

PRIESKUMNÉ TESTOVANIE V POČÍTAČOVÝCH HRÁCH

*„Otvor oči dokorán a rozhliadni sa navôkol, to čo
práve hľadáš, nemusí byť ďaleko.“*

Márius Šajgalík

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
xsajgalik[zavináč]stuba[.]sk

Abstrakt. Počítačové hry predstavujú špecifický typ softvérového systému. V dôsledku ich osobitých vlastností, dynamického rozvoja a pokroku, ktorý v súčasnosti podstupujú, narastá aj ich komplexnosť a otázka zabezpečenia kvality sa dostáva čím ďalej, tým viac do popredia. Od kvality hry ako softvérového produktu nakoniec závisí aj jej celkové postavenie na trhu herného priemyslu. V tejto eseji predstavujem prieskumné testovanie ako alternatívny prístup k štandardným metódam testovania a opisujem výhody jeho použitia pri testovaní počítačových hier. Opisujem dôležitosť určovania priorít zadaných požiadaviek, od ktorých sa neskôr odvíjajú aj priority chýb. Práve prieskumné testovanie je prístup, ktorý zohľadňuje tieto priority a v niektorých prípadoch je preto efektívnejšie a výhodnejšie ako klasické skriptované testovanie.

Kľúčové slová: testovanie hier, prieskumné testovanie, priorita chyby, priorita požiadavky, testovanie založené na požiadavkách

Úvod

Počítačové hry poznáme prakticky už od začiatku éry osobných počítačov. Ich účelom je zabaviť používateľa, aby sa mohol aj pri práci s počítačom odreagovať. V súčasnosti sa čoraz viac dostáva do popredia práve softvér a spolu s ním sa vyvíjajú aj počítačové hry, ktoré patria prakticky medzi tie najnáročnejšie aplikácie. Dôvodom je, že väčšina hier sa snaží simulovať reálny svet. To znamená, že aj v takomto hernom svete sa neustále niečo

odohráva, mení a tieto zmeny musí počítač v reálnom čase vykresľovať hráčovi, aby bol schopný sledovať dianie v hre a včas reagovať na príslušné udalosti.

Vzhľadom na komplexnosť súčasných počítačových hier, ktoré už tiež majú za sebou značný pokrok, otázka zabezpečenia kvality sa stáva čoraz významnejšou a dostáva sa stále viac a viac do popredia. Testovanie je dôležité v každom softvérovom projekte a pri hrách sa stáva čím ďalej, tým viac netriviálnou záležitosťou. Celková kvalita hry, ktorá nakoniec do značnej miery závisí od testovania, zohráva veľmi dôležitú rolu aj pre jej postavenie v hernom biznise.

Testovanie počítačových hier

Keďže počítačové hry sa snažia simulovať určitý život v hernom svete, testovanie nemožno chápať len ako overenie správnosti výstupu pre nejaké metódy a funkcie. Vo všeobecnosti sa hra skladá z viacerých podsystémov, ktoré zabezpečujú vykonávanie hernej logiky. Tieto sú medzi sebou rôzne prepojené v závislosti od konkrétnej hry. Je veľmi náročné zabezpečiť kvalitu a bezchybnosť pri prakticky nespočetnom množstve možných stavov, do ktorých sa môže hra dostať.

Vzhľadom na špecifický charakter počítačovej hry ako softvérového systému, testovanie nie je triviálna záležitosť a chyby v hrách je často nesmierne ťažké nájsť. Na rozdiel od väčšiny softvérových systémov, hry sa spoliehajú nielen na náhodné čísla, ale taktiež menia obrovské množstvo údajov nesmierne rýchlo, zdanlivo nepredvídateľným spôsobom. Obtiažnosť nájdania chyby v hre spočíva v jednoduchšej skutočnosti, že v hrách sa vykonáva tak veľa kódu a tak rýchlo, že ľahko sa môže ukázať, že chyba pochádza z ktoréhokoľvek podsystému, ktorý manipuluje so stavom hry.

Otázkou je, ako by sme mali hru testovať, aby sme dokázali zabezpečiť jej kvalitu? V súčasnosti existujú viaceré techniky a metódy testovania, ktoré sa používajú na zabezpečenie kvality softvéru. Tieto vieme rozdeliť na základe viacerých kritérií do niekoľkých kategórií. To však neznamená, že každý softvérový systém vieme bez problémov testovať. Takéto prakticky stále narastajúce množstvo testovacích techník a metód spôsobuje značné problémy pri rozhodovaní ako testovať daný softvér. Nevieme totiž jednoznačne povedať, ktorý spôsob je optimálny, lebo každý spôsob má svoje výhody a nevýhody. Preto sa väčšinou snažíme určiť kľúčové vlastnosti, resp. požiadavky na produkt, tie, ktoré majú najväčší význam a dôležitosť. Na základe toho, ako aj na základe charakteru samotného produktu, potom vieme určiť, ktoré spôsoby testovania majú najviac výhod vzhľadom na tieto požiadavky.

Skriptované testovanie

Jedným z bežne používaných prístupov je skriptované testovanie. Ak poznáme množinu možných vstupov a dokážeme overiť správnosť príslušných výstupov, môžeme napísať aj testovací skript. Podľa Kanera [1] testovací skript určuje operácie, ktoré sa v teste vykonajú, očakávané výsledky a porovnania, ktoré sa majú vykonať. Keďže testovací skript má presne napísanú a teda definovanú postupnosť krokov, ktoré sa majú vykonať, nazývame takéto testovanie skriptované.

Problém spočíva v tom, že pri testovaní hier nám takéto testy nestačia. Dôvodov je viacero a niektoré už boli aj spomenuté. V hre sa často simuluje reálny svet, vykonáva sa značné množstvo kódu každý herný cyklus a v dôsledku nielen náhodných čísel a jedinečnosti používateľského vstupu, ale aj neúplnosti a viacznačnosti špecifikovaných požiadaviek, nevieme takéto testovacie skripty napísať. Preto je výhodné využiť alternatívne prístupy k testovaniu hier. Jeden z takýchto prístupov k testovaniu a vytváraniu testov predstavuje prieskumné testovanie.

Prieskumné testovanie

Čo to teda prieskumné testovanie je? Existuje viacero definícií, James Bach [2] definuje prieskumné testovanie ako: „...súbežné učenie sa, navrhovanie testov a ich vykonávanie zároveň.“ Tinkham a Kaner [3] uvádzajú trochu odlišnú definíciu: „...akékoľvek testovanie, pri ktorom tester aktívne riadi návrh testov počas behu samotných týchto testov a využíva informácie získané pri testovaní na vytvorenie nových a lepších testov.“ Podľa Kanera, Bacha a Pettichorda [4] ďalej prieskum znamená: „...zmysluplné bádanie: navigácia v priestore so všeobecným cieľom, no bez predpísanej cesty. Prieskum zahŕňa neustále učenie sa a experimentovanie.“ Je to teda vcelku intuitívny spôsob k testovaniu v softvérovom projekte. Je to vytváranie testov bez ich predchádzajúcej špecifikácie, pričom sa zároveň aj vykonávajú. Testy nie sú skriptované. Prieskumné testovanie nie je iba nejaká technika testovania, je to prístup k testovaniu softvéru. Nie je tu oddelená fáza návrhu testov pred fázou ich vykonania. Testy sa vytvárajú počas testovania, počas prieskumu softvéru. Preto sa teda takéto testovanie nazýva prieskumné.

Prieskumné testovanie má viacero potenciálnych výhod ako napríklad efektívnosť, schopnosť využiť kreativitu testera a taktiež rýchla spätná väzba. Tieto výhody teda neplynú len zo schopnosti vedieť použiť prieskumné testovanie. Podľa SWEBOK [5] (Software Engineering Body of Knowledge) efektívnosť prieskumného testovania závisí od vedomostí a znalostí testera softvéru. Craig a Jaskiel [6] navyše zdôrazňujú fakt, že v prieskumnom testovaní môžu vzniknúť celé produktívne oblasti testovania. Pri skriptovanom testovaní sa naopak vytvára mnoho testov, ktoré vôbec nevyzerajú také užitočné počas ich vykonávania ako počas ich vytvárania.

Pri testovaní počítačových hier vidím veľký potenciál práve v tomto prístupe. Keď sa prejde z úrovne overovania správnosti jednotlivých metód na úroveň testovania samotnej hrateľnosti v rámci herného sveta, testovanie sa dá intuitívne chápať ako prieskumné. Samozrejme, že si možno dopredu určiť konkrétne scenáre testov, ktoré treba overiť, ako napríklad postava sa musí vedieť dostať do vnútra domčeka, alebo po stlačení vypínača sa musí rozsvietiť lampa. Keď sa však dá voľne pohybovať v hernom svete (a nielen vtedy), prieskumným testovaním vieme overiť, aké sú napríklad možné spôsoby splnenia postupnosti logických úloh, alebo ako je možné deformovať nejaký objekt, keď po ňom budeme udierať kladivom.

Ako ďalší dôvod, prečo považujem prieskumné testovanie za vhodné pri testovaní hier, je oddelenie hernej logiky od vizuálnej časti (aj keď nie úplné). Pri vývoji hry totiž modely jednotlivých objektov v hernom svete vytvára grafik. Tieto modely sú následne rozmiestnené v hernom svete, čo vykonáva návrhár hry zvyčajne pomocou herného editora. Pre úplnosť musí programátor zabezpečiť vykonávanie samotnej hernej logiky. V

dôsledku takéhoto rozdelenia úloh môžu veľmi ľahko vznikáť chyby, ktoré plynú z nevedomosti všetkých detailov niektorej z častí. Ako jedna z ďalších výhod prieskumného testovania je súbežné učenie sa. Keď sa tester neriadi predpísanými skriptami, aktívne sa učí o špecifikách testovaného a získava vedomosti o jednotlivých detailoch správania. Takto dokáže objaviť aj tie podstatné chyby, ktoré vznikli v dôsledku nevedomosti a rozdelenia vývoja. Zjavne sa objavajú len tie chyby, ktoré sú navonok viditeľné alebo významné. Z tohto vyplýva ďalšia skutočnosť a to je fakt, že niektoré chyby možno zanedbať. Pre každú chybu teda možno identifikovať jej závažnosť a na základe toho jej priradiť aj určitú prioritu.

Prioritizácia chýb

V manažmente kvality softvéru je teda dôležité vedieť správne priradiť priority chybám a rozličným aspektom zabezpečenia kvality. Ako príklad z bežného života v rámci vývoja počítačových hier uvádzam nasledujúcu príhodu [7]:

Vývojári Savage Empire použili skutočne nádherný spôsob opravenia chyby (tzv. „hack“). Používali vtedy v neskorých deväťdesiatych rokoch kompilátor Borland C 3.1, ktorého modul vždy pri ukončení kontroloval isté pamäťové miesto, aby zistil, či nejaký kus kódu náhodou neprepísal ten kúsok v pamäti, ktorý sa už nepoužíval. Ak zistil zmenu v umiestnení pamäte, vypísal chybu "Error: (null) pointer assignment." navrchu obrazovky. Tieto chyby bolo vtedy dosť náročné vedieť lokalizovať, keďže procesor šťastne a spokojne predpokladal, že programátor vie, čo robí. Programátori Savage Empire sa márne snažili chybu vystopovať až do posledných dní vývoja. Vedenie spoločnosti už chcelo zaradiť hru do výroby, keďže Vianoce už boli za dverami. Jeden z programátorov nakoniec "opravil" chybu nádherným spôsobom. Upravil vykonateľný súbor savage.exe v HEX editore a zmenil textový reťazec "Error: (null) pointer assignment." na iný s presne tou istou dĺžkou: "Thanks for playing Savage Empire."

Z tejto príhody plynie ponaučenie, že nie všetky chyby je nutné odstrániť, pokiaľ to nie je nutné. Niekedy je možné uprednostniť zakamuflovanie chyby pred zbytočným a náročným procesom hľadania príčiny a spôsobu, ako túto príčinu eliminovať. V uvedenej príhode bolo pre hru postačujúce, aby chyba nebola viditeľná. Pre vývojárov bolo prioritou, aby sa hra vypla a aby si hráč nevšimol, že nejaká chyba vznikla.

Teraz sa vrátíme na začiatok celého vývoja softvéru, kedy sa spisujú požiadavky na konečný softvérový produkt – hru. Podľa doterajších úvah môžeme vidieť, že nie je dôležité, aby bola hra bez chýb, ale aby spĺňala určité špecifikované požiadavky, tak ako hociký iný softvérový systém. Otázkou je, prečo by sme sa nemali primárne sústrediť na odstraňovanie chýb? Vo všeobecnosti totiž platí, že sa snažíme spraviť softvér bez chýb. Ako však ukazuje prax, v značnej časti prípadov to nie je také jednoduché a vždy sa nejaká tá chyba nájde. Keďže navyše ani odstraňovanie týchto chýb nemusí byť triviálna záležitosť, každej chybe sa priradí určitá priorita na základe jej závažnosti. Posledná vec, ktorú je potrebné si nakoniec uvedomiť, je ako identifikujeme závažnosť chyby? Musíme si teda dať do vzťahu chybu s požiadavkami a určiť mieru, do akej táto chyba narúša naplnenie daných požiadaviek. Od miery naplnenia požiadaviek potom závisí aj konečná celková kvalita výsledného produktu.

Postupne sme sa teda dopracovali k skutočnosti, že kvalita nemusí byť o tom, či je daný program bez chýb, či beží rýchlo, alebo ani raz nepadne. Podľa normy ISO 8402 je kvalita definovaná ako množina vlastností a charakteristík výrobku alebo služby, ktoré preukazujú jeho schopnosť splniť určené alebo odvodené potreby. Takže podľa uvedených odvodení, ktoré sme podložili touto definíciou, priority s cieľom zabezpečiť kvalitu sa odvodzujú od požiadaviek.

Prieskum a prioritizácia požiadaviek

Ak si stanovíme ako prioritu splniť danú požiadavku a tester má za úlohu vyhodnotiť kvalitu, t.j. mieru naplnenia tejto požiadavky, tak tester musí byť s touto požiadavkou veľmi dobre oboznámený. Uvažujme teraz situácie, v ktorých toto platí, teda vyžadujeme, aby sa splnila daná požiadavka. Pravdou je, že zadané požiadavky často nie sú postačujúce a teda ani jediné. V dôsledku neúplnosti a nejednoznačnosti požiadaviek totiž úloha testera nie je iba vyhodnotiť mieru naplnenia danej požiadavky, ale aj preskúmať význam a dôsledky tejto požiadavky. Takýmto spôsobom preto vznikajú ďalšie požiadavky, ktoré ďalej upresňujú a bližšie špecifikujú cieľový produkt. Vidíme teda, že testovať možno aj keď nie sú úplne špecifikované všetky požiadavky a vtedy je to dokonca užitočné [8].

Mali by sme sa zbaviť mýtu, že testovanie je iba nejaký robotický proces. Z vyššie opísaného prístupu k testovaniu môžeme vidieť, že testovanie nie je iba akési mechanické vyhodnotenie kvality a môže naopak predstavovať kreatívny proces. Taktiež je dôležitá spolupráca medzi testerami, vývojármi a v prípade hier aj grafikmi a návrhármi hry, ktorá určite v značnej miere ovplyvňuje takýto kreatívny proces testovania. Pri vyhodnocovaní kvality berie skúsený tester do úvahy aj svoju predstavu o nešpecifikovaných požiadavkách a využíva svoje pozorovania pri konzultáciách ku kvalite produktu s ostatnými členmi tímu. Skúsený tester sa vždy snaží nájsť medzery v špecifikovaných požiadavkách a snaží sa ich vyplniť tak, aby zbytočne nevznikali chyby v dôsledku potenciálnych nejednoznačností.

Takýto prístup k testovaniu nazývame testovanie založené na požiadavkách. Keď sa však zamyslíme trochu viac, môžeme si všimnúť istú mieru podobnosti s prieskumným testovaním. Aj v prípade uvedeného kreatívneho prístupu k testovaniu – testovaniu založenom na požiadavkách – tester skúma danú požiadavku, hľadá už spomínané medzery v špecifikácii a teda tiež robí akýsi prieskum. Taktiež platí, že je to „súbežné učenie sa, navrhovanie testov a ich vykonávanie zároveň“, čo je vyššie spomínaná definícia prieskumného testovania. Preto si myslím, že toto „kreatívne“ testovanie založené na požiadavkách možno zaradiť medzi prieskumné testovanie ako jeden z jeho špecifických prístupov.

Pri vývoji hier a ako sme zistili aj konkrétne pri testovaní v hrách, zohráva dôležitú rolu práve kreativita. Kreatívny prístup je teda dôležitý aj pri testovaní, nielen pri návrhu, pretože práve ten sa v hrách dokáže podstatne meniť. V hrách sa prakticky už od začiatku neustále udržiava nejaký prototyp, ktorý slúži na testovanie a formovanie samotného návrhu. Tento sa teda podrobuje neustálemu skúmaniu a jednoznačne tu dostáva priestor prieskumné testovanie.

Taktiež práve v hrách je veľmi bežné, že požiadavky na cieľový produkt sú dosť neurčité a niekedy dokonca ani nepoznáme spôsob, ako overiť jednoznačné splnenie danej požiadavky. Jedným z bežných príkladov je napríklad požiadavka, aby bola hra zábavná. Toto je príliš všeobecná požiadavka na to, aby sme vedeli jednoznačne určiť, či ju hra spĺňa, alebo nie. Mnohokrát sa ukáže až po vydaní hry, že hra nie je až taká zábavná ako sa to javilo pri jej vývoji a pred tým, než sa začala predávať. Veľa mladých a začínajúcich vývojárov hier si kladie za cieľ, aby mala hra dokonalú fyziku a aby umelá inteligencia postáv bola na úrovni človeka. Skúsení vývojári hier však vedia, že zábavnosť je pre hru jedna z najdôležitejších požiadaviek a je priam kritické zaoberať sa touto požiadavkou už od začiatku vývoja. Myslím si, že práve prieskumné testovanie má potenciál veľmi dobre identifikovať priority jednotlivých požiadaviek.

Záver

Prieskumné testovanie zohráva teda dôležitú rolu pri vývoji hier. Hry totiž predstavujú dosť špecifický typ softvéru, pre ktorý veľmi často nestačí len skriptované testovanie a iné bežne používané prístupy ako napríklad testovanie bielej skrinky. To však neznamená, že takéto testovanie pri hrách nemá význam a možno ho vynechať. Avšak napriek tomu, že aj hry majú nejaký prvotný návrh a špecifikované požiadavky, často podliehajú niekedy až výrazným zmenám a vždy sa nájdu nejaké nedostatky špecifikácie, nejasnosti, alebo viacznačnosti. Je dôležité, aby testovanie odhalilo a identifikovalo takéto nedostatky a predišlo sa tak potenciálnym problémom a konfliktom. Myslím si, že prieskumné testovanie predstavuje výborný prístup ako možno takéto nedostatky identifikovať. Vďaka nemu sa totiž preskúmajú zadané požiadavky, na základe čoho je možné vyplniť medzery špecifikácie v podobe neúplnosti a viacznačnosti. Takto sa preskúmajú aj detaily, ktoré sa neskôr môžu ukázať ako kritické. Z toho vyplýva, že všetko sa odvíja od neustáleho určovania priorít, čo prispieva k dosiahnutiu čo najvyššej kvality konečného produktu.

Použitá literatúra

1. Kaner, C.: *A Tutorial in Exploratory Testing*, QAI QUEST Conference, Chicago, April 2008.
2. J. Bach, *Exploratory Testing*, in *The Testing Practitioner*, Second ed., E. van Veenendaal Ed., Den Bosch: UTN Publishers, pp. 253-265, 2004.
3. A. Tinkham and C. Kaner, *Exploring Exploratory Testing*, 2003, Accessed 2010 12/1, <http://kaner.com/pdfs/ExploringExploratoryTesting.pdf>
4. C. Kaner, J. Bach and B. Pettichord, *Lessons Learned in Software Testing*, John Wiley & Sons, Inc., New York, 2002.
5. IEEE, *Guide to the Software Engineering Body of Knowledge*, IEEE, Tech. Rep. IEEE - 2004 Version, 2004.
6. R.D. Craig and S.P. Jaskiel, *Systematic Software Testing*, Artech House Publishers, Boston, 2002.
7. M. McShaffry et al, *Game Coding Complete*, 3rd edition, Charles River Media, 2009.

8. Bach, J.: *Risk and Requirements-Based Testing*, IEEE, 1999.

Annotation

Exploratory testing in computer games

Computer games represent a specific type of software system. Due to their distinctive character, dynamic expand and progress they have made till now, their complexity grows as well and the quality assurance issue becomes more and more important. On the quality of a game as software product depends also its final position on the game industry market. In this essay I present the exploratory testing as an alternative approach to the standard testing methods and I describe the benefits of its use in computer game testing. I describe the importance of defining the priorities of stated requirements, which are the basis for defining the bug priorities. The exploratory testing is just that approach, which takes all of these priorities into account and in some cases is therefore more efficient and convenient than classic scripted testing.