Testarea aplicației MoneyMap cu FastBot2

1 Introducere

1.1 Context

În prezent, piața este saturată de aplicații concepute pentru a răspunde diverselor nevoi ale utilizatorilor. Deoarece aproape fiecare aplicație este construită în jurul unei necesități specifice, experiența utilizatorului în timpul navigării și utilizării acesteia devine esențială. Testarea interfeței unei aplicații este un proces complex și de durată, însă indispensabil pentru a asigura o funcționare fără erori, prevenind bug-uri sau crash-uri care ar putea afecta utilizatorii.

1.2 FastBot2

Pentru a accelera procesul de testare, ByteDance a introdus FastBot2, un model automat de testare pentru aplicațiile Android. Acesta se bazează pe reutilizarea datelor din testările anterioare pentru a îmbunătăți acoperirea funcționalităților testate în timp. FastBot utilizează un model probabilistic pentru a ghida testarea, luând în considerare probabilitatea ca un eveniment să ajungă într-o anumită activitate, precum și un algoritm de reinforcement learning bazat pe un tabel Q, care stochează probabilitatea ca un hyper-event să ajungă într-o activitate neexplorată până în acel moment.

Inițial, FastBot primește fișierul APK al aplicației împreună cu rapoartele anterioare de crash-uri. Procesul de testare are două faze principale. În prima fază, APK-ul este decompilat, iar modelul probabilistic este populat cu datele existente, dacă acestea sunt disponibile. A doua fază constă în explorarea ghidată a paginilor aplicației: FastBot identifică hyper-event-urile, selectează cele care contribuie la extinderea acoperirii testării și actualizează datele colectate.

FastBot2 a demonstrat performanțe superioare altor instrumente de testare, precum Monkey, Ape și Stoat. De la implementarea sa în procesul de testare al ByteDance, a reușit să detecteze aproximativ 50,8

Am utilizat FastBot2 pentru a testa aplicația MoneyMap, un expense tracker cu o interfață intuitivă și ușor de utilizat, destinat utilizatorilor care doresc să își monitorizeze bugetul și cheltuielile în mod regulat.

2 Principalele funcționalități FastBot2

2.1 Modelul probabilistic

Pentru a stoca eficient rulările anterioare ale FastBot-ului, se utilizează un model probabilistic pentru testarea model-based. Acest model înregistrează probabilitatea calculată pentru toate execuțiile anterioare, indicând șansa ca un anumit hyper-event să conducă la o anumită activitate, sub forma unei funcții de tranziție δ.

Un hyper-event este generat pe baza unui UI widget și este definit prin patru proprietăți: activitatea asociată, textul widget-ului, ID-ul resursei și tipul acțiunii (de exemplu, click sau long press). Astfel, evenimentele care au aceleași proprietăți sunt grupate sub un singur hyper-event pentru a îmbunătăți acuratețea și scalabilitatea modelului.

De exemplu, pentru widget-ul „expense item," FastBot creează un singur hyper-event, deoarece toate instanțele acestuia au același ID de resursă și același layout. Această abordare elimină redundanța și optimizează procesul de testare.

Modelul probabilistic $M = (\epsilon, A, \delta)$ stochează:

- ϵ: Mulțimea hyper-event-urilor
- A: Mulțimea activităților
- δ: Funcția de tranziție $E \to P(A \times [0, 1])$ ce mapeaza fiecare hyper-event la o actiune cu o anumita probabilitate, unde P este funcție de putere (powerset) care generează toate submulțimile posibile de perechi (acțiune, probabilitate)

Calcularea acestor probabilitati se face pe baza formulei :

$$P(e, A_i) = \frac{N(e, A_i)}{N(e)} \tag{1}$$

unde $N(e, A_i)$ este numarul de selectii ale evenimentului e ce au condus la actiunea A_i, iar $N(e)$ este totalul execuțiilor lui e.

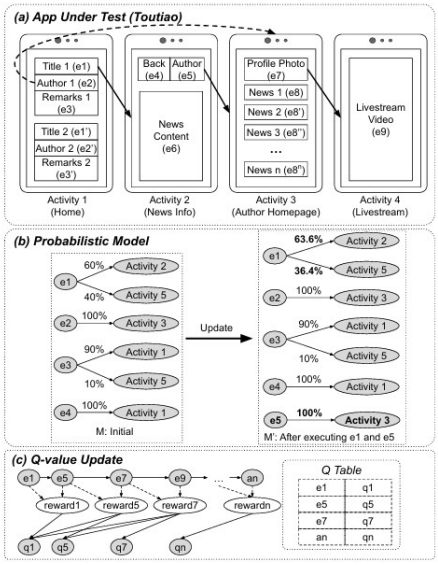

Figure 2: Activity Transition Example from Toutiao App

2.2 Explorarea ghidată a UI

Eficiența FastBot-ului se datorează faptului că utilizează datele din rulările anterioare pentru a selecta următorul eveniment de testare. Această selecție se bazează pe două metode combinate: selecția model-based și selecția learning-based.

Selecția model-based funcționează în două moduri: **model-expansion** și **model-exploitation**.

- În modul **model-expansion**, FastBot alege aleatoriu un hyper-event care nu a fost testat anterior, facilitând astfel explorarea și testarea unor funcționalități noi.
- FastBot intra in modul **model-exploitation**, atunci când toate evenimentele au fost deja incluse în model. În această etapă, selectează evenimentele cu o probabilitate mai mare de a acoperi acțiuni care nu au fost încă testate în execuția curentă.

Probabilitatea de acoperire a acțiunilor prin selectarea evenimentului e_i este definită prin formula:

$$E(e_i) = \sum_{A \notin A_t} P(e_i, A), \quad 0 \leq i \leq |E_c|$$

Pentru alegerea finală a evenimentului, FastBot aplică o variantă adaptată a funcției softmax, unde α este un parametru care controlează gradul de randomizare a selecției.

$$P_M(e_i) = \frac{\exp\left(\frac{E(e_i)}{\alpha}\right)}{\sum_{e_i \in E_c} \exp\left(\frac{E(e_i)}{\alpha}\right)} \tag{2}$$

2.3 Selectia learning-based

Modelul probabilistic are capacitatea de a lua decizii pas cu pas. Pentru a ghida testarea pe mai mulți pași, a fost introdus un agent de *reinforcement learning* care îmbunătățește calitatea testării prin utilizarea valorilor din tabelul Q, indicând evenimentele cu cea mai mare acoperire.

Valorile Q sunt actualizate la fiecare pas folosind formula:

$$Q(e_t) \leftarrow Q(e_t) + \alpha(G_{t,t+n} - Q(e_t))$$

unde la vechiul scor se adaugă diferența dintre recompensa $G_{t,t+n}$ și scorul curent, înmulțită cu rata de învățare α a agentului.

Recompensa este calculată prin adăugarea recompensei imediate pentru utilitatea acțiunii la pasul curent și a recompenselor viitoare estimate pe baza valorilor Q. Acestea sunt ponderate cu un factor de discount γ pentru a reduce influența recompenselor mai îndepărtate.

FastBot2 utilizează o politică *softmax* $P_Q(e_i)$ pentru a alege *hyper-event*-urile. Probabilitatea depinde de scorurile $Q(e_i)$, care reflectă utilitatea anterioară, și de parametrul de explorare β. Acest mecanism asigură un echilibru între testarea acțiunilor eficiente și descoperirea de noi stări.

$$P_Q(e_i) = \frac{\exp\left(\frac{Q(e_i)}{\beta}\right)}{\sum_{e_j \in E_c} \exp\left(\frac{Q(e_j)}{\beta}\right)}$$

3 Testarea aplicației MoneyMap cu FastBot2

3.1 Descrierea aplicației

MoneyMap este o aplicație simplă și intuitivă pentru gestionarea finanțelor personale, care îi ajută pe utilizatori să își monitorizeze cheltuielile, să planifice bugete și să înțeleagă obiceiurile lor financiare. Cu o interfață minimalistă și vizualizări clare, aplicația oferă:

- **Urmărirea cheltuielilor**: Înregistrează tranzacții cu dată și categorie, afișând detalii zilnice/lunare.
- **Bugete inteligente**: Setează limite lunare pentru fiecare categorie, cu bare de progres care arată cât ai cheltuit față de buget.
- **Categorii personalizate**: Sistem colorat și editabil pentru organizare flexibilă.

- **Statistici utile**: Calendare vizuale și grafice care arată distribuția cheltuielilor.

3.2 Configurarea FastBot2

Am configurat mediul de lucru prin instalarea *Android Command Line Tools* și setarea variabilelor de mediu necesare în WSL (*Windows Subsystem for Linux*). Am instalat componentele SDK, inclusiv:

- **platform-tools**
- **build-tools**
- **CMake**
- **NDK**

pentru a asigura compatibilitatea. Apoi, am compilat bibliotecile native și fișierele JAR ale *FastBot2* folosind scripturile dedicate.

Am pregătit un emulator Android și am utilizat **ADB** pentru a transfera fișierele necesare pe dispozitiv, inclusiv bibliotecile native (`libfastbot.so`) și modulele *FastBot2* (`monkeyq.jar`). Testarea propriu-zisă a fost lansată prin comenzi **ADB**, cu parametri precum:

- Durata de execuție: **60 de minute**
- Intervalul dintre acțiuni: **300 ms**

Rezultatele testelor au fost salvate în fișiere de log pentru analiză, identificând potențiale probleme de stabilitate și acoperire a interfeței.

3.3 Rezultatele testării

Log-urile de testare relevă trei categorii principale de probleme:

1. **Crash-uri în** `MainActivity` datorate unui delay de **126ms** pe *UI thread*, unde sistemul a forțat închiderea activității și a afișat dialogul de crash.

```
[Fastbot]*** WARNING *** 03-01 21:02:41.670  1670  3993 W ActivityManager:   Force finishing activity com.example.moneymap/.MainActivity
[Fastbot]*** WARNING *** 03-01 21:02:41.694  1670  1685 I ActivityManager: Showing crash dialog for package com.example.moneymap u0
[Fastbot]*** WARNING *** 03-01 21:02:41.853  1670  1685 W Looper  : Dispatch took 126ms on android.ui, h=Handler (android.view.Choreographer$FrameHandler) {e9565e2}
cb=android.view.Choreographer$FrameDisplayEventReceiver@9736d73 msg=0
[Fastbot]*** WARNING *** 03-01 21:02:42.172  1670  1684 W ActivityManager: Activity pause timeout for ActivityRecord{d39287c u0 com.example.moneymap/.MainActivity t6 f}
[Fastbot]*** WARNING *** 03-01 21:02:42.477  2712  2928 E ActivityThread: Failed to find provider info for com.google.android.apps.gsa.testing.ui.audio.recorded
[Fastbot]*** WARNING *** 03-01 21:02:42.742  1670  1684 I ActivityManager: Killing 3858:com.google.android.setupwizard/u0a24 (adj 906): empty #17
[Fastbot]*** WARNING *** 03-01 21:02:43.295  1670  1773 E TaskPersister: File error accessing recents directory (directory doesn't exist?).
[Fastbot]*** WARNING *** 03-01 21:02:44.561  1670  3993 I ActivityManager: Killing 4431:com.example.moneymap/u0a80 (adj 900): crash
```

2. **Excepții** `NullPointerException` în două scenarii diferite:
    - Accesarea unui `PackageManager` `null`.
    - Setarea resursei de fundal a unui buton `null` în `AddCategoryActivity.java:77`.

```
[Fastbot]*** ERROR *** // CRASH: com.example.moneymap (pid 15346) (elapsed nanos: 21116058664780)
[Fastbot]*** ERROR *** // Version:
[Fastbot]*** ERROR *** // Short Msg: java.lang.NullPointerException
[Fastbot]*** ERROR *** // Long Msg: java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.Button.setBackgroundResource(int)' on a null object reference
[Fastbot]*** ERROR *** // java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.Button.setBackgroundResource(int)' on a null object reference
[Fastbot]//     at com.example.moneymap.AddCategoryActivity.lambda$onCreate$0$com-example-moneymap-AddCategoryActivity(AddCategoryActivity.java:77)
[Fastbot]//     at com.example.moneymap.AddCategoryActivity$$ExternalSyntheticLambda0.onClick(D8$$SyntheticClass:0)
[Fastbot]//     at android.view.View.performClick(View.java:6294)
[Fastbot]//     at android.view.View$PerformClick.run(View.java:24770)
[Fastbot]//     at android.os.Handler.handleCallback(Handler.java:790)
[Fastbot]//     at android.os.Handler.dispatchMessage(Handler.java:99)
[Fastbot]//     at android.os.Looper.loop(Looper.java:164)
[Fastbot]//     at android.app.ActivityThread.main(ActivityThread.java:6494)
[Fastbot]//     at java.lang.reflect.Method.invoke(Native Method)
[Fastbot]//     at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:438)
[Fastbot]//     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:807)
[Fastbot]//
```

3. **Erori repetitive de tip null reference** în activități diferite, sugerând o problemă la gestionarea *lifecycle-ului* componentelor sau de inițializare a *view*-urilor.

Aceste erori indică fie lipsa verificărilor de `null`, fie o inițializare incorectă a elementelor UI în anumite stări ale aplicației.

Sesiunea de testare a generat un total de **28.981** de evenimente (*clicuri, swipe-uri, input*), acoperind **50%** din activități (**8 din 16** ecrane).

FastBot2 a explorat cu succes fluxurile principale (**bugete, cheltuieli, categorii**), dar a omis ecrane secundare (**înregistrare, selecție avatar**) și unele activități Firebase.

De asemenea, a fost identificată o eroare de metodă lipsă (`setActivityController`), sugerând posibile incompatibilități.

```
[Fastbot][2025-03-29 05:37:53.082] Events injected: 28981
[Fastbot][2025-03-29 05:37:53.083] // Monkey is over!
[Fastbot][2025-03-29 05:37:53.090] Total app activities:
[Fastbot][2025-03-29 05:37:53.090]    1 com.example.moneymap.BudgetActivity
[Fastbot][2025-03-29 05:37:53.090]    2 com.example.moneymap.AddExpenseActivity
[Fastbot][2025-03-29 05:37:53.090]    3 com.google.android.gms.common.api.GoogleApiActivity
[Fastbot][2025-03-29 05:37:53.090]    4 com.example.moneymap.ExpenseActivity
[Fastbot][2025-03-29 05:37:53.091]    5 com.example.moneymap.ProfileActivity
[Fastbot][2025-03-29 05:37:53.091]    6 com.example.moneymap.ExpenseDetailsActivity
[Fastbot][2025-03-29 05:37:53.091]    7 com.google.android.gms.auth.api.signin.internal.SignInHubActivity
[Fastbot][2025-03-29 05:37:53.091]    8 com.example.moneymap.AddCategoryActivity
[Fastbot][2025-03-29 05:37:53.091]    9 com.google.firebase.auth.internal.GenericIdpActivity
[Fastbot][2025-03-29 05:37:53.091]   10 com.google.firebase.auth.internal.RecaptchaActivity
[Fastbot][2025-03-29 05:37:53.091]   11 com.example.moneymap.AvatarSelectionActivity
[Fastbot][2025-03-29 05:37:53.091]   12 com.example.moneymap.RegisterActivity
[Fastbot][2025-03-29 05:37:53.091]   13 com.example.moneymap.LoginActivity
[Fastbot][2025-03-29 05:37:53.091]   14 com.example.moneymap.HomeActivity
[Fastbot][2025-03-29 05:37:53.091]   15 com.example.moneymap.MainActivity
[Fastbot][2025-03-29 05:37:53.092]   16 com.example.moneymap.CategoryActivity
[Fastbot][2025-03-29 05:37:53.093] Explored app activities:
[Fastbot][2025-03-29 05:37:53.093]    1 com.example.moneymap.AddCategoryActivity
[Fastbot][2025-03-29 05:37:53.093]    2 com.example.moneymap.AddExpenseActivity
[Fastbot][2025-03-29 05:37:53.093]    3 com.example.moneymap.BudgetActivity
[Fastbot][2025-03-29 05:37:53.093]    4 com.example.moneymap.CategoryActivity
[Fastbot][2025-03-29 05:37:53.094]    5 com.example.moneymap.ExpenseActivity
[Fastbot][2025-03-29 05:37:53.094]    6 com.example.moneymap.HomeActivity
[Fastbot][2025-03-29 05:37:53.094]    7 com.example.moneymap.LoginActivity
[Fastbot][2025-03-29 05:37:53.095]    8 com.example.moneymap.MainActivity
[Fastbot][2025-03-29 05:37:53.095] Activity of Coverage: 50.0%
[Fastbot]*** ERROR *** [Fastbot][2025-03-29 05:37:53.100] findMethod() error, NoSuchMethodException happened, there is no such method: setActivityController// App appears 4 crash, 0 anr, monkey using seed:
1743483958320
```

4 Concluzii și soluții

Pentru remedierea problemelor detectate de FastBot, este necesară:

- Revizuirea inițializării *view*-urilor în `AddCategoryActivity.java:77`.
- Gestionarea cazurilor de `null` în `MainActivity`.
- Extinderea testării pe ecranele neacoperite.

În concluzie, integrarea FastBot2 în procesul de testare a aplicațiilor Android demonstrează avantajele utilizării unui model probabilistic combinat cu reinforcement learning pentru optimizarea explorării interfeței. Rezultatele testării asupra aplicației MoneyMap au evidențiat capacitatea FastBot2 de a detecta crash-uri și probleme legate de gestionarea ciclului de viață al componentelor UI, asigurând o acoperire extinsă a scenariilor de utilizare.