *Sign your app | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/publish/app-signing>
53. *Upload your app to the Play Console | Android Studio | Android Developers*. (n.d.). Retrieved May 2, 2023, from <https://developer.android.com/studio/publish/upload-bundle>
54. *Electromagnetic Radiation | ucolick.org* [https://www.ucolick.org/~woosley/lectures\\_winter2016/lecture7.16.pdf](https://www.ucolick.org/~woosley/lectures_winter2016/lecture7.16.pdf)
55. Kardaras, I., & Kallery, M. (2020). A teaching module for blackbody radiation using the continuous spectra of stars. *Physics Education*, 55(4), 045010. <https://doi.org/10.1088/1361-6552/AB853C>
56. 6.2: *Blackbody Radiation - Physics LibreTexts*. (n.d.). Retrieved May 7, 2023, from [https://phys.libretexts.org/Bookshelves/University\\_Physics/Book%3A\\_University\\_Physics\\_%28OpenStax%29/University\\_Physics\\_III\\_-](https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_University_Physics_%28OpenStax%29/University_Physics_III_-)



[Optics and Modern Physics %28OpenStax%29/06%3A Photons and Matter Waves/6.02%3A Blackbody Radiation](#)

57. *jjoe64/GraphView: Android Graph Library for creating zoomable and scrollable line and bar graphs*. (n.d.). Retrieved May 3, 2023, from <https://github.com/jjoe64/GraphView>
58. *Home · jjoe64/GraphView Wiki*. (n.d.). Retrieved May 3, 2023, from <https://github.com/jjoe64/GraphView/wiki>
59. *Blackbody Spectrum - Blackbody | Planck's Law | Wien's Law - PhET Interactive Simulations*. (n.d.). Retrieved May 7, 2023, from <https://phet.colorado.edu/en/simulations/blackbody-spectrum/about>
60. *Blackbody – interactive simulations – eduMedia*. (n.d.). Retrieved May 7, 2023, from <https://www.edumedia-sciences.com/en/media/944-blackbody>
61. *Blackbody Curves (NAAP)*. (n.d.). Retrieved May 7, 2023, from <https://astro.unl.edu/classaction/animations/light/bbexplorer.html>
62. *Blackbody Radiation Spectrum*. (n.d.). Retrieved May 7, 2023, from [https://iwant2study.org/lookangejss/04waves\\_13electromagneticspectrum/ejss\\_model\\_BlackbodyRadiationSpectrumwee/BlackbodyRadiationSpectrumwee\\_Simulation.xhtml](https://iwant2study.org/lookangejss/04waves_13electromagneticspectrum/ejss_model_BlackbodyRadiationSpectrumwee/BlackbodyRadiationSpectrumwee_Simulation.xhtml)

[Οπισθόφυλλο. Κενή σελίδα]

---

i