ST10444988

*IMAD5112*

***purpose***

*A Tamagotchi app is a virtual pet simulation available for smartphones and tablets, pro*viding users with an interactive experience reminiscent of the iconic Tamagotchi toy. These apps recreate the essence of caring for a digital pet, allowing users to feed, play with, and nurture their virtual companions through various activities. The virtual pets within the app require regular attention to ensure their well-being, much like their physical counterparts. Users must monitor their pet's hunger, happiness, and health levels, responding promptly to their needs to prevent them from becoming unhappy or even perishing.

These apps often offer a wide range of customization options, allowing users to choose their preferred pet species, customize their appearance, and even decorate their virtual environment. Some Tamagotchi apps also incorporate additional features such as mini-games, achievements, and social interactions with other virtual pet owners.

Tamagotchi apps serve as both nostalgic reminders of the original Tamagotchi craze from the 1990s and convenient modern-day alternatives for those seeking a virtual pet experience on their mobile devices. They provide users with a sense of companionship and responsibility while offering a fun and engaging way to pass the time.

```kotlin

class Tamagotchi(var name: String) {

private var hungerLevel: Int = 0

private var happinessLevel: Int = 0

fun feed() {

hungerLevel -= 10

println("$name has been fed. Hunger level decreased.")

if (hungerLevel < 0) hungerLevel = 0

}

fun play() {

happinessLevel += 10

println("$name is happy! Happiness level increased.")

if (happinessLevel > 100) happinessLevel = 100

}

fun getStatus(): String {

return "$name's current status:\nHunger Level: $hungerLevel\nHappiness Level: $happinessLevel"

}

}

```

```kotlin

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import kotlinx.android.synthetic.main.activity\_main.\*

import java.util.\*

class MainActivity : AppCompatActivity() {

private lateinit var pet: Tamagotchi

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val scanner = Scanner(System.`in`)

println("Welcome to Android Tamagotchi!")

print("Enter your Tamagotchi's name: ")

val petName = scanner.nextLine()

pet = Tamagotchi(petName)

btnFeed.setOnClickListener {

pet.feed()

updateStatus()

}

btnPlay.setOnClickListener {

pet.play()

updateStatus()

}

btnStatus.setOnClickListener {

updateStatus()

}

btnExit.setOnClickListener {

println("Exiting...")

finish()

}

}

private fun updateStatus() {

val status = pet.getStatus()

txtStatus.text = status

}

}

```

```xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="16dp"

tools:context=".MainActivity">

<Button

android:id="@+id/btnFeed"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Feed"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="16dp"/>

<Button

android:id="@+id/btnPlay"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Play"

android:layout\_below="@id/btnFeed"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="16dp"/>

<Button

android:id="@+id/btnStatus"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Check Status"

android:layout\_below="@id/btnPlay"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="16dp"/>

<TextView

android:id="@+id/txtStatus"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text=""

android:layout\_below="@id/btnStatus"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="16dp"/>

<Button

android:id="@+id/btnExit"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Exit"

android:layout\_below="@id/txtStatus"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="16dp"/>

</RelativeLayout>

```

```kotlin

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import kotlinx.android.synthetic.main.activity\_main.\*

import java.util.\*

class MainActivity : AppCompatActivity() {

private lateinit var pet: Tamagotchi

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val scanner = Scanner(System.`in`)

println("Welcome to Android Tamagotchi!")

print("Enter your Tamagotchi's name: ")

val petName = scanner.nextLine()

pet = Tamagotchi(petName)

btnFeed.setOnClickListener {

pet.feed()

updateStatus()

}

btnClean.setOnClickListener {

pet.clean()

updateStatus()

}

btnPlay.setOnClickListener {

pet.play()

updateStatus()

}

btnStatus.setOnClickListener {

updateStatus()

}

btnExit.setOnClickListener {

println("Exiting...")

finish()

}

}

private fun updateStatus() {

val status = pet.getStatus()

txtStatus.text = status

}

}

```

I've also added a `clean()` method to the `Tamagotchi` class. Here's the updated `Tamagotchi.kt` file:

```kotlin

class Tamagotchi(var name: String) {

private var hungerLevel: Int = 0

private var happinessLevel: Int = 0

private var cleanlinessLevel: Int = 100

fun feed() {

hungerLevel -= 10

println("$name has been fed. Hunger level decreased.")

if (hungerLevel < 0) hungerLevel = 0

}

fun play() {

happinessLevel += 10

println("$name is happy! Happiness level increased.")

if (happinessLevel > 100) happinessLevel = 100

}

fun clean() {

cleanlinessLevel = 100

println("$name is clean now!")

}

fun getStatus(): String {

return "$name's current status:\nHunger Level: $hungerLevel\nHappiness Level: $happinessLevel\nCleanliness Level: $cleanlinessLevel"

}

}

```