

Medplatformni razvoj grafično intenzivnih aplikacij - osnutek diplomske naloge

Anže Pečar (63060257)

February 19, 2013

1 Motivacija

Mobilne naprave dandanes postajajo vedno bolj vsakdanje. Pametni telefoni imajo v sebi več procesorske moči, kot namizni računalniki izpred parih let. Poleg procesorske moči praviloma vsebujejo tudi grafične procesne enote, ki jih razvijalci lahko izkoristijo za razvoj grafično intenzivnih aplikacij. Poleg telefonov pa so se pojavili tudi tablični računalniki, ki imajo praviloma še boljše karakteristike kot pametni telefoni.

Med posameznimi proizvajalci telefonov in tablic obstajajo velike razlike v razvojnem okolju. Vsak izmed mobilnih operacijskih sistemov uporablja drug programski jezik za razvoj nativnih aplikacij pa tudi pri izvedbi različnih knjižnic (npr. OpenGL ES) se porajajo razlike. Razvoj grafične aplikacije, ki bi jo napisali enkrat in bi delovala povsod, je tako skorajda nemogoč.

Problem postane še težji, če želimo poleg vseh mobilnih naprav podpreti še namizne računalnike. Sedaj imamo poleg različnih programskih jezikov in knjižnic, še različne možnosti interakcije z uporabnikom - vnos z dotikom na mobilnih napravah in vnos z miško in tipkovnico na namiznih računalnikih.

2 Možnosti

2.1 2D canvas[1]

Programski jezik Javascript je edina skupna točka pametnih telefonov, tablic in namiznih računalnikov. Vse te naprave imajo namreč vgrajen tak ali drugačen spletni brskalnik. Spletni brskalniki nudijo razvijalcem element *canvas*, ki razvijalcu omogoči dostop do funkcij za risanje na zaslon. *Canvas* je dobro podprt na vseh platformah, vendar za izrisovanje uporablja samo centralno procesno enoto, ne izrablja pa moči, ki jo nudi grafična procesna enota.

2.2 WebGL[2]

WebGL je kontekst *canvasa*, ki ima direkten dostop do grafične kartice. WebGL torej popravlja glavno pomankljivost *2D canvasa*, vendar za ceno podprtosti po platformah. Določeni spletni

brskalniki namreč WebGLa sploh ne podpirajo (Internet Explorer), podpore še nimajo omogočene (Safari na iOS) ali pa je podpora še v beta stanju (Chrome na Androidu).

WebGL uporablja API knjižnico OpenGL ES 2.0, ki je sicer dobro podprta na večini mobilnih naprav. Glavni problem WebGLa je varnost, saj je skorajda nemogoče narediti peskovnik (sandbox) za spletne strani, ki imajo direkten dostop do grafične kartice.

2.3 libgdx[3]

Libgdx je medplatformno ogrodje za razvoj grafičnih aplikacij, s poudarkom na računalniških igrah. Razvijalec svojo aplikacijo napiše v programskem jeziku Java, ogrodje pa potem poskrbi za izvoz napisane aplikacije v različne platforme. Podprte platforme so Windows, Linux in OSX ter Android in iOS, s pomočjo Google Web Toolkit (GWT) knjižnice pa je možen izvoz tudi za spletne brskalnike (WebGL).

2.4 V8-gl[4] in LycheeJS[5]

V8-gl omogoča razvoj grafičnih aplikacij za namizne računalnike v jeziku Javascript. Z malo abstrakcije je tako možno spletno aplikacijo pretvoriti v namizno aplikacijo. Delo lahko poenostavi knjižnica LycheeJS, cilj njenega avtorja pa je razširiti V8-gl tudi na Android in iOS.

2.5 Unity[6]

Unity je razvojno okolje, ki nam omogoča razvoj grafično intenzivnih aplikacij in izvoz za različne platforme. Za Windows, OSX in Linux je izvoz brezplačen, za iOS in Android pa je avtorjem potrebno plačati.

References

- [1] Canvas Element W3C - <http://www.w3.org/wiki/HTML/Elements/canvas>
- [2] WebGL - Khronos Group - <http://www.khronos.org/webgl/>
- [3] libGDX - <http://libgdx.badlogicgames.com/>
- [4] V8-gl izvorna koda - <https://github.com/philogb/v8-gl>
- [5] LycheeJS - <http://martens.ms/lycheeJS/>
- [6] Unity - <http://unity3d.com/>