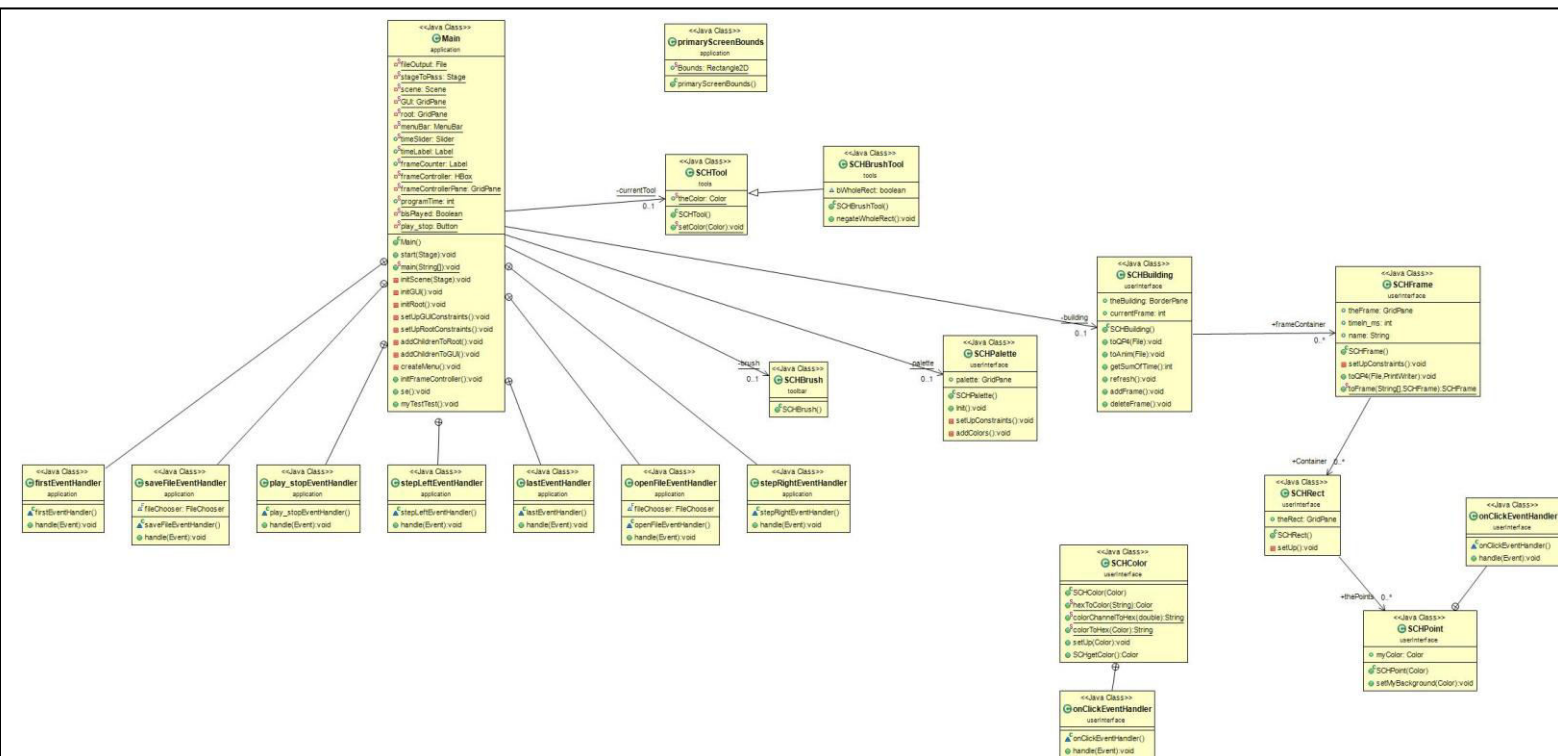


Házi feladatként egy Schönherz Mátrix animeditor program tervezését és kivitelezését választottam. A program tulajdonképpen egy pixelenként szerkeszthető rajztábla, a Schönherz kollégium ablakaira levetítve 2x2 pixeles négyzetenként. Több ilyen elmentett rajztáblából (frame-ből) tevődik össze egy animáció. A program az elkészült frame-eket egymás után sorbatéve egy külön fájlba menti, amit később levetítésre alkalmas formátumra alakítani, illetve importálni lehet a hivatalos <http://oriaskijelzo.hu/> oldalon is megtalálható animeditoral.

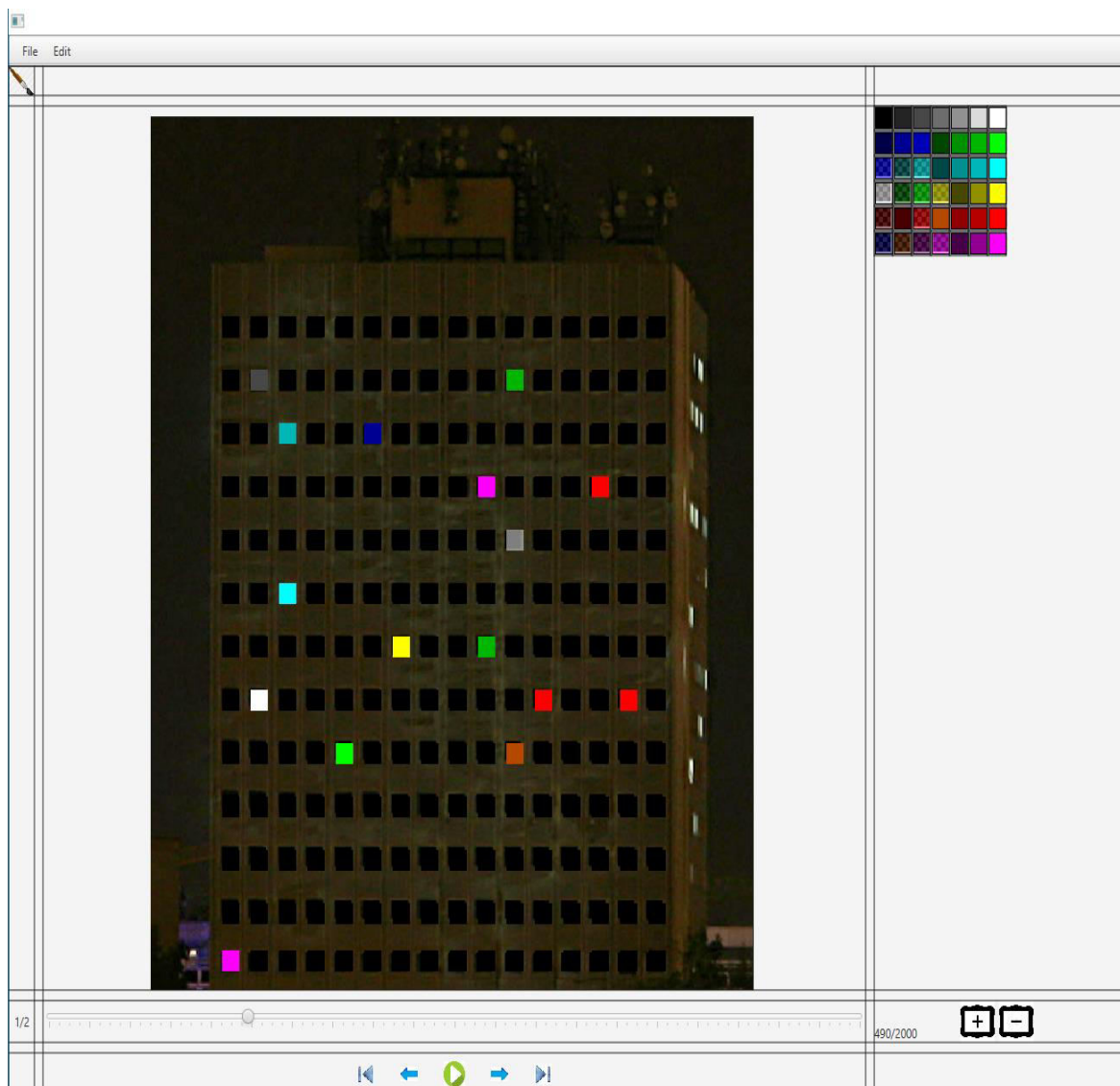
A megvalósításhoz használt osztályokat az alábbi osztálydiagram szemlélteti



Az osztályok rövid jellemzése:

- **Main:** Ez az osztály a program főosztálya, tartalmazza az alkalmazás futásához szükséges elemeket. A grafikus felhasználói felület összes eleme itt van példányosítva vagy eltárolva.
- **primaryScreenBounds:** tárolja a képernyő adatait, segítségével más felbontásban is használható a program
- **SCHTool:** egy alaposztály, mely lényegében tárol egy színt, amivel az adott eszköz dolgozik. Tőle öröklődnek a különböző eszközök
- **SCHBrushTool:** jelenleg az egyetlen elérhető eszköz, vele egy darab pixel színeztethető ki.
- **SCHBrush:** egy button, mely segítségével a jelenlegi eszköz SCHBrushTool-ra változik.
- **firstEventHandler:** extends EventHandler, ha az első frame-re jutás gombjára kattintunk hívódik meg.
- **saveFileEventHandler:** a mentéshez használt osztály.
- **play_stopEventHandler:** a lejátszás gombjához hozzáadott EventHandler
- **stepLeftEventHandler:** az egy frame-vel előre lépéshez hozzáadott EventHandler
- **lastEventHandler:** az utolsó frame-re ugrik a program.
- **openFileEventHandler:** fájlokat lehet megnyitni a segítségével
- **stepRightEventHandler:** egy frame-vel hátrébb lépéshez használt EventHandler
- **SCHPalette:** SCHColor objektumokat tartalmaz, lényegében egy gyűjtő a színváltó gombokhoz. Az alap animeditor-ban megtalálható összes színt tartalmazza
- **SCHColor:** extends Button, rákattintva színt vált a jelenlegi eszközünk.
- **SCHBuilding:** tartalmazza az épület képét háttérként, illetve tartalmaz egy darab adott Frame-t
- **SCHFrame:** tárolja 16x13-as mátrixként egy GridPane-n az 2x2-es ablakokat
- **SCHRect:** 2x2-es mátrixot tárol egy GridPane-n, ezek az ablakon lévő pixelek
- **SCHPoint:** extends button, az ablakon lévő egy pixel osztálya
- etc eventHandlererek

A program működés közben az alábbi képen látható a debug-segédvonalakkal, melyek CSS



segítségével vannak megjelenítve.

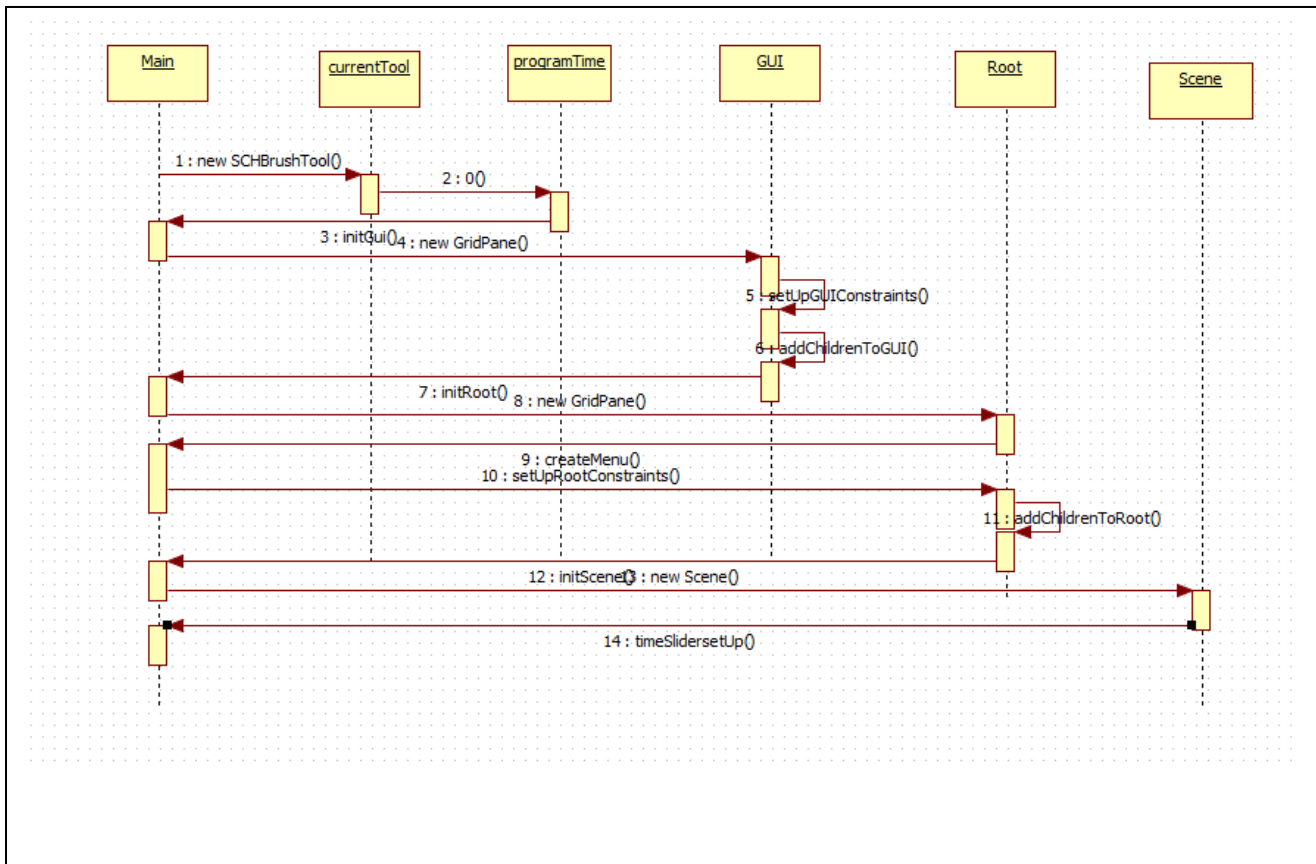
A program fontosabb adatai:

- `Main.fileOutput` : a fájl útvonalát tartalmazza, amiben dolgozunk
- `Main.stageToPass`: Stage típusú objektum, segítségével a `primaryStage` átadható egy függvénynek
- `Main.scene`: tulajdonképpen az ablakot tárolja
- `Main.GUI`: a felhasználói felület minden elemét(`building`,`tools`,`palette`,`Labels`,`etc...`) tartalmazó `GridPane`, menü nélkül.
- `Main.root`: a root `GridPane`, GUI + menü
- `Main.menuBar`: a menüt tartalmazza. Itt érhető el a mentés és megnyitás funkció
- `Main.timeSlider`: ezen a `Slider`-en állítható a program ideje. Dinamikusan változik a frame-ek számával.
- `Main.timeLabel`: a `Slider` mellett megtekinthető rajta a program jelenlegi ideje/a frame-k összideje
- `Main.frameCounter`: a `Slider` jobb oldalán található frame állapotjelző
- `Main.frameController`: a `Slider` alatt található Frame-manipuláló sávot tartalmazza
- `Main.FrameControllerPane`: Az előző `HBox`-ot tartalmazza egy `Pane`-n. Erre azért van szükség, hogy a megfelelő helyen jelenjen meg a dolog
- `Main.programTime`: a program jelenlegi idejét tartalmazza (hogyan áll az időszáv)
- `Main.bIsPlayed`: jelenleg nem használt változó (ld. `known-issues`)
- `Main.play_stop`: a `play` gomb tárolása

Fontosabb metódusok:

- `Main.initScene(Stage)`: inicializálja a scene változót.
- `Main.initGUI()`: inicializálja a GUI Pane-t, meghívja a függvényt, felállítja a Constrainteket, illetve hozzáadja a tagokat.
- `Main.initRoot()`: inicializálja a root Pane-t, meghívja a függvényt, felállítja a Constrainteket, illetve hozzáadja a tagokat.
- `Main.setUpGUIConstraints()`: felállítja a GUI-n a megfelelő határokat, hogy jó helyen jelenjen meg minden a GridPane-n
- `Main.setUpRootConstraints()`: felállítja a root-on a megfelelő határokat, hogy jó helyen jelenjen meg minden.
- `Main.addChildrenToRoot()`: itt adódnak hozzá a root-hoz a gyerekei
- `Main.addChildrenToGUI()`: itt adódnak hozzá a GUI-hoz a gyerekei
- `Main.createMenu()`: létrehozza a menüt
- `Main.initFrameController()`: inicializálja a frameController nevű Pane-t, és hozzáadja a gyerekeit.
- `SCHBuilding.toQP4(File)`: kimentti az adott animációt egy QP4 fájlba, a megfelelő formátumba
- `SCHBuilding.toAnim(File)`: beolvassa a QO4 formátumú fájlt, és átalakítja animációvá
- `SCHBuilding.getSumOfTime`: visszatér a frame-k összidejével
- `SCHBuilding.refresh()`:frissíti a jelenleg látható Frame-t
- `SCHBuilding.addFrame()`: üres fekete Frame-t told hozzá az animáció végéhez.
- `SCHFrame.toQP4(File,Printwriter)`: kiírja saját magát a fájlba.
- `SCHFrame.toFrame(String[], SCHFrame)`: A paraméterül kapott SCHFrame objektumot fel setup-olja a kapott String[] alapján, majd visszatér vele. A paraméterül kapott String[] kizárólag a színeket tartalmazza hexadecimális kódként, soronként.
- `SCHColor.hexToColor(String)`: a paraméterül kapott hexadecimális string-et vagy 0-t Color típusú objektummá alakítja
- `SCHColor.colorToHex()`: fordítva működik, mint az egyel feljebb lévő függvény

A Main-ban lefutó folyamatok szekvenciadiagramon:



known-issues:

- A stop gomb jelenleg nem működik, erre lesz később használva a `isPlayed` változó.
- A `Frame`-k neve nem megváltoztatható-
- A `Frame`-k ideje nem megváltoztatható, fix 1000.
- A program a követelmények szerint 10 darab metódusra tesztelve van JUnit4 segítségével, azonban a JavaFX használata miatt ezek nem futnak le. (illetve csak néhány). Mivel a program kézi bemenetet vár, ezt elég nehéz így tesztelni.