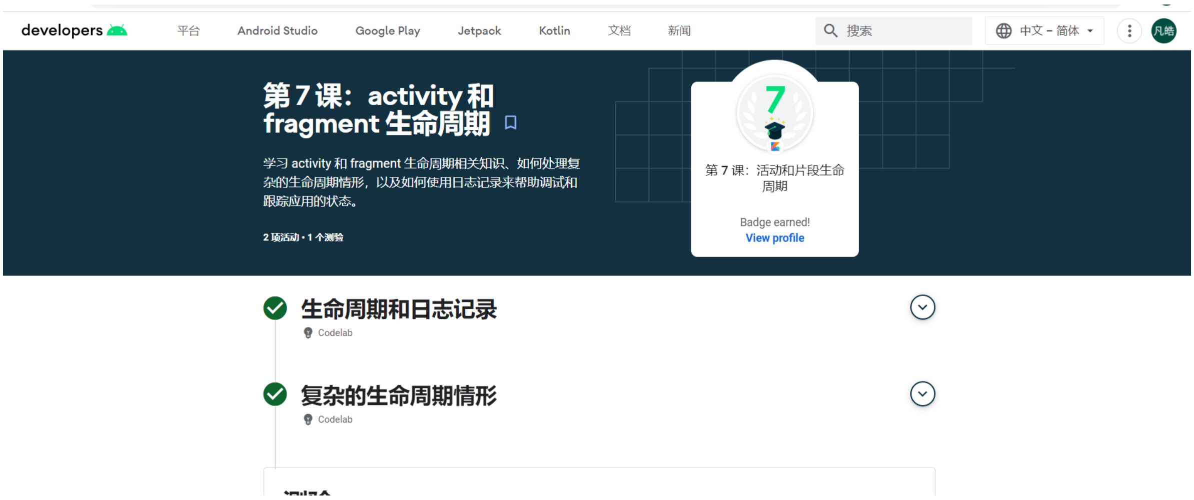


第 7 课：activity 和 fragment 生命周期

学习 activity 和 fragment 生命周期相关知识、如何处理复杂的生命周期情形，以及如何使用日志记录来帮助调试和跟踪应用的状态。

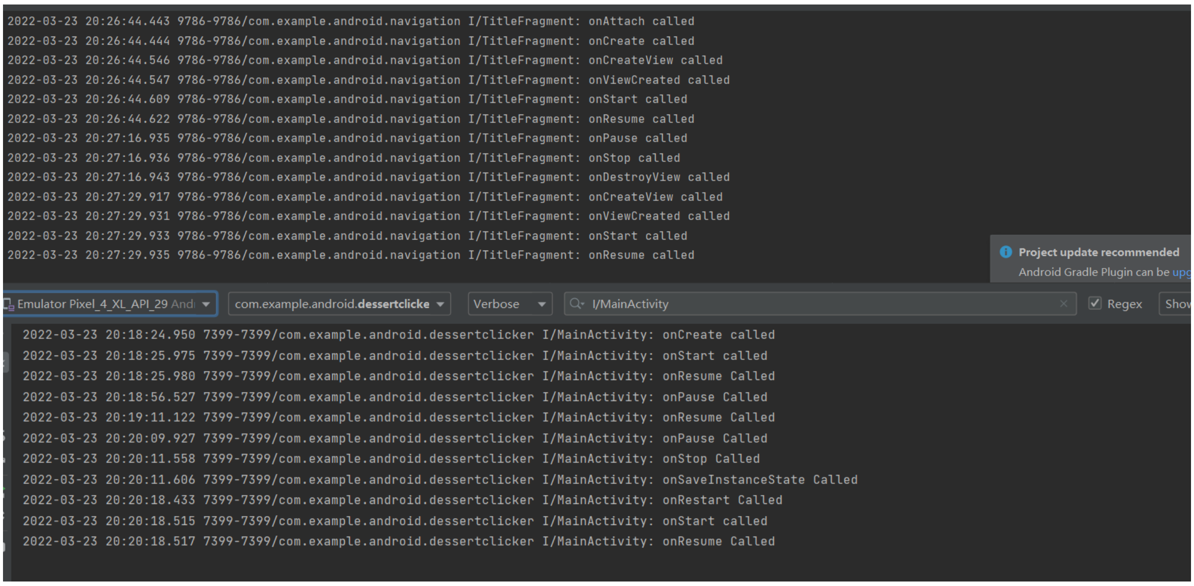
2 项活动 • 1 个测验

生命周期和日志记录

Codelab

复杂的生命周期情形

Codelab

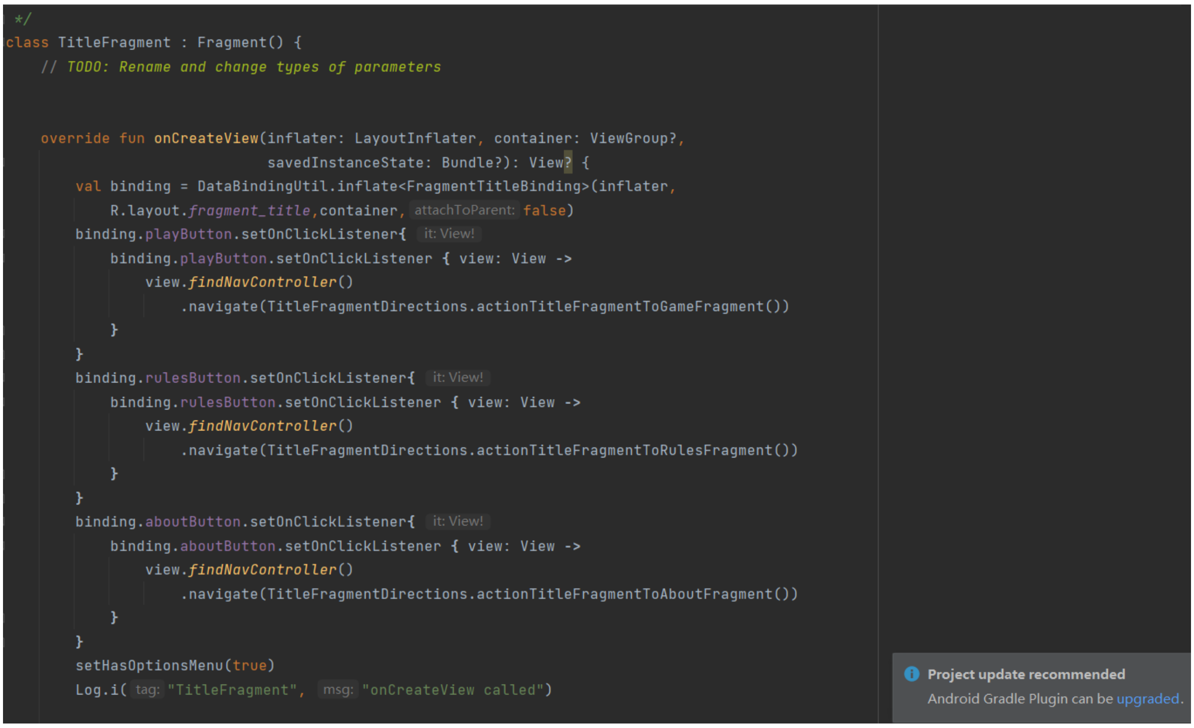

```
2022-03-23 20:26:44.443 9786-9786/com.example.android.navigation I/TitleFragment: onAttach called
2022-03-23 20:26:44.444 9786-9786/com.example.android.navigation I/TitleFragment: onCreate called
2022-03-23 20:26:44.546 9786-9786/com.example.android.navigation I/TitleFragment: onCreateView called
2022-03-23 20:26:44.547 9786-9786/com.example.android.navigation I/TitleFragment: onViewCreated called
2022-03-23 20:26:44.609 9786-9786/com.example.android.navigation I/TitleFragment: onStart called
2022-03-23 20:26:44.622 9786-9786/com.example.android.navigation I/TitleFragment: onResume called
2022-03-23 20:27:16.935 9786-9786/com.example.android.navigation I/TitleFragment: onPause called
2022-03-23 20:27:16.936 9786-9786/com.example.android.navigation I/TitleFragment: onStop called
2022-03-23 20:27:16.943 9786-9786/com.example.android.navigation I/TitleFragment: onDestroyView called
2022-03-23 20:27:29.917 9786-9786/com.example.android.navigation I/TitleFragment: onCreateView called
2022-03-23 20:27:29.931 9786-9786/com.example.android.navigation I/TitleFragment: onViewCreated called
2022-03-23 20:27:29.933 9786-9786/com.example.android.navigation I/TitleFragment: onStart called
2022-03-23 20:27:29.935 9786-9786/com.example.android.navigation I/TitleFragment: onResume called
```

Project update recommended

Android Gradle Plugin can be upg

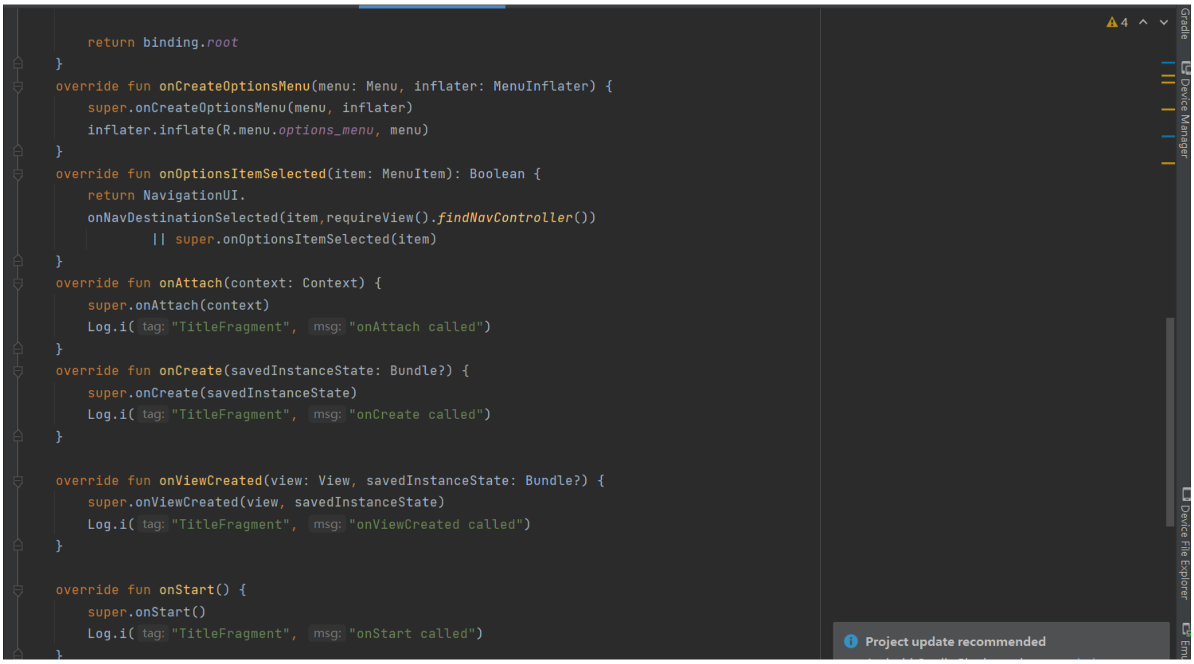
Emulator Pixel_4_XL_API_29 Andr
com.example.android.dessertclicke
Verbose
I/MainActivity
☑ Regex
Show

```
2022-03-23 20:18:24.950 7399-7399/com.example.android.dessertclicker I/MainActivity: onCreate called
2022-03-23 20:18:25.975 7399-7399/com.example.android.dessertclicker I/MainActivity: onStart called
2022-03-23 20:18:25.980 7399-7399/com.example.android.dessertclicker I/MainActivity: onResume Called
2022-03-23 20:18:56.527 7399-7399/com.example.android.dessertclicker I/MainActivity: onPause Called
2022-03-23 20:19:11.122 7399-7399/com.example.android.dessertclicker I/MainActivity: onResume Called
2022-03-23 20:20:09.927 7399-7399/com.example.android.dessertclicker I/MainActivity: onPause Called
2022-03-23 20:20:11.558 7399-7399/com.example.android.dessertclicker I/MainActivity: onStop Called
2022-03-23 20:20:11.606 7399-7399/com.example.android.dessertclicker I/MainActivity: onSaveInstanceState Called
2022-03-23 20:20:18.433 7399-7399/com.example.android.dessertclicker I/MainActivity: onRestart Called
2022-03-23 20:20:18.515 7399-7399/com.example.android.dessertclicker I/MainActivity: onStart called
2022-03-23 20:20:18.517 7399-7399/com.example.android.dessertclicker I/MainActivity: onResume Called
```

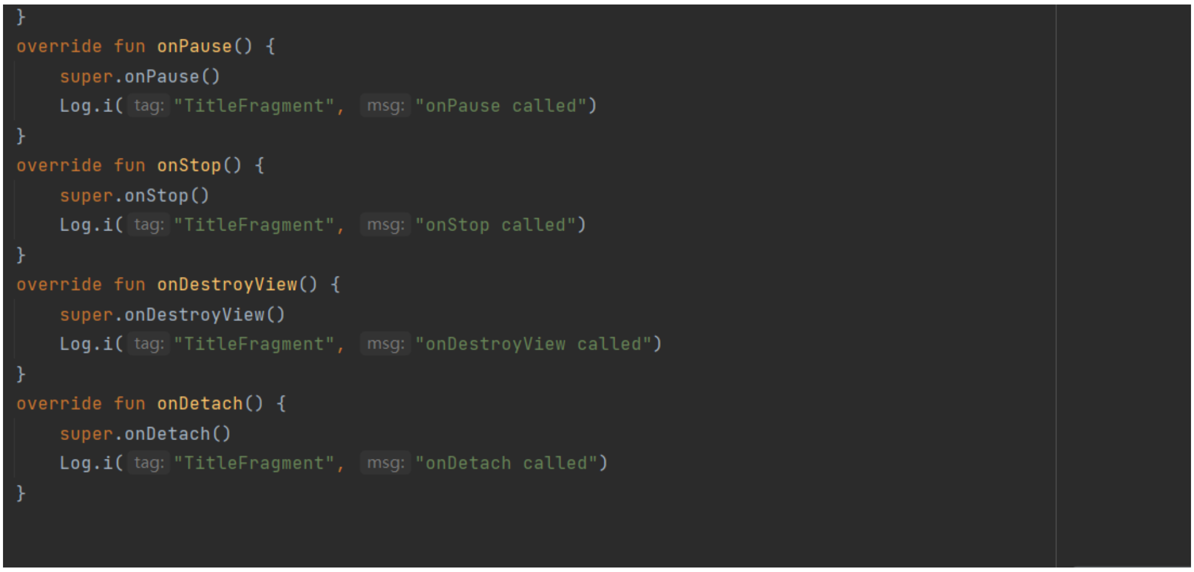

```
 */
class TitleFragment : Fragment() {
    // TODO: Rename and change types of parameters


    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        val binding = DataBindingUtil.inflate<FragmentTitleBinding>(inflater,
            R.layout.fragment_title,container, attachToParent: false)
        binding.playButton.setOnClickListener{ it: View!
            binding.playButton.setOnClickListener { view: View ->
                view.findNavController()
                    .navigate(TitleFragmentDirections.actionTitleFragmentToGameFragment())
            }
        }
        binding.rulesButton.setOnClickListener{ it: View!
            binding.rulesButton.setOnClickListener { view: View ->
                view.findNavController()
                    .navigate(TitleFragmentDirections.actionTitleFragmentToRulesFragment())
            }
        }
        binding.aboutButton.setOnClickListener{ it: View!
            binding.aboutButton.setOnClickListener { view: View ->
                view.findNavController()
                    .navigate(TitleFragmentDirections.actionTitleFragmentToAboutFragment())
            }
        }
        setHasOptionsMenu(true)
        Log.i( tag: "TitleFragment", msg: "onCreateView called")
```

Project update recommended

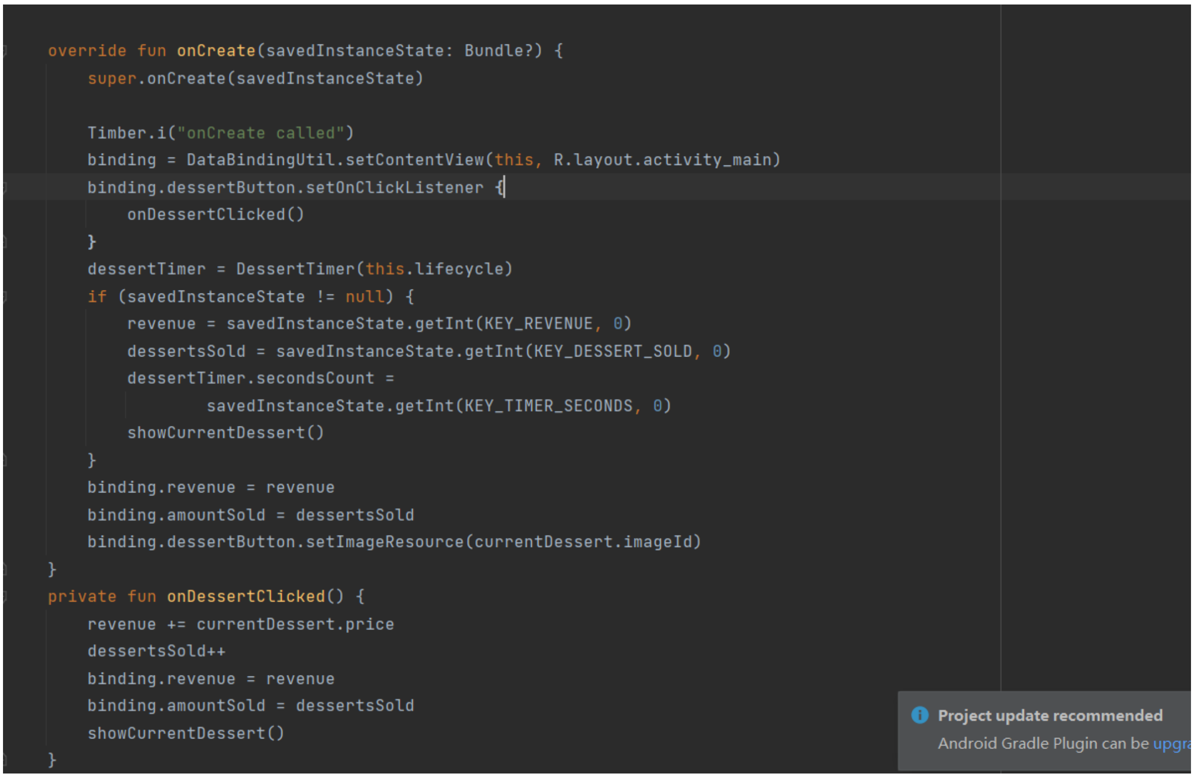
Android Gradle Plugin can be upgraded.



```
        return binding.root
    }
    override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
        super.onCreateOptionsMenu(menu, inflater)
        inflater.inflate(R.menu.options_menu, menu)
    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return NavigationUI.
        onNavDestinationSelected(item,requireView().findNavController())
                || super.onOptionsItemSelected(item)
    }
    override fun onAttach(context: Context) {
        super.onAttach(context)
        Log.i( tag: "TitleFragment",  msg: "onAttach called")
    }
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        Log.i( tag: "TitleFragment",  msg: "onCreate called")
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        Log.i( tag: "TitleFragment",  msg: "onViewCreated called")
    }

    override fun onStart() {
        super.onStart()
        Log.i( tag: "TitleFragment",  msg: "onStart called")
    }
```

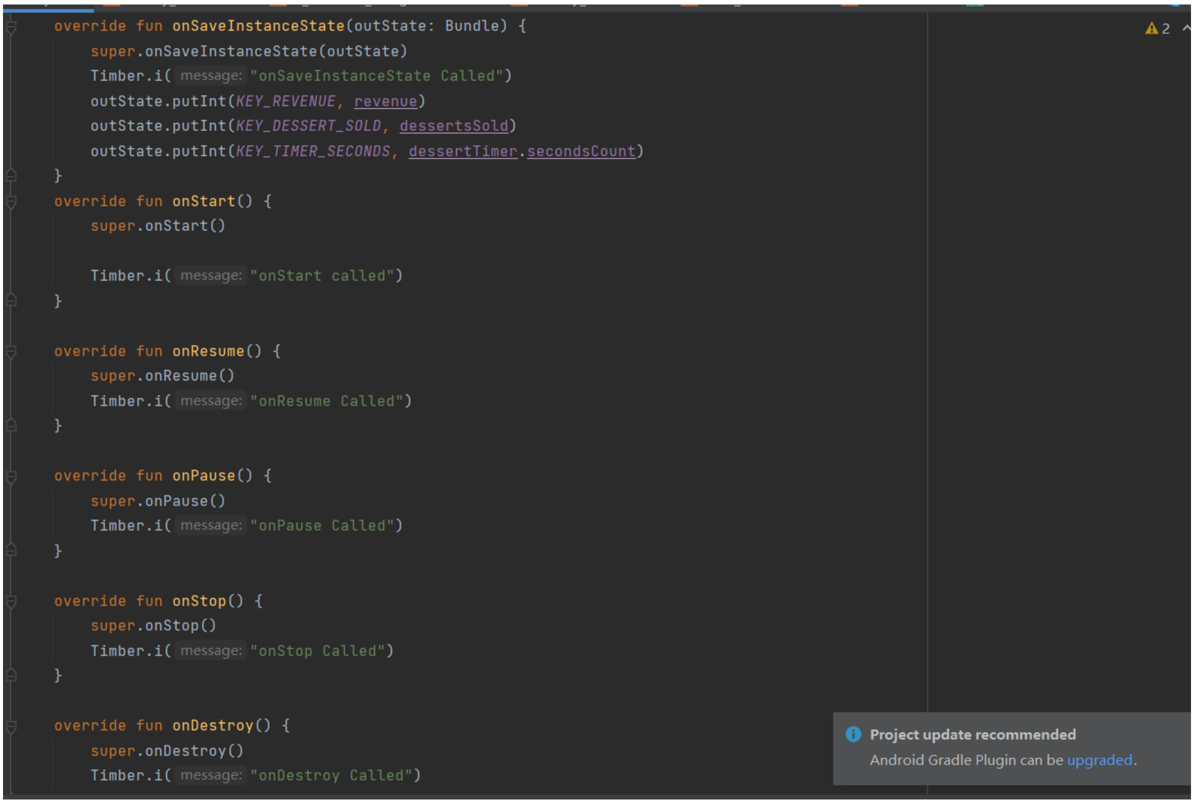


```
}
override fun onPause() {
    super.onPause()
    Log.i( tag: "TitleFragment",  msg: "onPause called")
}
override fun onStop() {
    super.onStop()
    Log.i( tag: "TitleFragment",  msg: "onStop called")
}
override fun onDestroyView() {
    super.onDestroyView()
    Log.i( tag: "TitleFragment",  msg: "onDestroyView called")
}
override fun onDetach() {
    super.onDetach()
    Log.i( tag: "TitleFragment",  msg: "onDetach called")
}
```



```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        Timber.i("onCreate called")
        binding = DataBindingUtil.setContentView(this, R.layout.activity_main)
        binding.dessertButton.setOnClickListener {
            onDessertClicked()
        }
        dessertTimer = DessertTimer(this.lifecycle)
        if (savedInstanceState != null) {
            revenue = savedInstanceState.getInt(KEY_REVENUE, 0)
            dessertsSold = savedInstanceState.getInt(KEY_DESSERT_SOLD, 0)
            dessertTimer.secondsCount =
                    savedInstanceState.getInt(KEY_TIMER_SECONDS, 0)
            showCurrentDessert()
        }
        binding.revenue = revenue
        binding.amountSold = dessertsSold
        binding.dessertButton.setImageResource(currentDessert.imageId)
    }
    private fun onDessertClicked() {
        revenue += currentDessert.price
        dessertsSold++
        binding.revenue = revenue
        binding.amountSold = dessertsSold
        showCurrentDessert()
    }
```

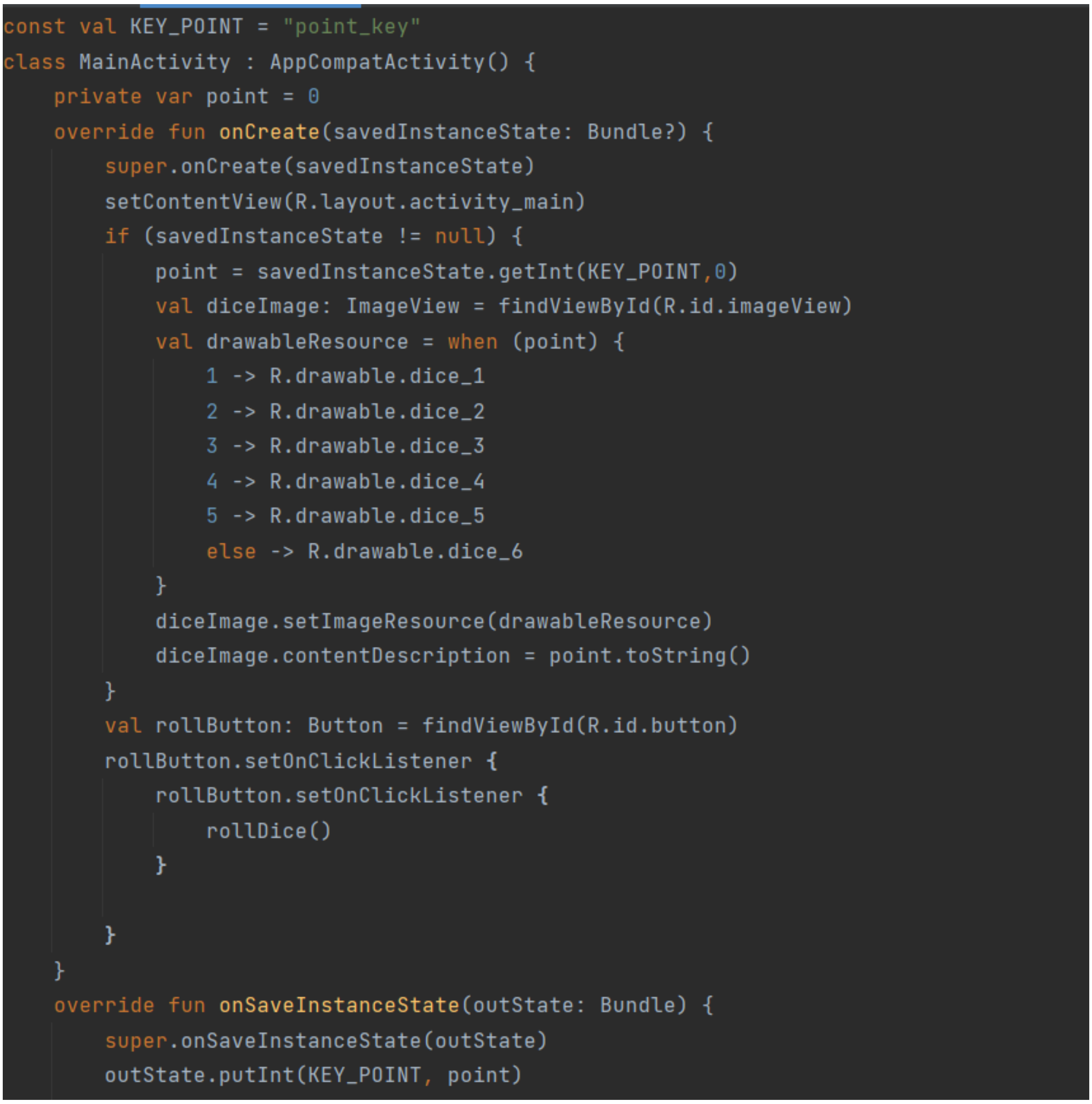
Project update recommended
Android Gradle Plugin can be upgra



```
override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    Timber.i( message: "onSaveInstanceState Called")
    outState.putInt(KEY_REVENUE, revenue)
    outState.putInt(KEY_DESSERT_SOLD, dessertsSold)
    outState.putInt(KEY_TIMER_SECONDS, dessertTimer.secondsCount)
}
override fun onStart() {
    super.onStart()

    Timber.i( message: "onStart called")
}

override fun onResume() {
    super.onResume()
    Timber.i( message: "onResume Called")
}

override fun onPause() {
    super.onPause()
    Timber.i( message: "onPause Called")
}

override fun onStop() {
    super.onStop()
    Timber.i( message: "onStop Called")
}

override fun onDestroy() {
    super.onDestroy()
    Timber.i( message: "onDestroy Called")
```

Project update recommended

Android Gradle Plugin can be upgraded.



```
const val KEY_POINT = "point_key"
class MainActivity : AppCompatActivity() {
    private var point = 0
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        if (savedInstanceState != null) {
            point = savedInstanceState.getInt(KEY_POINT,0)
            val diceImage: ImageView = findViewById(R.id.imageView)
            val drawableResource = when (point) {
                1 -> R.drawable.dice_1
                2 -> R.drawable.dice_2
                3 -> R.drawable.dice_3
                4 -> R.drawable.dice_4
                5 -> R.drawable.dice_5
                else -> R.drawable.dice_6
            }
            diceImage.setImageResource(drawableResource)
            diceImage.contentDescription = point.toString()
        }
        val rollButton: Button = findViewById(R.id.button)
        rollButton.setOnClickListener {
            rollButton.setOnClickListener {
                rollDice()
            }

        }
    }
    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        outState.putInt(KEY_POINT, point)
```