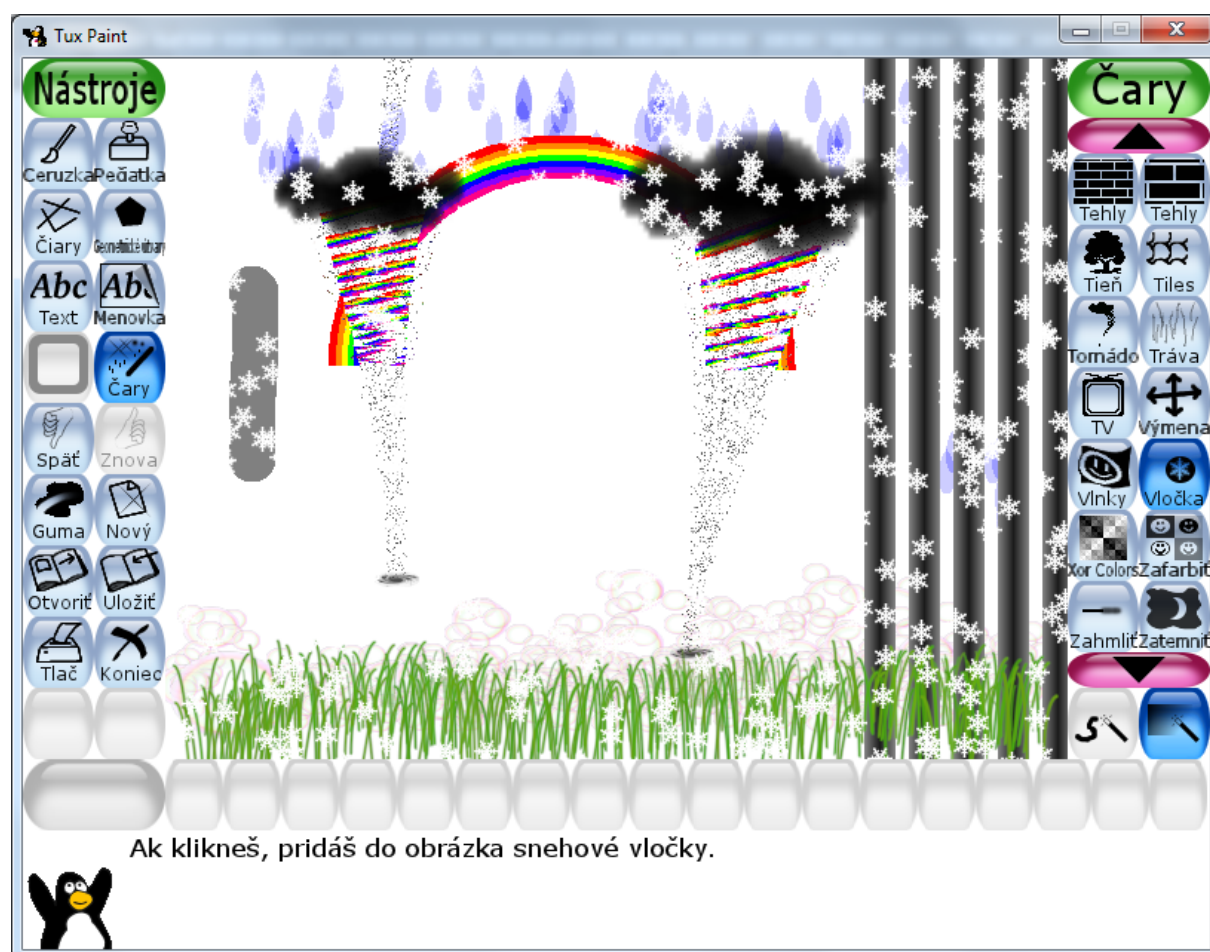


Teoretická časť

1 etapa

Softvéru na kreslenie a editovanie fotografií je v súčasnosti mnoho. Medzi nimi je aj grafický softvér určený predovšetkým pre deti. Tento softvér obsahuje základné kresliace nástroje, aké ponúkajú aj iné programy. Teda ceruzka, štetec, guma, kreslenie geometrických útvarov a podobne. Avšak grafický softvér pre deti sa musí vyznačovať predovšetkým jednoduchým a intuitívnym ovládaním a taktiež špeciálnymi nástrojmi pre deti. Špeciálne nástroje umožňujú napríklad kreslenie známych tvarov ako srdiečko, hviezdička, kvapka a podobne.

Jeden zo softvérov pre deti je Tux Paint. Tux Paint je voľne dostupná aplikácia, ktorá funguje na všetkých operačných systémoch a je možné si ju stiahnuť z internetu. Tento program umožňuje deťom načítanie vlastných obrázkov, kreslenie a upravovanie a následné uloženie obrázka. Na kreslenie a upravovanie majú k dispozícii mnoho nástrojov – ceruzka, pečiatka, kreslenie geometrických útvarov, čary. Každý z týchto nástrojov má mnoho nastaviteľných atribútov - farba, hrúbka, veľkosť. Čary sú špeciálnym nástrojom, ktorý umožňuje kreslenie mnohých tvarov a farieb – koľajnice, tehly, dúhu, srdiečka, hviezdičky, tráva. Tento softvér je špecifický aj tým, že pri kreslení prehráva rôzne detské zvuky – keď kreslíte kvapky, počujete dážď.



2 etapa: Teoretické zázemie

V aplikácii sme využili prvky objektovo orientovaného programovania jazyka JavaScript, ktorý pracuje iba s objektmi. JavaScript umožňuje využitie všetkých technologických aspektov objektovo orientovaného programovania, predovšetkým zapuzdrenie, dedičnosť a polymorfizmus. Tieto vlastnosti sme využili aj my - pri programovaní kresliacich nástrojov. Ako vidieť na diagrame nižšie, pre všetky tieto nástroje existuje abstraktná „trieda“ `AbstractTool`. Všetky kresliace nástroje následne využívajú dedičnosť a sú potomkami tejto triedy. Trieda `ToolManager` využíva polymorfizmus týchto objektov na ktoré volá jednotlivé udalosti bez znalosti aktuálne vybraného nástroja.

Nástrojom sme pridali vlastnosti hrúbka čiary, farba čiary a farba výplne. Na ovládanie hrúbky čiary sme zvolili interaktívny prvok slider (posuvník, bežec). Na výber farby sme zvolili prvok `ColorPicker`. Pri oboch prvkoch sme využili existujúce knižnice z nasledovných zdrojov: [slider](#), [ColorPicker](#).

3 etapa: Články zaoberajúce sa podobnou tematikou

3.1 Integrating Canvas into your Web App

url: <http://www.html5rocks.com/en/tutorials/canvas/integrating/>

Autor článku popisuje, ako sa používa HTML5 canvas v spojení s programovacím jazykom Javascript. Niektoré staršie prehliadače, napríklad IE6, nepodporujú HTML5 canvas, ale vieme zistiť, ktoré a upozorniť na to používateľa. Ďalej sa zameriava na to, ako inicializovať používanie canvasu v Javascripte, ako vykresliť .jpg obrázok, čiaru. Ukazuje exportovanie canvasu do .png obrázka pomocou metódy `toDataURL()` zavolanej na kontexte canvasu. Takisto sa dá obsah canvasu uložiť do *HTML5 localStoreage*-u, čo sa dá využiť pri použiteľnosti aplikácie - keď používateľ omylom zatvorí prehliadač, tak obrázok nakreslený v canvase nemusí byť stratený, keď sa využije táto funkcia a pôvodný obrázok sa po novom načítaní stránky opätovne vykreslí do canvasu. Obsah canvasu sa tiež dá uložiť na server ako binárny súbor pre opätovné použitie. Na konci sú uvedené grafické editory a iné pomôcky v HTML5 canvase:

<http://mihai.sucan.ro/paintweb/trunk/demos/demo1.html> (terba kliknúť na tlačidlo "Edit image!")

3.2 Canvas Inspection using Chrome DevTools

url: <http://www.html5rocks.com/en/tutorials/canvas/inspection/>

V tomto článku sme sa dozvedeli, že tak ako má funkciu debuggovať každé vyspelejšie IDE programacieho jazyka, rovnako môžeme krokovat' a "debugovať" HTML5 canvas pomocou debuggera integrovaného v prehliadačoch. Článok konkrétne spomína debuggovanie canvasu v prehliadači Chrome, no rovnaké funkcie má dnes už aj napr. Firefox, či Safari. Tieto nástroje na ladenie sa však musia zapnúť pred spustením, či kreslením do canvasu, zrejme z optimalizačných dôvodov. Tento nástroj je však vhodný pre aplikácie, kde sú animácie a prekresľovanie, keďže sa tu dajú zobrazovať jednotlivé keyframes.

3.3 HTML5 Canvas Guide

url: <https://developer.apple.com/library/safari/documentation/AudioVideo/Conceptual/HTML-canvas-guide/DrawingLinesandShapes/DrawingLinesandShapes.html>

Možno nie pre začiatočníkov, ale o to komplexnejšie popísaná práca s HTML5 canvasom sa dá nájsť pri dokumentáciách pre vývojárov k jednotlivým prehliadačom. Konkrétne URL vyššie nám pomohla pri jednotlivých nástrojoch, kde potrebujeme rôzne "správanie sa" čiar - vieme nastaviť, ako majú vyzeráť konce úsečiek tvoriacich nejaký celok. V článku je detailne popísaná práca s čiarami a základnými geometrickými tvarmi - obdĺžnik, krivka, oblúk a kružnica/kruh.

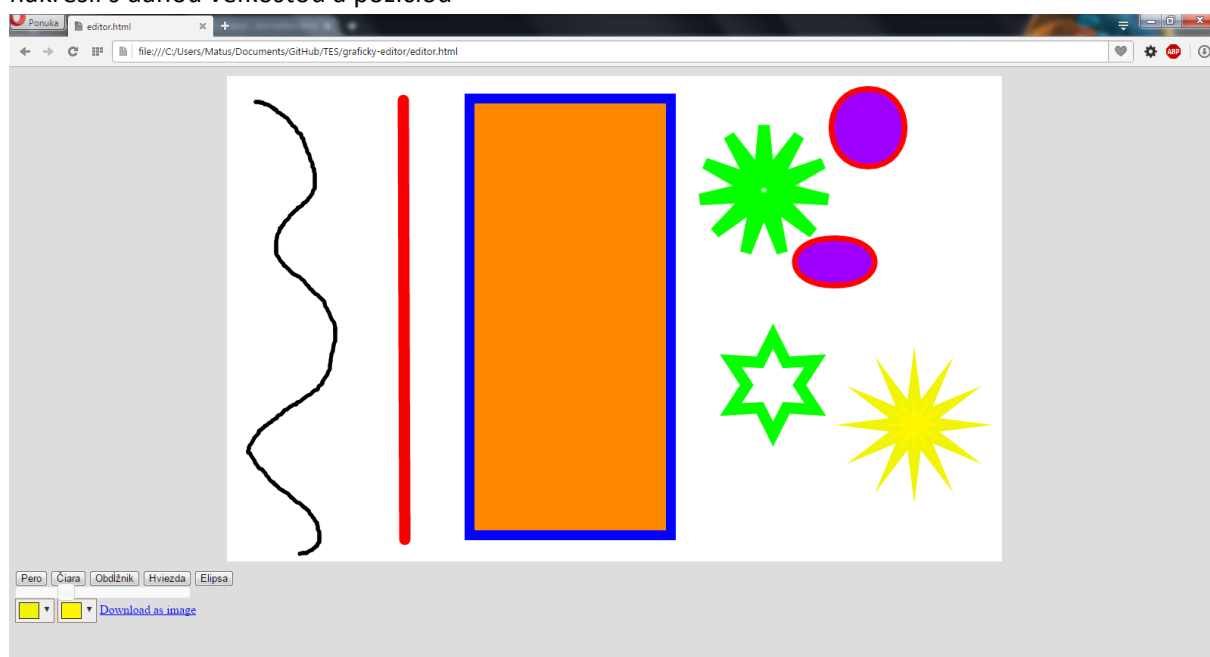
Implementačná časť

1.etapa (8.3.2016)

- Vytvorenie stránky k projektu využitím vhodných webových technológií – HTML5, JavaScript, JQuery, Bootstrap, CSS3, HTML5 Canvas
 - stránka projektu: <http://adam.studenthosting.sk/TES/>
 - editor: <http://adam.studenthosting.sk/TES/editor.html>
- Vytvorenie základnej HTML stránky pre vznik grafického editora s importovaním JavaScript knižnice JQuery
- Implementovanie prvého základného nástroja – pero. Týmto nástrojom sa dá jednoducho kresliť ťahaním kurzora po ploche tzv. dragovaním
- Implementovanie tlačidla na uloženie obrázku na lokálny disk

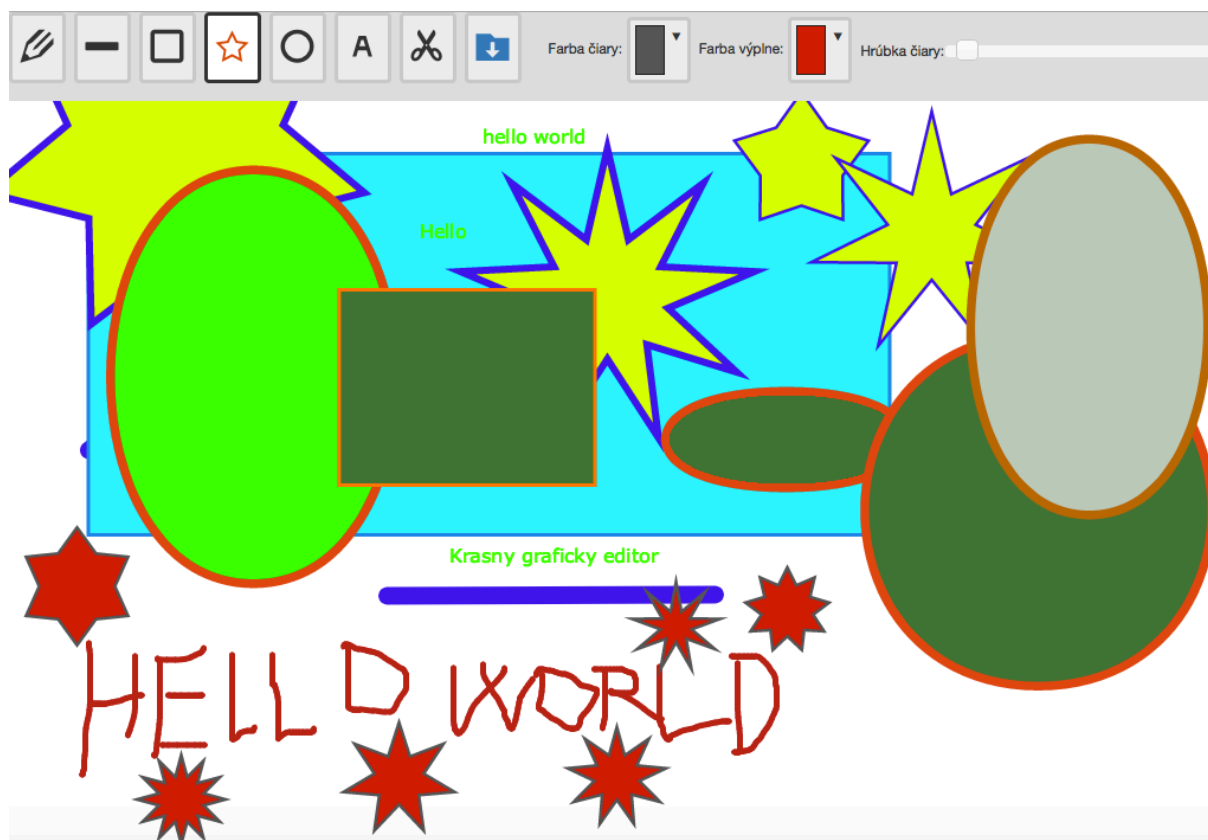
2.etapa (22.3.2016)

- Úprava nástroja pero – Úprava interpolácie pomocou kubického splajnu
- Pridaná vlastnosť nástrojov – nastavenie hrúbky nástroja, zapamätanie si hrúbky nástroja
- Pridanie komponentu tzv. „ColorPicker-a“, ktorý umožňuje výber farby pre čiary nástrojov
- Taktiež možnosť zvoliť si farbu výplne. Farbu výplne možno nastaviť aj na transparentnú (priehľadnú), teda nástroj kreslí bez výplne
- Implementovanie nového nástroja Čiara. Nástroj nakreslí čiaru s farbou a hrúbkou akú si užívateľ zvolí pred kreslením
- Implementovanie nástroja Elipsa, ktorý umožňuje kresliť elipsy a kružnice
- Implementovanie nástroja Štvoruholník, ktorý umožňuje kresliť štvorce a obdĺžniky
- Implementovanie nového nástroja Hviezda (nástroj pre deti) - kreslí náhodné n-cípe hviezdy. Dieťa môže meniť veľkosť hviezdy a jej pozíciu ťahaním po kresliacej ploche
- Pridanie vlastnosti pre nové nástroje – pokým užívateľ nepustí tlačidlo myši (dragovanie), tak môže meniť veľkosť a pozíciu objektu, ktorý práve kreslí. Po uvoľnení tlačidla myši sa objekt nakreslí s danou veľkosťou a pozíciou



3. etapa (12.4.2016)

- Úprava nástroja hviezda - nástroj hviezda kreslí hviezdy s náhodným počtom (max 12) cipov a náhodným vnútorným polomerom. Aby hviezdy vizerali vizuálne krajšie, zmenšil sa počet cipov a minimálny počet vnútorného polomeru sa zväčšil
- Pridanie nástroja Text - používatelia majú možnosť pridávať jednoriadkový text do obrázka. Môžu pridať text obsahujúci znaky, čísla a medzery
- Pridanie nástroja Výrez - nástroj umožňujúci vyrezať časť nakresleného obrázka. Na vyrezaný obrázok sa majú dať aplikovať transformácie - škálovanie v pomere, rotácia a translácia. Zložitou časťou je vymyslieť interaktívny spôsob, ako je pre deti najjednoduchšie a prirodzené aplikovať tieto transformácie na výrez.
- Grafické spracovanie - pridali sme grafické spracovanie pre textový editor
 - funkcie na kreslenie (pero, čiara, ...) sú umiestnené v hornej časti. Jednotlivé funkcie sú vizualizované pomocou obrázka, ktorý by mal byť intuitívny. Obrázky sme prevzali z voľne dostupnej zbierky obrázkových fontov pomocou online aplikácie <https://icomoon.io/>
 - Po kliknutí na niektorý obrázok nástroja sa tento nástroj zvýrazní (tmavší rámček, bledšie pozadie, výraznejšia farba), ostatné sú nevýrazné.
 - Vľavo sú umiestnené nástroje na kreslenie a vpravo sú atribúty nástrojov - napr. farba čiary/výplne, hrúbka pera.



4. Etapa (9.5.2016)

Programátorsky zaujímavé časti: Projekt obsahuje niekoľko tried, ktoré sú pri spustení inštancované. Zaujímavá je trieda `AbstractTool`, od ktorej "dedia" triedy jednotlivých nástrojov na kreslenie. `AbstractTool` vytvorí `Object` literal, ktorému pridá spoločné metódy a členské premenné a vráti ho. Vrátený `Object` literal môže byť ľubovoľne modifikovaný pri vytváraní nástroja. Tým sme dosiahli jednoduchý "polymorfizmus" v Javascripte.

Trieda `ToolManager` spravuje objekty jednotlivých nástrojov, uchováva si referenciu na aktuálne zvolený nástroj. Odchytáva udalosti a preposiela ich objektu prislúchajúcemu zvolený nástroj. Vždy sa kontroluje, či objekt obsahuje metódu prislúchajúcu preposielaniu udalosti a ak áno, táto metóda sa na ňom zavolá. Sú to napr. metódy:

- `enable()` - pri zvolení aktuálneho nástroja
- `disable()` - pri odchode z aktuálneho nástroja
- `click(eventObj)` - pri kliknutí do plátna
- `dragStart(eventObj)` - pri začatí dragovania na plátne
- `drag(eventObj)` - pri dragovaní na plátne
- `dragEnd(eventObj)` - pri ukončení dragovania na plátne
- `penWidthChanged(value)` - pri zmene hrúbky čiary
- `lineColorChanged(color)` - pri zmene farby čiary
- ...

Jednotlivé triedy pre nástroje obsahujú aj metódy `paint()`, resp. `repaint()`, ktoré buď kreslia do kanvasu alebo sa prekresľujú na vrchnom kanvase a volajú sa pri udalostiach.

V projekte využívame viacero HTML5 canvas elementov. Plátno, do ktorého kreslíme nástroje obsahuje dva elementy. Spodný je pre časti obrázka, ktoré sa už nebudú prekreslovať a vrchný je na prekresľovanie (napr. keď kreslíme štvorec, alebo hviezd, tak sa pri dragovaní kanvas prekresľuje, aby mal používateľ predstavu, ako bude vyzeráť, keď dokončí kreslenie). Takisto, pri ukladaní obrázka chceme uložiť len výber, ktorý vznikne po zmene pozícií pravého horného rohu a ľavého pri zmene veľkosti plátna. Túto časť si prekopírujeme do nového canvas elementu a prevedieme na PNG formát pomocou vstavanej funkcie.

Zaujímavým problémom bola aj práca s výrezom, kde bolo treba zakomponovať interaktívne škálovanie, rotáciu a transformáciu. Jednotlivé parametre pre transformácie sme mali uložené v premenných a počas práce s výrezom sme ich aplikovali na pôvodný výrez. Inak dochádzalo postupne k znižovaniu kvality obrázka vo výreze.

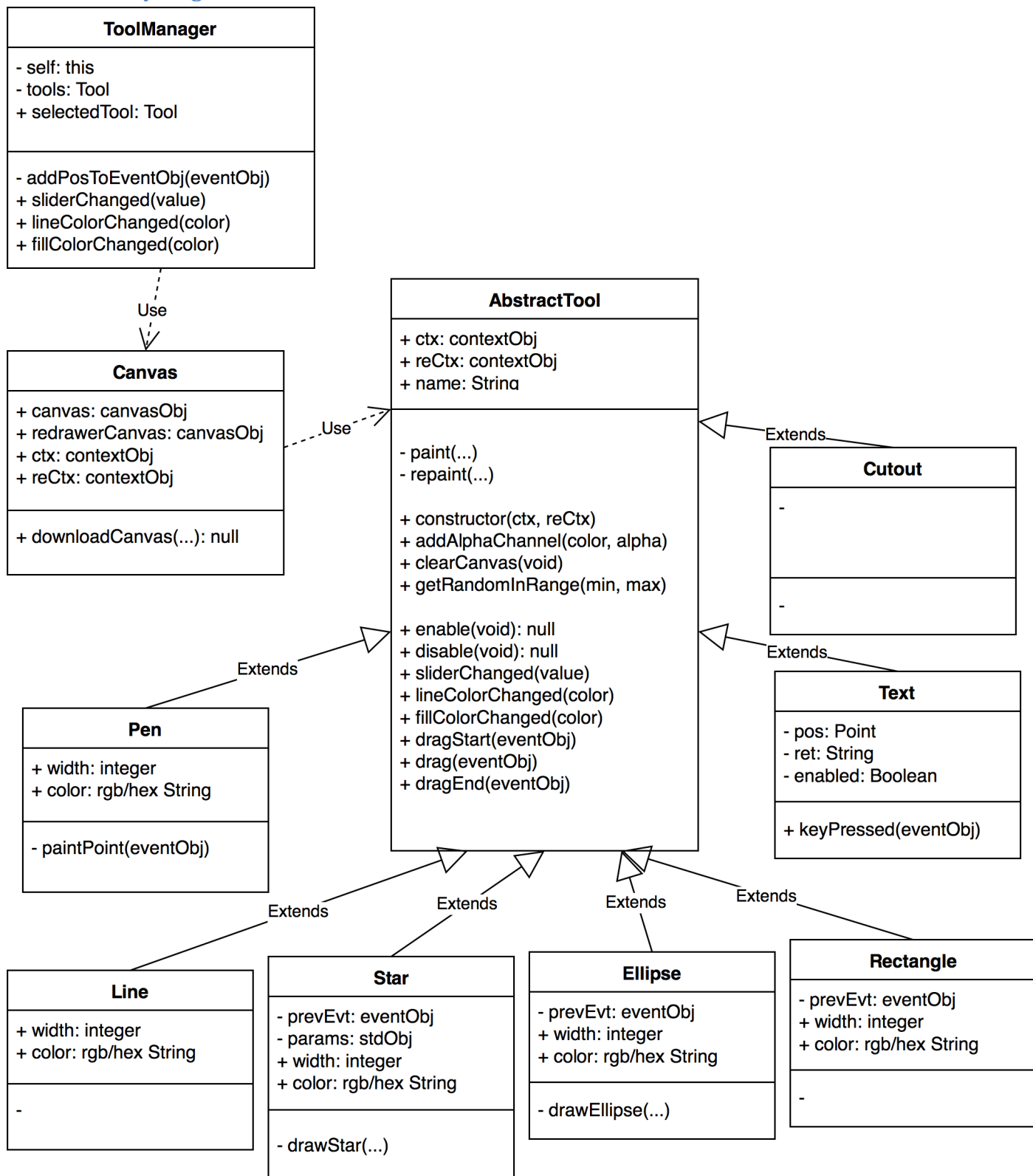
Od poslednej etapy sme spravili viacero zmien, najdôležitejšími sú:

- fix pre prácu s výrezom
- pridanie klávesových skratiek pre prácu s výrezom: `ctrl+v`, `delete`, `esc`
- fix pre zvolené uchovávanie atribútov (farby pera, výplne, ...) nástrojov
- fix pre nástroj text, možnosť písať viacriadkový text, funguje `backspace`, zmena veľkosti písma
- pridanie nástroja nový papier
- pridanie nástroja nový papier
- pridanie nástroja guma so zvolenou hrúbkou

- pridanie nástroja pečiatka
- pridanie zmenšovania/zväčšovania plátna
- fix pri ukladaní obrázka pri zmene veľkosti plátna
- pridanie kroku späť - zmaže všetko, čo sa nakreslilo s aktuálnym nástrojom

UML diagramy:

Triedny diagram



Obrázok 1 Znáornenie triedneho diagramu aktuálnej verzie editora

Budúce etapy

- Doplňenie základných nástrojov na kreslenie
 - Možnosť načítania obrázku z lokálneho disku
 - Vytvorenie špeciálnych nástrojov pre deti
 - Implementovanie výberu farby pre nástroje
 - Upraviť grafický vzhľad aplikácie (obrázky pre nástroje a pod.)
-
- Pridanie nástroja na pridávanie textu
 - Možnosť zvoliť si pozadie na ktoré sa kreslí
 - Implementovanie „gumy“ bez mazania pozadia
-
- Dokončiť nástroje - guma, obrázok na pozadí
 - Dokončiť atribúty nástrojov a ich zapínanie/vypínanie - farba čiary, výplne, hrúbka obrysu, ...
 - Upraviť grafický design podľa možností
 - Prípadné zjednodušenie ovládania
 - Skontrolovanie chýb, oprava chýb