Szoftver vizualizáció

# A felhasznált szoftver

Az általunk választott szoftver a *Gource* volt, amely egy *OpenGL*-alapú 3D vizualizációs eszköz. A *CodeMetropolis*-al ellentétben, amely a szoftver különböző metrikái alapján próbálja vizualizálni a szoftver minőségét, a *Gource* egy másik szempontból közelíti meg, ugyanis a szoftverhez használt verziókövető rendszer log fájlját használja a vizualizációhoz. Ezért inkább a szoftver fejlesztésének (fejlődésének) folyamatába ad betekintést. Így időrendben megtekinthető, hogy milyen változtatások történtek a szoftverben (pl.: osztály hozzáadása, változtatása, törlése), illetve, hogy az adott módosítás ki által történt.

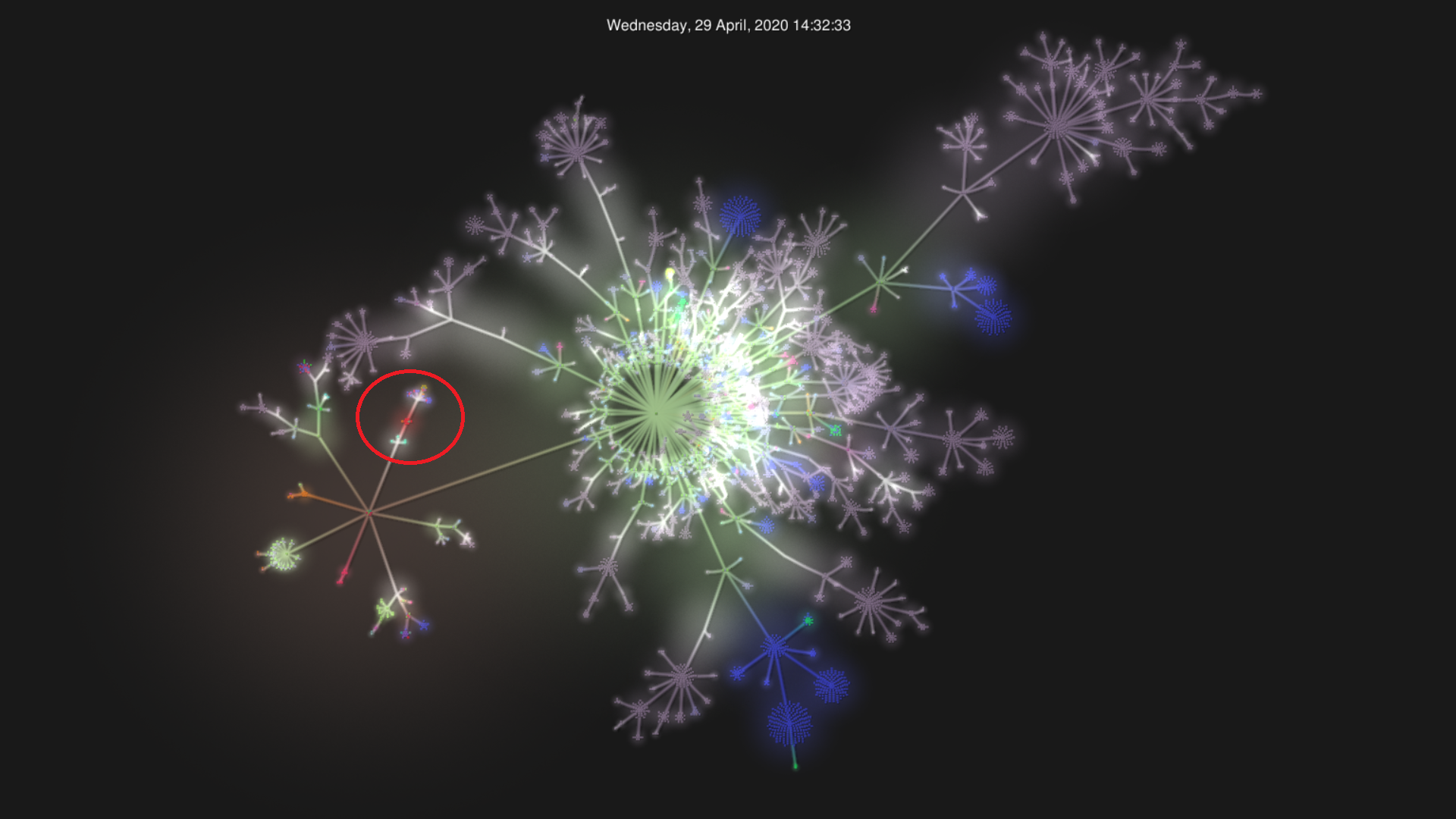
# Metafora

A *Gource* a gráf, pontosabban fagráf metaforát használja. A projekt gyökér könyvtára van a középpontban, onnan különböző ágak (élek) nőnek, amelyek a könyvtárakat jelképezik. A levelek jelentik a különböző a fájlokat, melyeknek a színe függ a kiterjesztésüktől, azaz két ugyanolyan kiterjesztésű fájlnak a színe meg fog egyezni. Egy fájl hozzáadása egy új levél megjelenését, törlése pedig annak eltünését jelenti. Változtatás esetén egy ideiglenes él jelenik meg a fájl és a változtatást végrehajtó fejlesztő között. A fejlesztőt egy bábu reprezentálja, felette a nevével. Ez kicserélhető képekre is a megfelelő kapcsolók használatával.

# Vizualizáció elvégzése

A mérföldkő alatt két vizualizáció készült. Az első a szoftver teljes életciklusát ábrázolja. Ennek elkészítéséhez a következő parancsot használtuk:

*gource -f -1440x900 --stop-position 1.0 --highlight-all-users --hide dirnames,filenames,mouse,progress --seconds-per-day 0.05 --output-framerate 60 --output-ppm-stream output.ppm ffmpeg -y -r 60 -f image2pipe -vcodec ppm -i output.ppm -vcodec libx264 -r 60 -qscale 0 full.mp4*



1. *ábra DBeaver ábrázolása Gource segítségével*

A teljes projekt végső gráfját az 1. ábra szemlélteti. A rózsaszín levelek a *java*, még a világoszöld levelek az *xml* kiterjesztésű fájlok. Ezen kívül a sötétkék levelek a *png* kiterjesztésű képeket ábrázolják. A projekt felépítését egyértelműen e három kiterjesztéssel rendelkező állományok dominálják. A pirossal bekarikázott rész, már az általuk, a dokumentálás érdekében létrehozott

állományokat jelöli. Ezzel ellentétben a *Fejlesztés és tesztelés* mérföldkő alatt érintett forrásfájlokat már nagyobb kihívás megtalálni a végső gráfon, ugyanis ezek többsége már korábban létre lett hozva. Ezért relevánsnak tartottuk az átalunk végzett módosításokat külön megvizsgálni.

A második gráf a csapat által végzett módosításokat tartalmazza, a félév kezdetétől számítva. Ennek ábrázolásához a következő parancsot használtuk:

*gource -f -1920x1080 --start-date "2020-03-02" --seconds-per-day 0.8 --highlight-all-users --output-framerate 60 --output-ppm-stream short.ppm ffmpeg -y -r 60 -f image2pipe -vcodec ppm -i short.ppm -vcodec libx264 -r 60 -qscale 0 short.mp4*

# A felhasznált szoftver hasznossága

Mivel *Gource* leginkább a fejlesztés folyamatába ad betekintést, így a végső gráf előálltával, a szoftver minőségéről nem kapunk visszajelzést. Például nincsenek megkülönböztetve a tesztosztályok a hagyományos osztályoktól. Bár meg tudjuk jeleníteni az egyes leveleknek megfelelő fájlok neveit, viszont ez rövid időn belül átláthatatlan gráfot eredményez. Egy másik hátrányos tényező, hogy az egyes állományok és fejlesztők csak színek alapján vannak megkülönböztetve (már amennyiben a fejlesztők nem rendelkeznek egyedi képekkel), így a színtévesztő és a színvak emberek nem feltétlen képesek megérteni a metaforát.