



Středoškolská technika 2019

Setkání a prezentace prací středoškolských studentů na ČVUT

Program pro vytváření a editování barevných animací na LED panelech

Kryštof Mára

VOŠ.SŠ, COP

Dukelská 421, Sezimovo Ústí,

Prohlášení

Prohlašuji tímto, že jsem maturitní projekt vypracoval samostatně pod vedením pana učitele Jiřího Bumby a uvedl jsem veškerou použitou literaturu a další informační zdroje včetně internetu.

V Sezimově Ústí dne

podpis autora

Poděkování

Maturitní práce byla zpracována jako závěrečný projekt v rámci řádného ukončení 4. ročníku maturitního studia Počítače a Robotika – Programování. Vedoucím práce byl pan učitel Jiří Bumba, kterému tímto děkuji za cenné rady týkající se struktury i obsahu práce. Cenné rady mi poskytl také odborný konzultant pan Jakub Macillis.

Anotace

Práce popisuje program k vytváření animací pro LED panel. Je zde podrobně vysvětleno, proč jsem si tuto práci vybral, čím je užitečná a jak funguje. Dozvíte se zde také, jaký programovací jazyk a software jsem využil, a jak jsem svého cíle dosáhl.

Annotation

The work describes a program for creating animations for the LED panel. It is explained in detail why I have chosen this work, how useful it is and how it works. You will also find out what programming language and software I have used and how I have achieved my goal.

Klíčová slova

EALeD; LED panel; Visual Studio; C-Sharp (C#)

Obsah

1	Úvod.....	6
2	ANALÝZA ZADÁNÍ A NÁVRH ŘEŠENÍ.....	6
2.1	Programovací jazyk JAVA	7
2.2	Programovací jazyk C++	7
2.3	Programovací jazyk C#	8
2.4	Programovací rozhraní Visual Studio	8
3	EALÉD	9
3.1	Přehledové panely	10
3.1.1	ANIM panel	11
3.1.2	LIST panel	12
3.1.3	BOX panel	14
3.2	Ovládací panely.....	15
3.2.1	RGB panel.....	18
3.2.2	TIME panel	19
3.3	Extra prvky.....	20
3.3.1	DGV	20
3.3.2	Vzory animací.....	22
3.4	Externí funkce	26
3.4.1	Panel tlačítek externích funkcí.....	26
3.4.2	Sériová komunikace.....	29
3.4.3	ColorSave a HexCode.....	31
4	Závěr	33
5	Použitá literatura	34
6	Seznam obrázků a tabulek	34
7	Přílohy na CD	35

1 ÚVOD

Ve škole, kde studuji, bylo potřeba vytvořit aplikaci pro vytváření barevných animací, které by rozsvěcely RGB LED diody ve světelném panelu. Tuto práci jsem si zvolil proto, že se velmi zajímám o programování a chci se v něm zdokonalit. Mým úkolem bylo vytvořit program, který by umožnil pohodlný návrh a editování barevných světelných animací na obrazovce počítače a odeslání pořízených dat po sériovém portu do mikroprocesoru LED panelu.

Se samotným programováním jsem se setkal poprvé v hodinách programování ve třetím ročníku studia mého oboru. Programování jsem si oblíbil a rychle jsem se v něm zlepšoval. Bohužel, jakmile začal můj čtvrtý ročník, mé hodiny programování skončily, a tak mé zkušenosti stagnovaly. Má práce SOČ byla to jediné, co mě inspirovalo se k programování vrátit a dále se v něm zlepšovat.

2 ANALÝZA ZADÁNÍ A NÁVRH ŘEŠENÍ

Minulý rok vytvořil žák školy propagační LED panel vytvářející barevné světelné animace. Vytváření těchto světelných animací bylo pracné a ten kdo je vytvářel, neměl okamžitou zpětnou vazbu, aby viděl výsledek své práce. Proto vznikla potřeba vytvořit program, který by tuto práci usnadnil a kde by tvůrce okamžitě viděl výsledek své práce. Mým úkolem bylo vytvoření programu pro vytváření a editování animací pro tento LED panel.

Bylo požadováno, aby program spustil na obrazovce počítače vývojové rozhraní pro vytvoření světelných animací pro LED panel. Musí umožňovat tuto animaci uložit a načíst do externích souborů a dále tuto animaci poslat přes sériovou komunikaci do LED panelu. Ovládání musí být snadné a uživatelsky přívětivé. Program by také měl nabízet několik přednastavených vzorů animací.

LED panel je zařízení obsahující šestnáct řádků s RGB LED diodami pod sebou. Každý řádek se může nezávisle na ostatních rozsvítit jakoukoliv barvou. Barevný odstín každého řádku vytváří mikropočítač uvnitř panelu mícháním tří základních barev (červené, zelené a modré). Informace o odstínu každé řádky v každém časovém intervalu má mikroprocesor uložené v paměti.

K realizaci mého programu bylo zapotřebí vybrat programovací jazyk a programovací rozhraní, ve kterém by se program vyvíjel. Měl jsem na výběr ze tří programovacích jazyků, a to JAVA, C++ a C#. Krátká charakteristika těchto jazyků je uvedena v následujících kapitolách. S jazykem C-Sharp jsem se naučil pracovat v mých hodinách programování. Proto jsem si zvolil tento jazyk k realizaci zadaného úkolu. Jako programovací rozhraní jsem využil Visual Studio, jelikož žádné jiné rozhraní neznám a podporuje všechny jazyky, které mám na výběr.

2.1 Programovací jazyk JAVA

„Java je objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems a představila 23. května 1995.

Jde o jeden z nejpoužívanějších programovacích jazyků na světě. Díky své přenositelnosti je používán pro programy, které mají pracovat na různých systémech počínaje čipovými kartami, přes mobilní telefony a různá zabudovaná zařízení, aplikace pro desktopové počítače až po rozsáhlé distribuované systémy, pracující na řadě spolupracujících počítačů, rozprostřené po celém světě.

Tyto technologie se jako celek nazývají platforma Java. Dne 8. května 2007 Sun uvolnil zdrojové kódy Javy a Java bude dále vyvíjena jako open source [1].“



Obr. 2.1 Logo jazyka Java [2]

2.2 Programovací jazyk C++

„C++ je multiparadigmatický programovací jazyk, který vyvinul Bjarne Stroustrup a další v Bellových laboratořích AT&T rozšířením jazyka C.

C++ podporuje několik programovacích stylů jako je procedurální programování, objektově orientované programování a generické programování, není tedy jazykem čistě objektovým. V současné době patří C++ mezi nejrozšířenější programovací jazyky [3].“



Obr. 2.2 Logo jazyka C++ [4]

2.3 Programovací jazyk C#

„C-Sharp je jednoduchý, moderní, mnohoúčelový a objektivě orientovaný programovací jazyk, jehož implementace poskytuje podporu pro principy softwarového inženýrství. Jeho důležité vlastnosti jsou robustnost, trvanlivost a programátorská produktivita.

C-Sharp je vhodný pro vývoj softwarových komponentů distribuovaných v různých prostředích a je navržen pro psaní aplikací jak pro zařízení se sofistikovanými operačními systémy, tak pro zařízení s omezenými možnostmi.

Přestože by programy psané v C# neměly plýtvat s přiděleným procesorovým časem a pamětí, nemohou se měřit s aplikacemi psanými v C nebo jazyce symbolických adres [5].“



Obr. 2.3 Logo jazyka C-Sharp (C#) [6]

2.4 Programovací rozhraní Visual Studio

Mé programovací rozhraní, které mi umožnilo tento program vytvořit, byl programovací software Visual Studio.

„Microsoft Visual Studio je vývojové prostředí vyvinuto firmou Microsoft. Může být použito pro vývoj konzolových aplikací a aplikací s grafickým rozhraním spolu s aplikacemi Windows Forms, webovými stránkami, webovými aplikacemi a webovými službami jak ve strojovém kódu, tak v řízeném kódu na platformách Microsoft Windows, Windows Mobile, Windows CE, .NET, .NET Compact Framework a Microsoft Silverlight.

Visual Studio podporuje jazyky prostřednictvím jazykových služeb, což umožňuje, aby editor kódu a debugger podporoval jakýkoliv programovací jazyk. Mezi vestavěné jazyky patří C/C++, VB .NET a C#.

Podpora dalších jazyků jako Oxygene, F#, Python a Ruby spolu s ostatními může být přidána jazykovými službami, které musí být nainstalovány zvlášť.

Také je podporováno XML/XSLT, HTML/XHTML, JavaScript a CSS. Existují i verze Visual Studia pro určitý jazyk, které uživatelům poskytují omezenější jazykové služby. Tyto individuální balíčky jsou Microsoft Visual Basic, Visual J#, Visual C# a Visual C++ [7].“

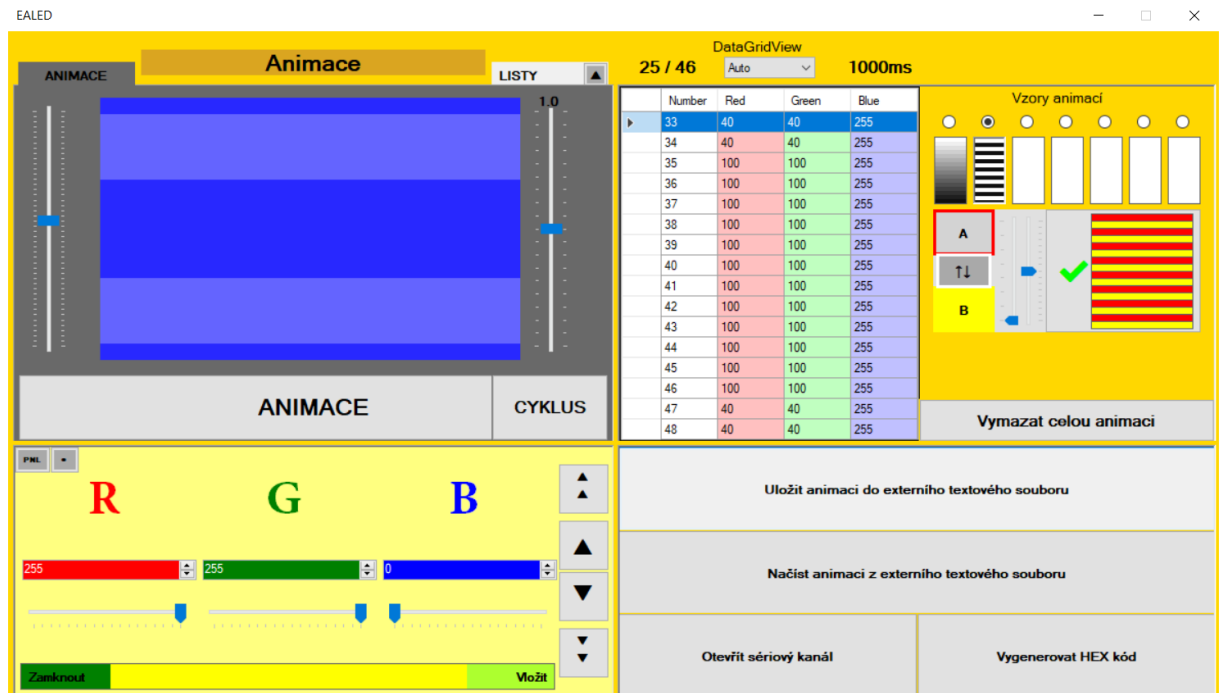
S programovacím rozhraním Visual Studio jsem se setkal spolu s programovacím jazykem C-Sharp v hodinách programování. Jelikož žádné jiné programovací rozhraní neznám, Visual Studio byla má jediná volba.



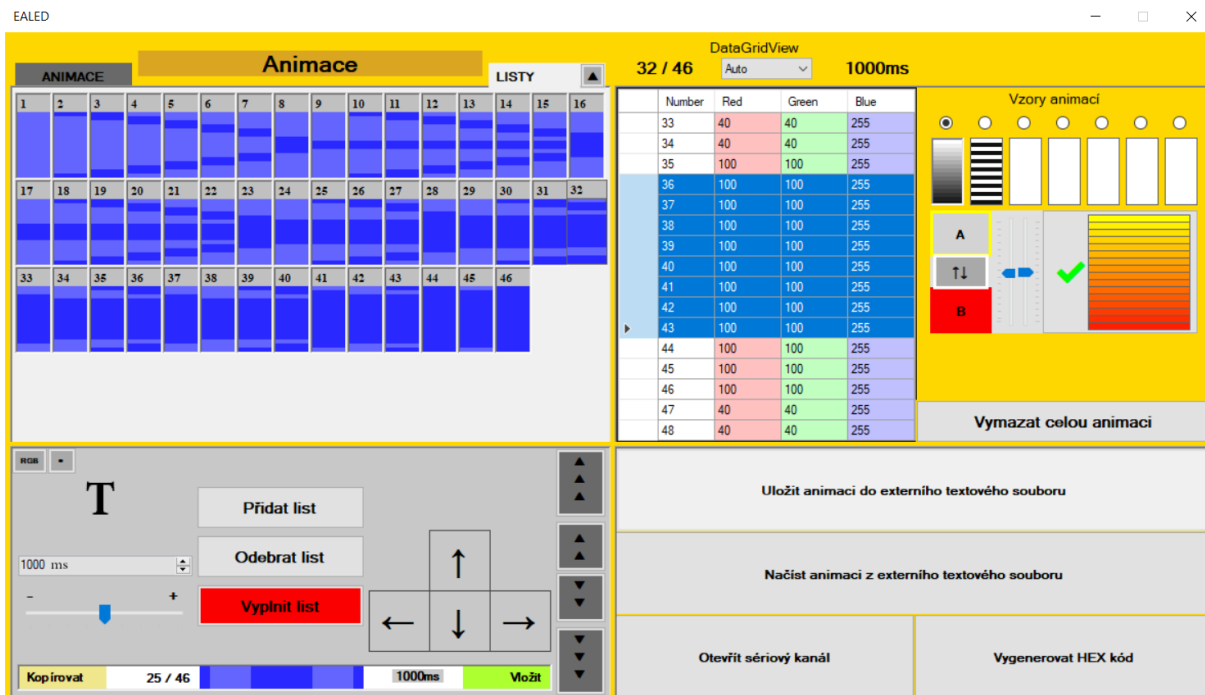
Obr. 2.4 Logo programovacího rozhraní Visual Studio [8]

3 EALÉD

Program pro vytváření světelných animací jsem nazval Ealed což znamená: Editace Animace LED. Obsahuje vše potřebné pro vytvoření animace, jako je nastavování barev jednotlivých řádků, prodleva mezi jednotlivými kroky animace, přehled nad každým krokem animace a uložit nebo načíst animaci z externí složky. Program umožňuje zakódovat hotovou animaci do hexadecimální podoby, která se dá odeslat přes sériový port do mikroprocesoru LED panelu.



Obr. 3.1 Program EALÉD (jeden krok)



Obr. 3.2 Programu Ealed (celá animace)

Rozhraní Ealedu je rozděleno do čtyř částí:

- Přehledové panely jsou vlevo nahoře a obsahují ANIM panel, LIST panel a BOX panel. Tyto panely umožňují kompletní přehled nad vytvořenou animací a představují rychlý způsob, jak editovat a pohybovat se mezi jednotlivými kroky animace.
- Ovládací panely, které jsou vlevo dole, slouží k míchání barev a nastavování prodlevy mezi jednotlivými kroky animace. Mezi ně patří RGB panel a TIME panel.
- Extra panely vpravo nahoře jsou panely jako DGV a vzory animací. Ty umožňují efektivnější editování kroků animace.
- Externí panely vpravo dole slouží pro práci s externími soubory. Zde se ukládá, načítá a generuje hexadecimální kód, který se posílá přes sériový port do LED panelu.

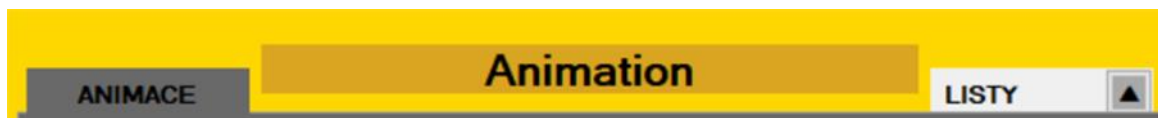
3.1 Přehledové panely

Samotná animace pro LED panel se skládá z jednotlivých kroků, zobrazujících se za sebou. Každý krok animace se skládá z šestnácti řádkové obrazovky a délky trvání jejího zobrazení. Přehledové panely proto slouží, aby uživatel měl stálý přehled nad každým krokem své animace a pro jednodušší editování individuálních řádků i celých kroků. Ukázky přehledových panelů jsou na obrázcích Obr. 3.5, Obr. 3.7 a Obr. 3.11.

Všechny přehledové panely sdílí tu samou pozici vlevo nahoře. K výběru požadovaného panelu a přepínání mezi nimi slouží tlačítka umístěná v horní části. Vlevo je tlačítko ‚ANIMACE‘, které slouží k přepnutí aktuálního přehledového panelu na ANIM panel.

V pravé části jsou dvě tlačítka. Tlačítko ‚KRABICE‘ slouží k přepnutí na LIST panel a tlačítko se šipkou nahoru ‚▲‘ slouží k přepnutí na BOX panel.

Uprostřed mezi tlačítky se zobrazuje jméno samotné animace, viz Obr. 3.3. Toto jméno je pak využito k pojmenování externí složky při jejím vytváření. Při načítání se zde naopak jméno animace zobrazí. Jméno lze kdykoliv změnit při vytváření animace.



Obr. 3.3 Horní část přehledových panelů



Obr. 3.4 Horní část přehledových panelů

3.1.1 ANIM panel

ANIM panel je nejzákladnějším ukazatelem stavu barev v kroku animace.

Disponuje šestnácti řádky barev, které představují jednotlivé řádky LED panelu. Umožňuje spustit samotný proces animace, ovládat její rychlost a pohyb uvnitř animace.

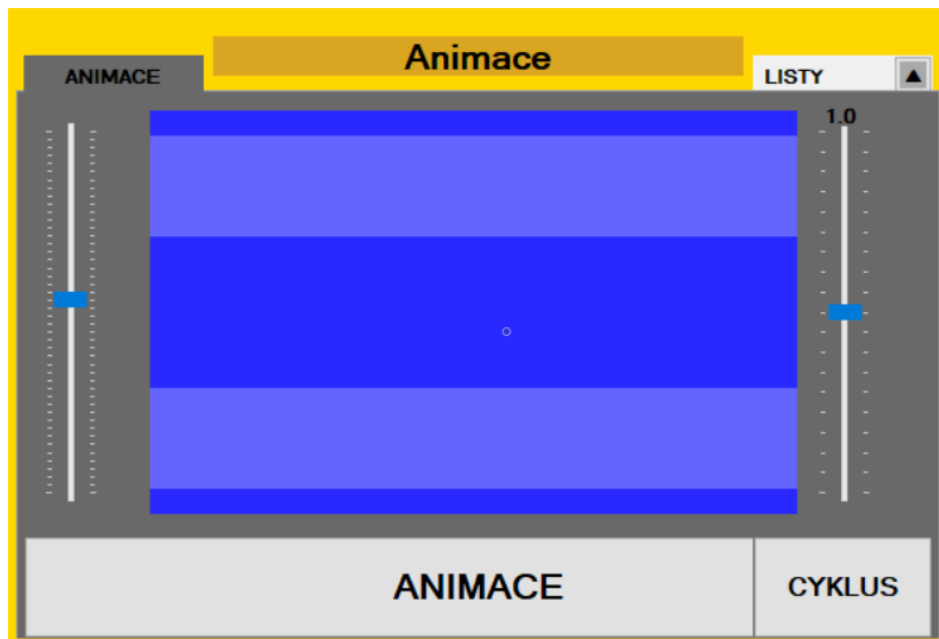
Uprostřed ANIM panelu je zobrazena pracovní plocha kroku animace. Při kliknutí na kteroukoliv řádku se tato zvýrazní černým orámováním a barvu této řádky lze konfigurovat pomocí ovládacích prvků RGB panelu.

Ve spodní části ANIM panelu jsou dvě tlačítka ovládající spuštění animace. Tlačítko ‚ANIMACE‘ slouží k spuštění animace jednou, kdežto tlačítko ‚CYKLUS‘ slouží k nepřetržitému provádění animace dokola, dokud se tlačítko nesepe znova.

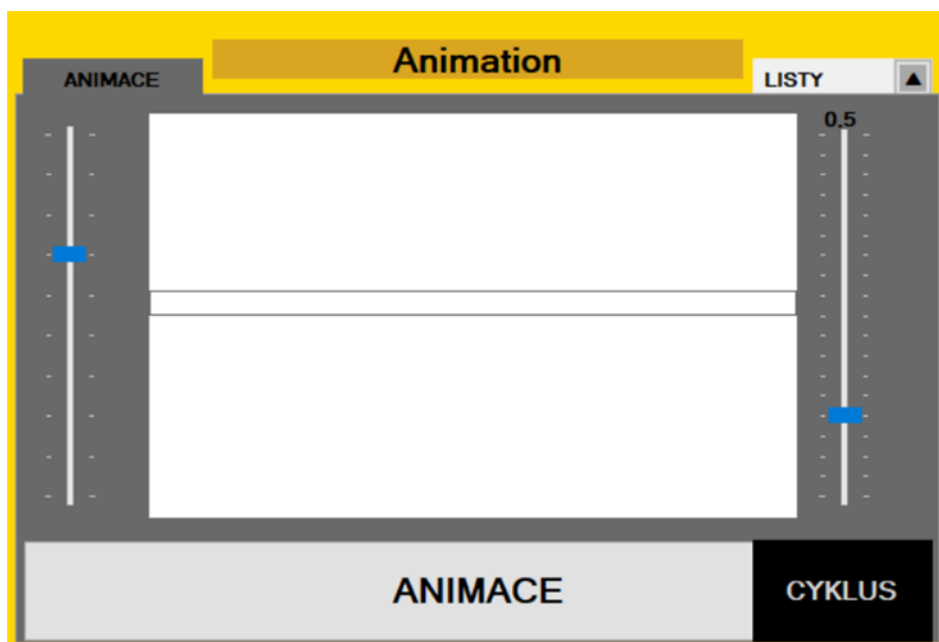
Vlevo ANIM panelu je potom posuvník sloužící pro ovládání pozice animace. Je to rychlá a efektivní možnost procházet mezi kroky animace.

Vpravo na panelu je druhý posuvník ovládající rychlost přehrávání animace. Ovládání funguje tak, že se hodnotou, odpovídající nastavené poloze na posuvníku, násobí prodleva mezi individuálními kroky animace. Prodleva se může pohybovat v rozmezí od 1 desetin až do 10 násobku skutečné (normální) rychlosti.

To znamená, že při hodnotě 1.0 se prodleva každého kroku násobí číslem 1 a animace bude probíhat nominální rychlostí bez jakékoliv změny. Při hodnotě 10 se prodleva mezi kroky navýší desetkrát a animace bude probíhat desetkrát pomaleji. Pro hodnoty menší než 1 se délka prodlevy zkrátí. Např. pro 0.1 bude animace probíhat desetkrát rychleji.



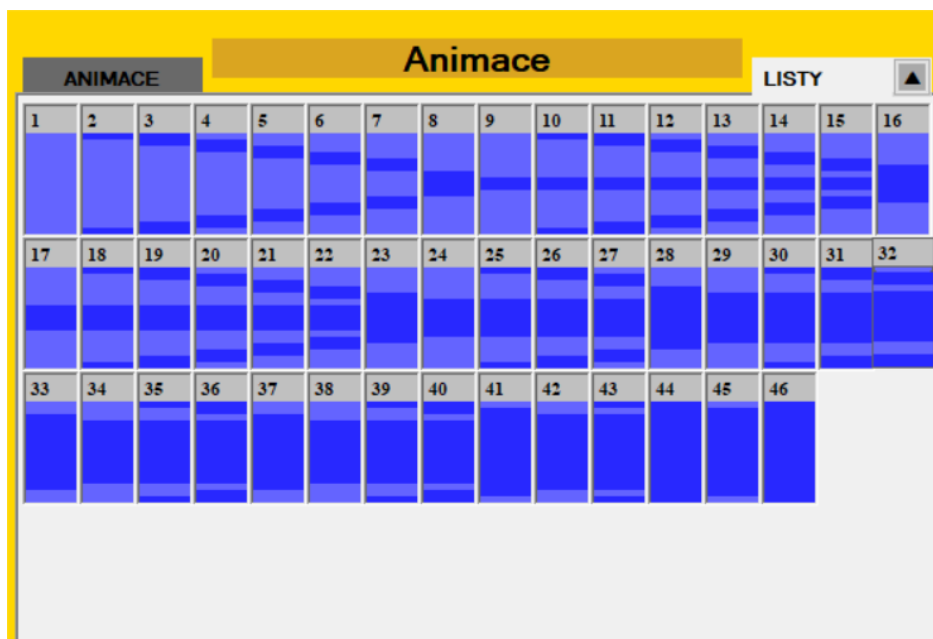
Obr. 3.5 Přehledový panel ANIM panel (s krokem animace)



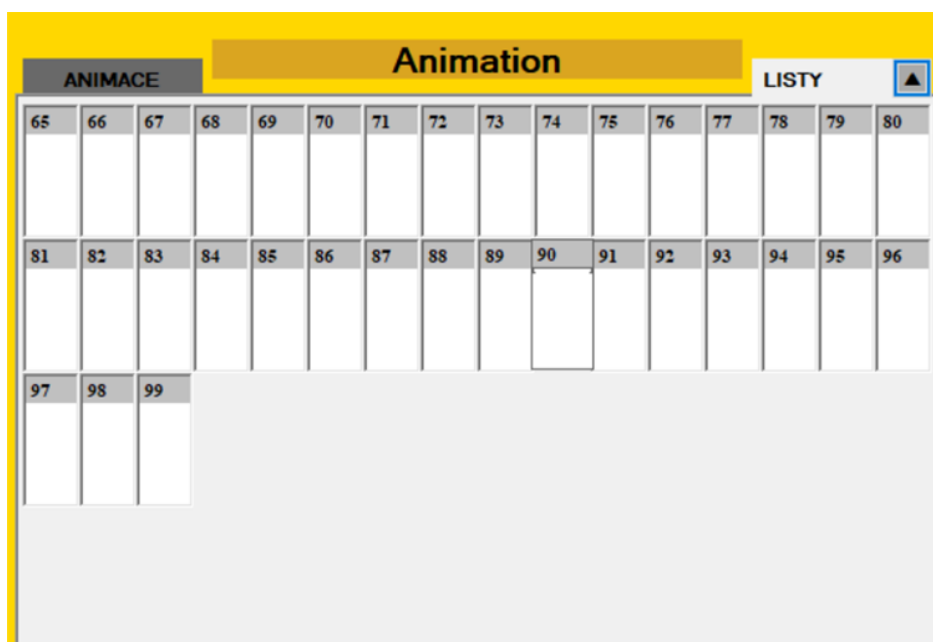
Obr. 3.6 Přehledový panelu ANIM panel (prázdný)

3.1.2 LIST panel

LIST panel je sekundární přehledový panel, který na rozdíl od ANIM panelu, slouží k zobrazení vyššího množství kroků animace najednou v podobě listů. LIST panel pouze zobrazuje přehled listů animace, umožňuje výběr požadované ho listu a přechod do jeho animace.



Obr. 3.7 Přehledový panel LIST panel (náhodně vyplněné listy)



Obr. 3.8 Přehledový panel LIST panel (nevyplněné listy)

Každý list má v horní části číslo, které určuje pořadí kroku v animaci. Obdobně jako u ANIM panelu, při kliknutí na jakýkoliv list se tento list označí a s ním se poté může pracovat pomocí kombinací ostatních ovládacích nebo extra prvků Ealedu. Kliknutím dvakrát na list přepne LIST panel zpět na ANIM panel, který umožní tento list editovat. Tato funkce slouží k rychlému přesouvání mezi různými kroky animace a jejich následné editaci.

4	5	6

Obr. 3.9 List LIST panelu (vyplněný)

89	90	91

Obr. 3.10 List LIST panelu (prázdný)

3.1.3 BOX panel

LIST panel může zobrazit až šedesát-čtyři listů současně. Vytvořením více než šedesáti-čtyř listů Ealed vynutí vytvoření dalšího LIST panelu. BOX panel proto slouží pro přehled nad všemi vytvořenými LIST panely.

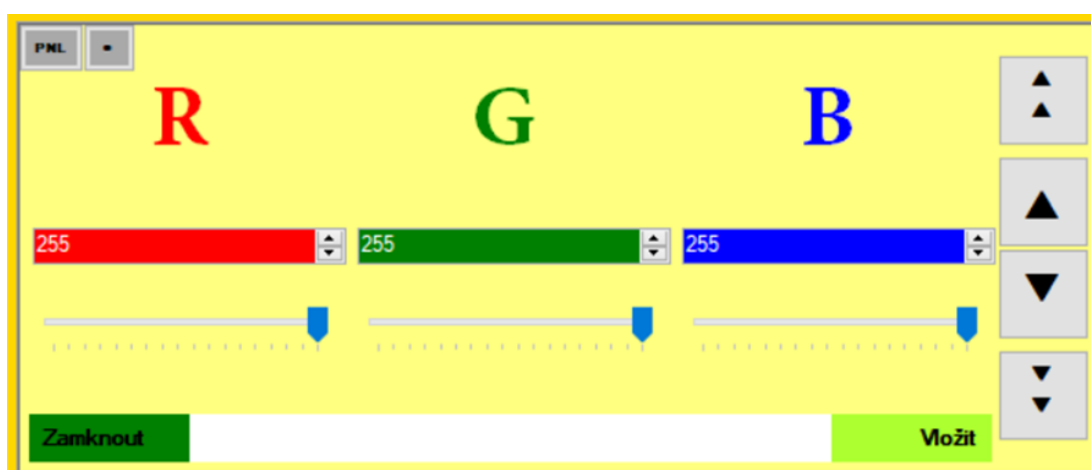
BOX panel funguje víceméně identicky jako LIST panel. Kliknutím dvakrát na list v BOX panelu přepne aktivní přehledový panel zpátky na LIST panel, který byl přiřazen k tomuto listu.

Animation										
ANIMACE										LISTY ▲
1	65	129	193	257	321	385	449	513	577	641
64	128	192	256	320	384	448	512	576	640	704
705	769	833	897	961						
768	832	896	960	1024						

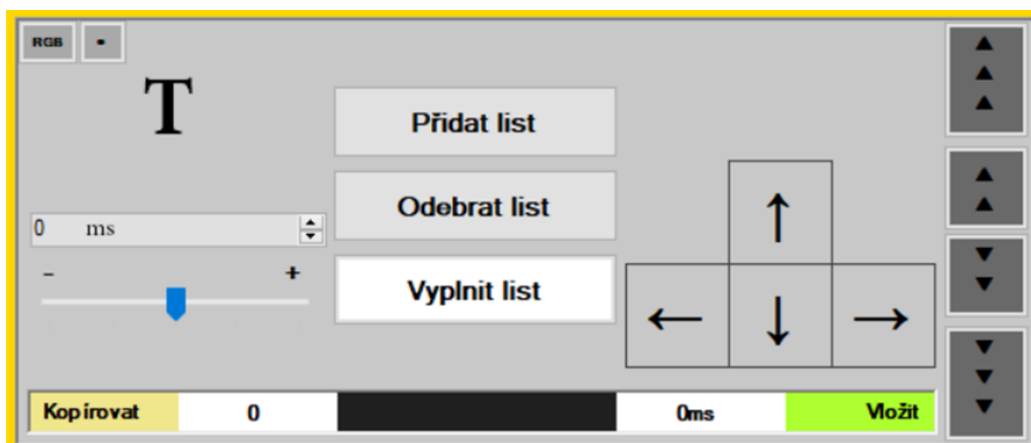
Obr. 3.11 Přehledový panel BOX panel

3.2 Ovládací panely

Ovládací prvky na ovládacím panelu slouží k samotné konfiguraci vlastností kroků animace, jako jsou samotné barvy řádků, čas prodlevy a pozice kroků animace. Obdobně jako u přehledových panelů Ealedu i ovládací panely sdílí společnou pozici. Ovládací panely mezi sebou přepínají automaticky podle toho, jaký přehledový panel je zrovna aktivní.

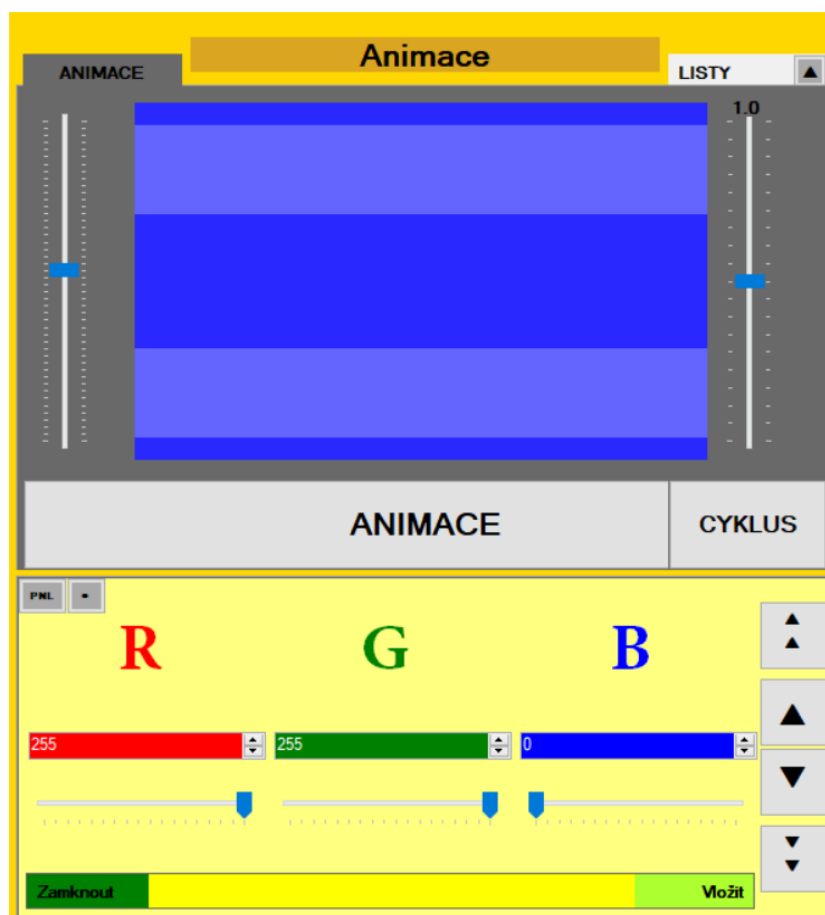


Obr. 3.12 Ovládací prvky RGB panelu

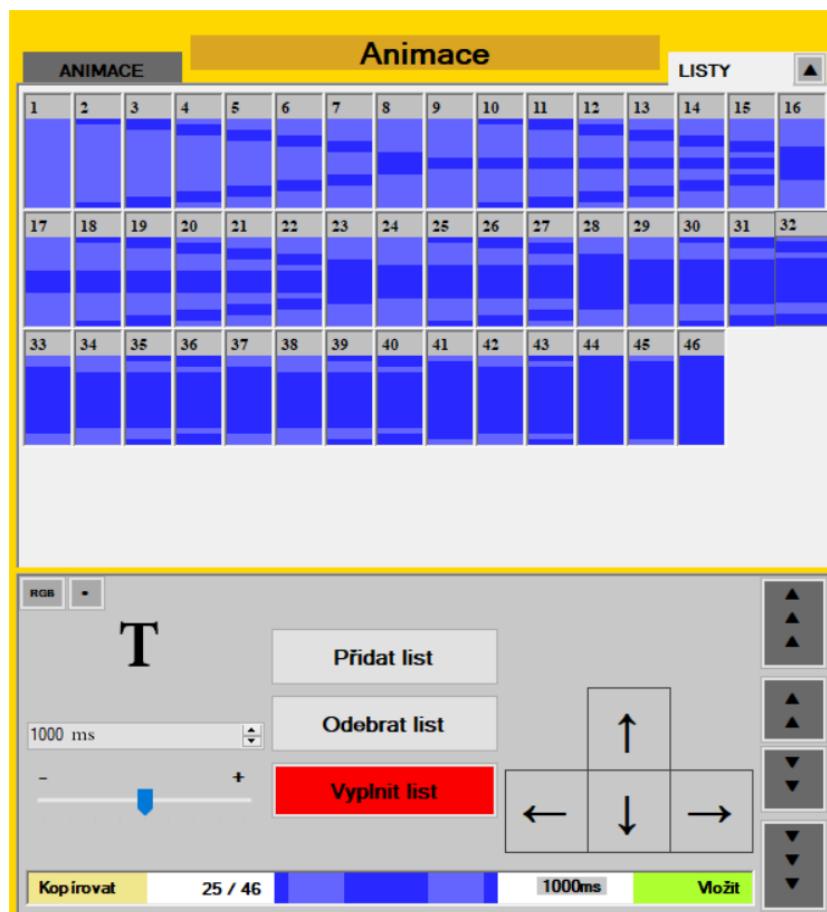


Obr. 3.13 Ovládací prvky TIME panelu

Ovládací RGB panel je tak spojen s přehledovým ANIM panelem. Pokaždé, když se přepne přehledový panel na ANIM panel, přepne se současně i ovládací panel na RGB panel. Stejně tak i TIME panel je spojen s LIST panelem a přepíná se spolu s ním.



Obr. 3.14 Ukázka závislosti ovládacích prvků na přehledových prvcích (ANIM - RGB)



Obr. 3.15 Ukázka závislosti ovládacích prvků na přehledových prvcích (BOX - TIME)

K manuálnímu přepínání mezi ovládacími panely slouží dvě tlačítka v levém horním rohu každého ovládacího panelu, viz Obr. 3.16 a Obr. 3.17. Pravé tlačítko slouží k samotnému přepínání mezi ovládacími panely, kdežto levé tlačítko slouží k uzamčení ovládacího panelu, který pak bude ignorovat jakékoliv další automatické přepínání.



Obr. 3.16 Tlačítka k přepínání ovládacích prvků (TIME)



Obr. 3.17 Tlačítka k přepínání ovládacích prvků (RGB)

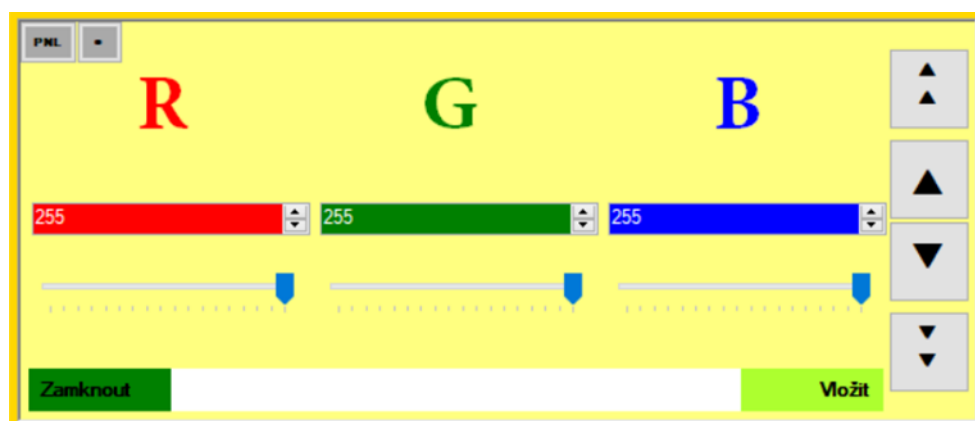
3.2.1 RGB panel

RGB panel je nejzákladnějším ovládáním Ealedu. Zde se míchají veškeré barvy, které se pak využívají v ostatních komponentech, jako jsou řady barev v každém kroku animace.

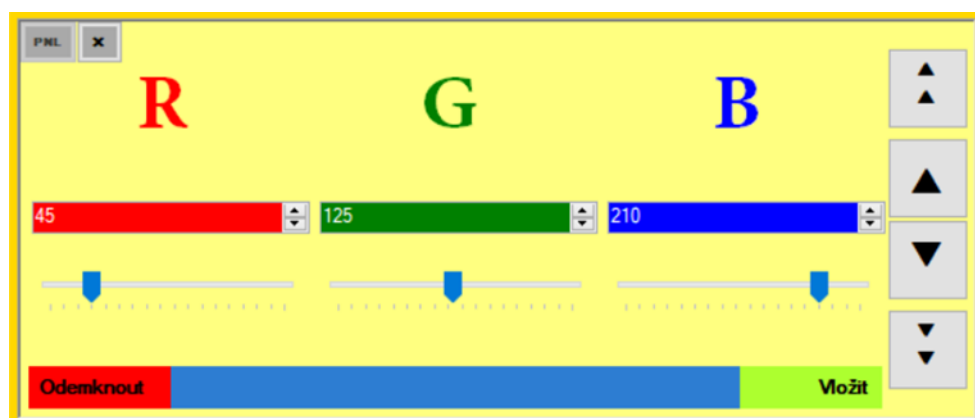
RGB panel disponuje míchacími prvky pro červenou, zelenou a modrou barvu, označené velkými písmeny ‚R‘, ‚G‘ a ‚B‘, které společně určují, jaký odstín barvy se namíchá. Odstín barvy závisí na hodnotách v numerických buňkách, do kterých se mohou přímo zadávat hodnoty jasu prvků R, G a B (v rozmezí od 0 až 255) a pod těmito buňkami jsou ovládací posuvníky pro rychlejší a jednodušší míchání odstínu.

Finální namíchaná barva, viz Obr. 3.19 dole, se poté zobrazuje ve spodu panelu na místě bílého proužku, viz Obr. 3.18. Tlačítko ‚Zamknout‘, vlevo od proužku, slouží k uzamknutí této barvy, aby se barva neměnila během vybírání řádku/ů. Spolu s tlačítkem ‚Vložit‘, vpravo od proužku, se může tato barva rychle kopírovat do jakékoli vybrané řady.

V pravé části RGB panelu je menší klávesnice, sloužící k pohybu v ANIM panelu. Tlačítka s jednou šipkou ovládají pozici označeného řádku v přehledovém panelu, tlačítka s dvěma šipkami slouží k pohybu mezi jednotlivými kroky animace.



Obr. 3.18 Ovládací panel RGB panel (základní nastavení)



Obr. 3.19 Ovládací panel RGB panel (využití v kroku animace)

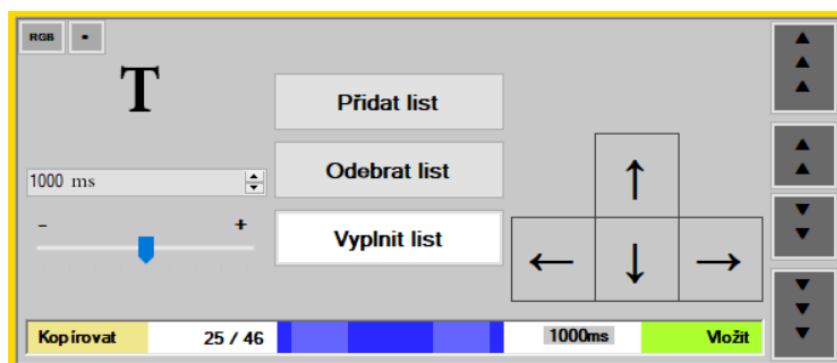
3.2.2 TIME panel

TIME panel je druhý ovládací panel Ealedu. Zaobírá se konfigurací samotného kroku animace, jako je nastavování času prodlevy mezi jednotlivými kroky animace, možnost vytvořit další krok animace nebo krok vymazat a poskytuje možnost měnit pořadí, ve kterém se budou kroky animace promítat.

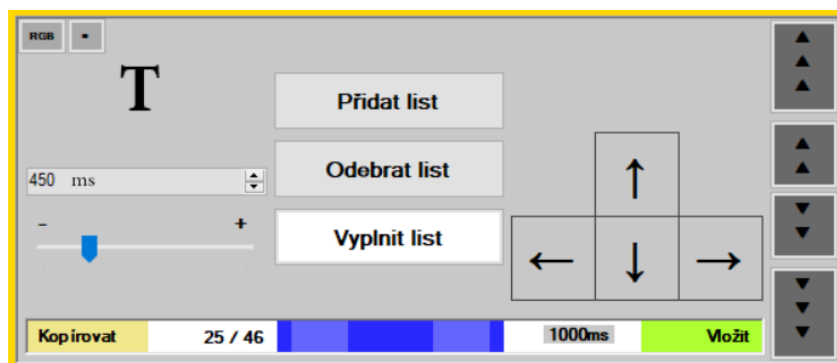
V levé části panelu, je samotný prvek ovládání času. Obdobně jako u RGB panelu, čas je označen velkým ‚T‘ a je ovládán číselnou buňkou a posuvníkem (v rozmezí 0 až 2000).

Uprostřed TIME panelu se nachází tři tlačítka. První tlačítko s nápisem ‚Přidat list‘ přidá další krok k animaci, se kterým se dále může pracovat. Druhé tlačítko s nápisem ‚Odebrat list‘ odebere vybraný list v LIST panelu. Poslední tlačítko s nápisem ‚Vyplnit list‘ slouží k zaplnění celého listu barev ve vybraném kroku animace jednou barvou. Tato barva závisí na barvě zamčené v RGB panelu.

Dále jsou zde dvě sady šipek. Obdobně jako u RGB panelu, pravá sada šipek slouží k přesouvání mezi kroky animace. Tlačítka s dvěma šípkami slouží k přesouvání mezi jednotlivými kroky animace, zatím co tlačítka s třemi šípkami slouží k přepínání mezi celými LIST panely, pokud animace má více než šedesát-čtyři kroků. Levá sada šipek slouží k přesouvání označeného listu v pořadí kroků animace.



Obr. 3.20 Ovládací panel TIME panel (základní nastavení)



Obr. 3.21 Ovládací panel TIME panel (využití v kroku animace)

3.3 Extra prvky

Mezi extra prvky patří panel DGV a Vzory animací.

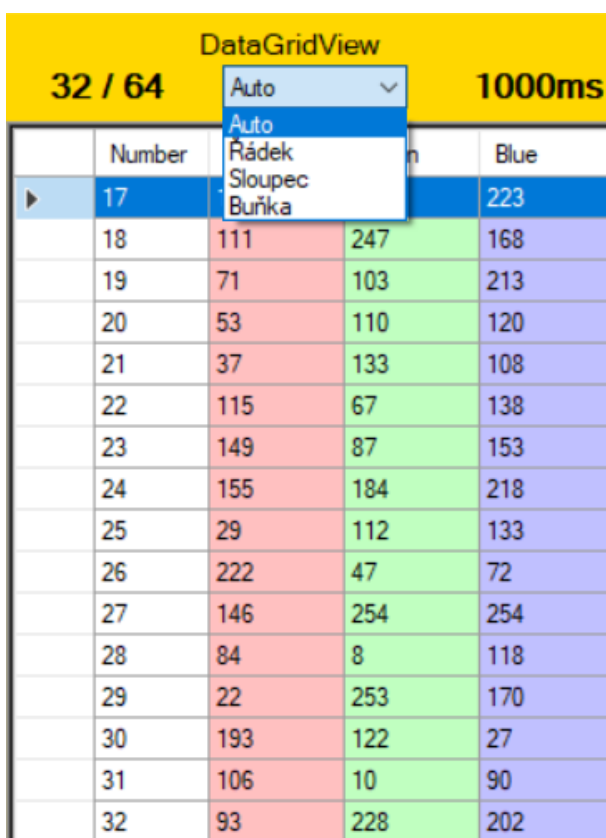
3.3.1 DGV

DGV (DataGridView) je nejdůležitější součástí Ealedu. Je to srdce mého programu, podle něhož se řídí veškeré komponenty Ealedu.

Jakmile se informace o barvách změní v DGV, ostatní panely se podle něho nastaví. Proto ovládací prvky mění barvu v DGV. DGV si totiž vždy vynutí, aby se ostatní panely jako přehledové panely měnily podle něj.

DGV ale neslouží jen pro konstantní přehled nad animací. Na rozdíl od přehledových panelů, DGV obsahuje přesná data o každém řádku, barvě, ale i informace o samotném kroku animace, jako je čas prodlevy a číslo, v jakém pořadí animace je tento krok. Dále DGV umožňuje měnit tato data přímo. Stačí pouze kliknout na buňku obsahující tato data a přepsat je.

DGV jinak obsahuje čtyři sloupce dat. Sloupec ‚Number‘ představuje číslo řádku v animaci, sloupec ‚Red‘ představuje odstín červené a stejně tak sloupce ‚Green‘ a ‚Blue‘ zobrazují odstíny zelené a modré.



32 / 64		Auto		1000ms	
	Number	Red	Green	Blue	
▶	17	111	247	168	223
	18	71	103	213	
	19	53	110	120	
	20	37	133	108	
	21	115	67	138	
	22	149	87	153	
	23	155	184	218	
	24	29	112	133	
	25	222	47	72	
	26	146	254	254	
	27	84	8	118	
	28	22	253	170	
	29	193	122	27	
	30	106	10	90	
	31	93	228	202	
	32				

Obr. 3.22 Extra panel DataGridView (příklad při vybraném kroku)

DataGridView				
1 / 64		Buňka ▾		1000ms
	Number	Red	Green	Blue
	1	97	225	125
	2	127	21	191
	3	41	105	5
	4	224	81	115
▶	5	251	97	208
	6	111	148	53
	7	223	32	68
	8	215	116	217
	9	12	1	97
	10	167	83	250
	11	93	152	209
	12	0	103	190
	13	183	107	60
	14	70	128	43
	15	171	14	142
	16	207	129	46

Obr. 3.23 Extra panelu DataGridView (ostatní prvky)

V levé horní části DGV se nacházejí dvě čísla. První číslo značí, jaký krok animace se zrovna zobrazuje, druhé číslo značí, kolik kroků celkem celá animace obsahuje.

V pravé horní části DGV je časový údaj. Tento údaj představuje prodlevu mezi tímto a dalším krokem v animaci. Čas prodlevy je udáván v milisekundách a může být v rozmezí 0 až 2000ms.

Uprostřed je rozklikávací seznam stylů vybírání, viz Obr. 3.24 a Obr. 3.22. V DGV také můžete vybírat více než jednu řadu barev. Díky třem stylům se může vybírat jakékoliv pořadí buňky barev a výběr se poté může editovat díky ovládacímu panelu RGB panel, který bude měnit data pouze této vybrané buňky.

DataGridView				
32 / 64		Auto ▾		1000ms
	Number	Red	Green	Blue
	17	122	244	223
	18	111	247	168

Obr. 3.24 Horní část DataGridView

- Styl Řádek: po kliknutí na jakoukoliv buňku se vybírá celá řádka.
- Styl Sloupec: po kliknutí na jakoukoliv buňku se vybere celý sloupec příslušné barvy.
- Styl Buňka: bude vybírat jen jednotlivé buňky.
- Styl Auto: mění automaticky styl vybírání podle toho, kam na DGV uživatel klikne. Např. uživatel klikne na hlavu řádku, Auto se bude od té chvíle chovat jako styl Řádek. Pak uživatel klikne na hlavu sloupce a Auto se změní na styl Sloupec. Aby se styl přepnul na styl Buňka, musí se na ni kliknout dvakrát.

The image shows four instances of a DataGridView control, each demonstrating a different selection style. Each window has a title bar with 'DataGridView' and a status bar with a range and a duration of '1000ms'. The selection styles are:

- Auto:** The selection menu is open, showing options: 'Auto', 'Řádek', 'Sloupec', and 'Buňka'. The grid shows a blue selection on the first row.
- Řádek:** The selection menu is set to 'Řádek'. The entire first row is highlighted in blue.
- Sloupec:** The selection menu is set to 'Sloupec'. The first column is highlighted in blue.
- Buňka:** The selection menu is set to 'Buňka'. Only the first cell (row 1, column 1) is highlighted in blue.

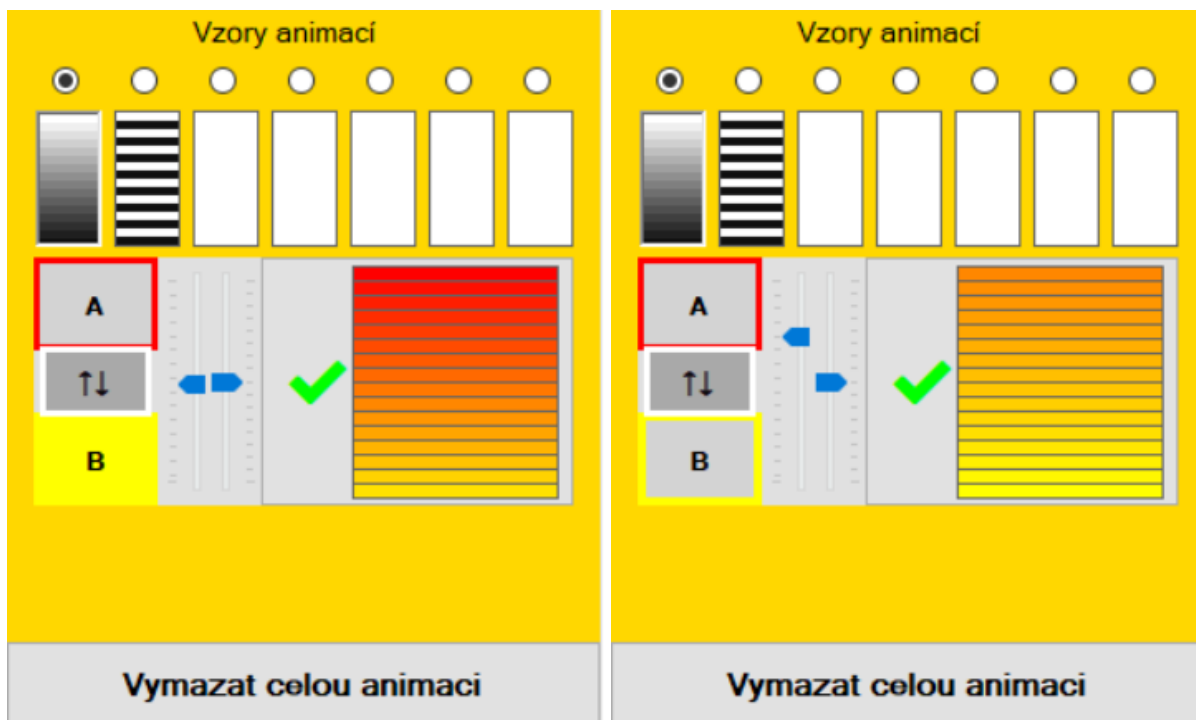
 All grids contain the same data:

Number	Red	Green	Blue
17	122	244	223
18	111	247	168
19	71	103	213
20	53	110	120
21	37	133	108
22	115	67	138
23	149	87	153
24	155	184	218
25	29	112	133
26	222	47	72
27	146	254	254
28	84	8	118
29	22	253	170
30	193	122	27
31	106	10	90
32	93	228	202

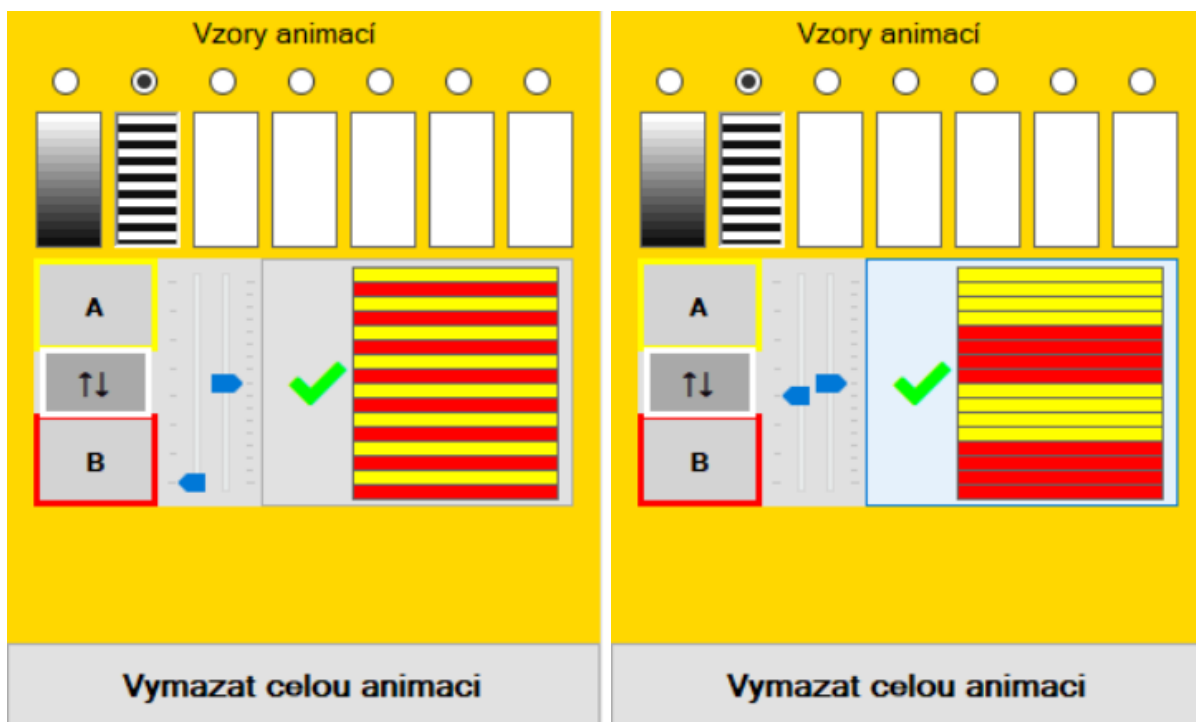
Obr. 3.25 Vybírání buněk v DataGridView

3.3.2 Vzory animací

Vzory animací umožňují vkládat celé přednastavené kroky animace najednou. Funguje to tak, že si uživatel vybere vzor animace v horní části a tento vzor si poté dále upraví v rozhraní nastavování vzoru.



Obr. 3.26 Extra panel Vzory animací (vzor animace 1)



Obr. 3.27 Extra panel Vzory animací (vzor animace 2)

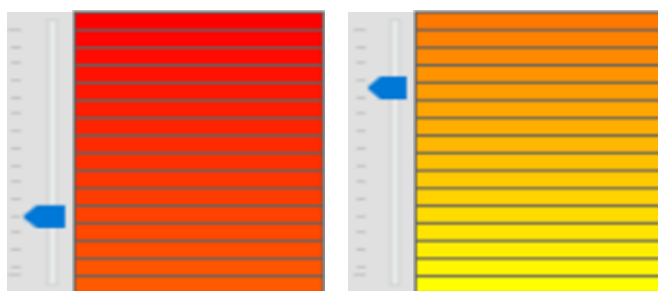
V současné době jsou hotové pouze dva vzory. První dostupný vzor animace je postupné prolínání dvou barev, kdy se jedna barva pozvolna přemění v druhou barvu. Druhým vzorem je střídání dvou barev. Tento vzor vytvoří střídající se pruhy vybraných barev, jejichž výška lze libovolně měnit posuvníky. Dalšími vzory animace, na kterých pracuji, bude vlnění, kde se vytvoří vlna jedné barvy s postupnou změnou jasu a tříbarevné pruhy.

Rozhraní nastavování vzoru pak obsahuje tři tlačítka k nastavení samotné barvy, dva ovládací posuvníky pro nastavení stylu vzoru animace a ukázkou, jak tento vzor animace bude vypadat.

V levé části rozhraní jsou tři tlačítka. Tlačítko s nápisem ‚A‘ značí primární barvu a tlačítko ‚B‘ značí sekundární barvu. Barva těchto tlačítek se počátečně automaticky nastaví podle horní a spodní řady přímo v DGV. Po kliknutí na tato tlačítka se jejich barva dá dále nastavit pomocí ovládacího prvku RGB panel. Třetí tlačítko s šipkami ‚↑↓‘ slouží k prohození barev těchto tlačítek.

Uprostřed rozhraní jsou dva posuvníky. Levý posuvník slouží k nastavení samotného stylu vzoru. Jeho efekt na vzor se mění podle toho, jaký vzor animace je právě použit. U prolínání funguje tak, že určuje, která ze dvou barev bude dominantní. U střídání barev, levý posuvník mění výšku barevné řady. Pravá dráha naopak pouze posouvá pozici vzoru.

V pravé části rozhraní je ukáзка, která zobrazuje, jak tento vzor ovlivní krok animace. Po stisknutí tlačítka ‚✓‘ označený krok animace nabere podobu vzoru animace, viz Obr. 3.32.



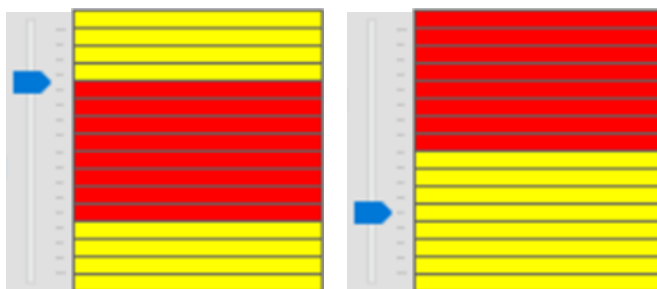
Obr. 3.28 Funkce Levé dráhy pro první vzor animace



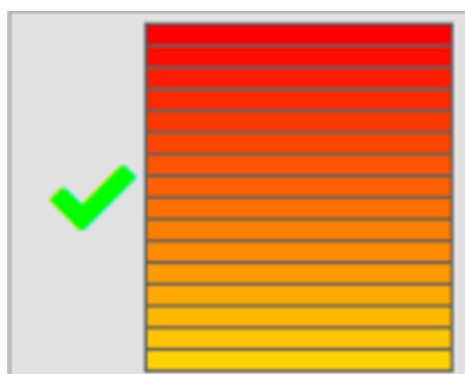
Obr. 3.29 Funkce Levé dráhy pro druhý vzor animace



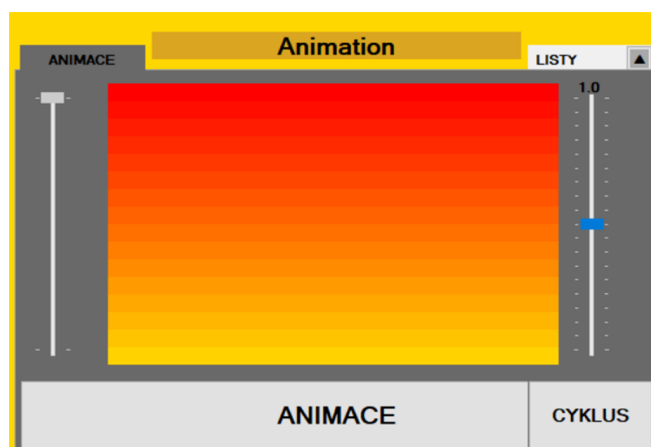
Obr. 3.30 Funkce Pravé dráhy pro první vzor animace



Obr. 3.31 Funkce Pravé dráhy pro druhý vzor animace



Obr. 3.32 Ukázka přenesení vzoru na krok animace



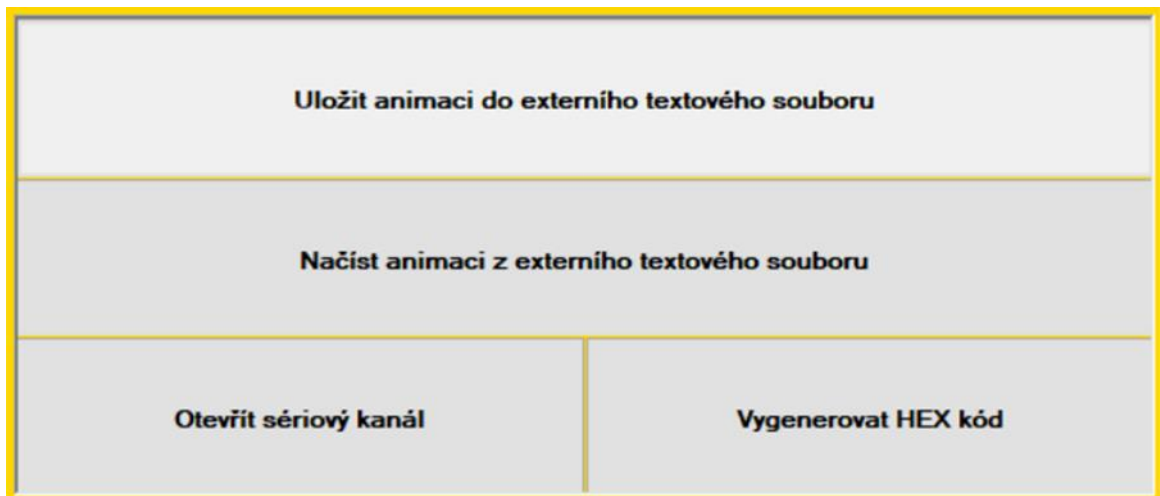
Obr. 3.33 Výsledný vzhled po přenesení vzoru na krok animace

3.4 Externí funkce

Externí funkce Ealedu jsou určeny pro ukládání animací. Složka ‚ColorSave‘ slouží pro ukládání animací Ealedu a pro opětovné načtení. Složka ‚HexCode‘ je přiřazena k samotnému přenosu přes sériový port.

3.4.1 Panel tlačítek externích funkcí

Odtud se spouští všechny externí funkce Ealedu, jako je ukládání a načítání datových souborů, sériová komunikace a generování hexadecimálního kódu. Celkem jsou zde čtyři tlačítka, viz obrázek Obr. 3.34.

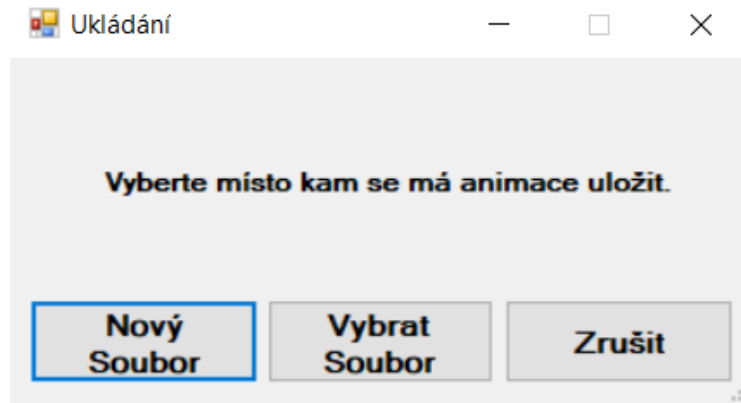


Obr. 3.34 Panel tlačítek externích funkcí

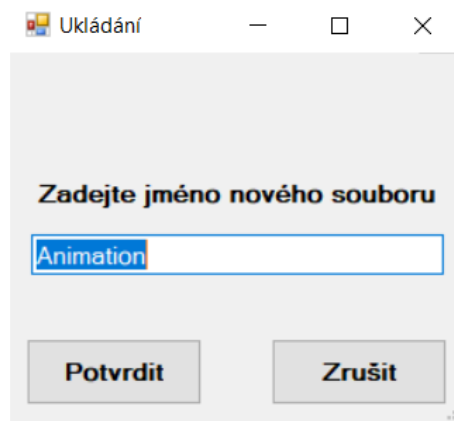
Horní tlačítko ‚Uložit animaci do externího textového souboru‘ umožňuje uložit vytvořenou animaci do externí složky. Po stisknutí tohoto tlačítka se otevře okno ‚Ukládání‘ s dotazem, kam se má animace uložit. Tlačítko ‚Zrušit‘ přeruší celý proces ukládání.

Tlačítko ‚Vybrat soubor‘ otevře výběrové okno a umožní tak uživateli přepsat již existující uloženou animaci. Tlačítko ‚Vytvořit nový soubor‘ otevře další okno ve kterém uživatel může upravit již předvyplněné jméno animace, které bylo zadáno v ANIM panelu.

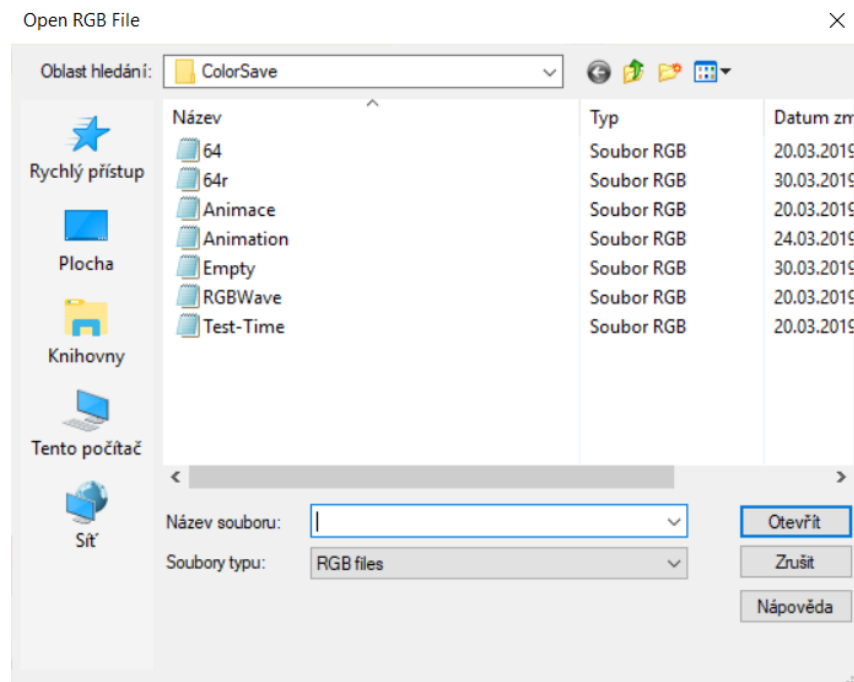
V tomto novém okně jsou další dvě tlačítka. Tlačítko ‚Potvrdit‘ poté vytvoří nový soubor .rgb ve složce ColorSave v souborech programu. Tlačítko ‚Zrušit‘ přeruší proces ukládání.



Obr. 3.35 Proces ukládání animace (dotaz k místu uložení)



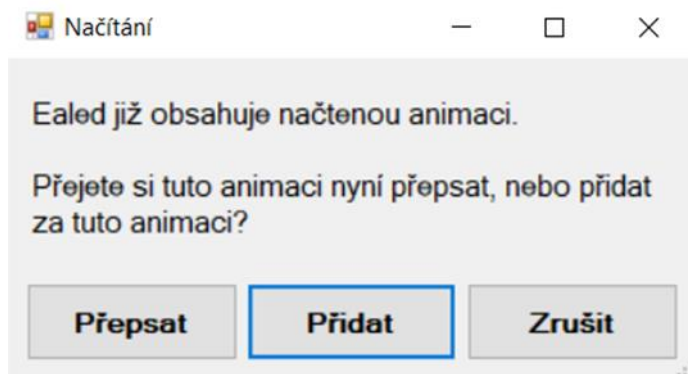
Obr. 3.36 Proces ukládání animace (dotaz k jménu souboru)



Obr. 3.37 Proces ukládání animace (výběr souboru k přeepsání)

Střední tlačítko na obrázku Obr. 3.34 ‚Načíst animaci z externího textového souboru‘ slouží k načítání animace z externích souborů Ealedu. Po kliknutí se okamžitě otevře výběrové okno stejné jako na Obr. 3.37 a po vybrání .rgb souboru se animace z tohoto souboru načte.

Pokud Ealed již obsahuje načtenou animaci, animace se nenačte ihned, ale zobrazí se dotazovací okno se třemi tlačítky a s otázkou, co si uživatel přeje se stávající animací udělat. Tlačítko ‚Zrušit‘ přeruší proces načítání. Tlačítko ‚Přepsat‘ vymaže stávající animaci v Ealedu a načte místo ní animaci ze souboru. Tlačítko ‚Přidat‘ načte animaci ze souboru a přidá jí za stávající animaci v Ealedu.

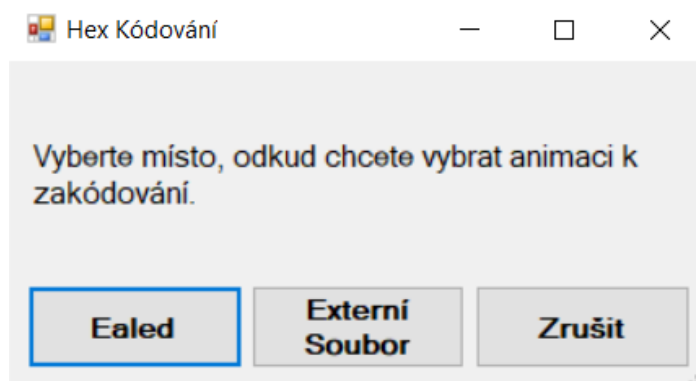


Obr. 3.38 Proces načítání animace (dotaz, pokud Ealed animaci obsahuje)

Dolní dvě tlačítka na obrázku Obr. 3.34 souvisí přímo se sériovou komunikací s LED panelem.

Pravé tlačítko ‚Vygenerovat HEX kód‘ slouží k zakódování animace do hexadecimální podoby, která se dá poslat přes sériový port do LED panelu. Po stisknutí se zobrazí dotazovací okno se třemi tlačítky a s otázkou, kterou animaci chce uživatel zakódovat. Tlačítko ‚Ealed‘ vybere animaci načtenou v Ealedu. Tlačítko ‚Externí Soubor‘ umožní uživateli vybrat již uloženou animaci z externí složky ‚ColorSave‘. Tlačítko ‚Zrušit‘ přeruší proces kódování.

Animace se uloží do složky ‚HexSave‘ v typu .hex. Levé tlačítko ‚Otevřít sériový kanál‘ slouží k zahájení sériové komunikaci s LED panelem.

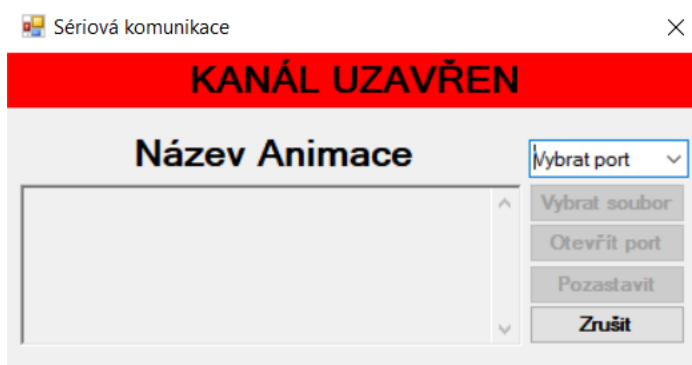


Obr. 3.39 Proces ukládání animace (dotaz odkud načíst animaci)

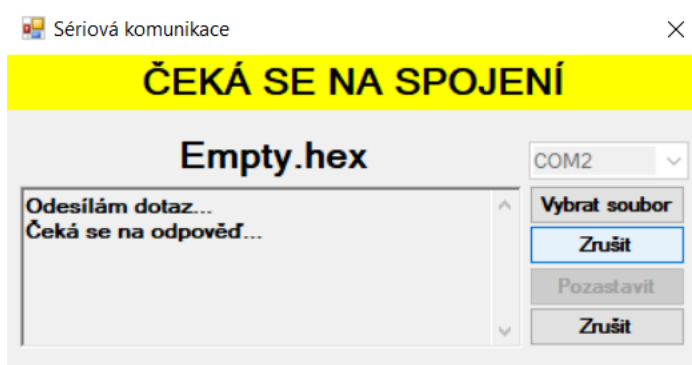
3.4.2 Sériová komunikace

Sériová komunikace slouží k samotné komunikaci mezi počítačem, ve kterém je Ealed spuštěn a mikroprocesorem v LED panelu. Po stisknutí tlačítka ‚Otevřít sériový kanál‘ v panelu tlačítek externích funkcí, se otevře samotné rozhraní k sériové komunikaci. Rozhraní disponuje všemi potřebnými prvky k zahájení sériové komunikace, jako je výběr komunikačního portu, výběr animace k přeposlání, tlačítka k otevření portu a zobrazení statusu portu.

Horní část rozhraní sériové komunikace slouží k zobrazení statusu otevření kanálu mezi LED panelem a Ealedem. Červená s nápisem ‚KANÁL UZAVŘEN‘ značí, že je kanál uzavřen a nepovoluje se žádná komunikace. Žlutá s nápisem ‚ČEKÁ SE NA PŘIPOJENÍ‘ znamená, že rozhraní sériové komunikace čeká na odezvu od mikroprocesoru. V tom případě se bude čekat po dobu deseti sekund a pak se kanál znovu uzavře. Zelená s nápisem ‚KANÁL OTEVŘEN‘ znamená, že kanál je otevřen a LED panel je připraven převzít animaci. Fialová s nápisem ‚KANÁL ODESÍLÁ‘ znamená, že animace je právě odesílána.



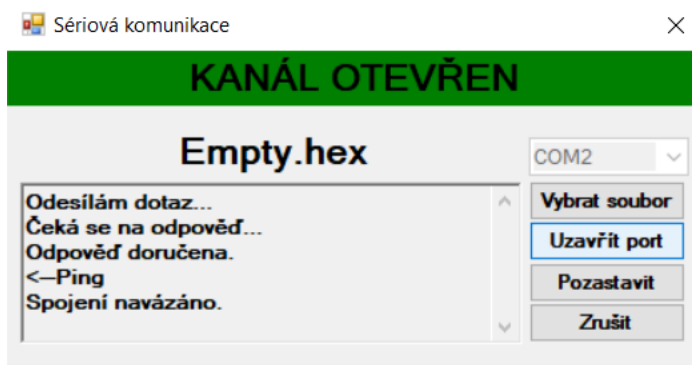
Obr. 3.40 Proces sériové komunikace (uzavřeno)



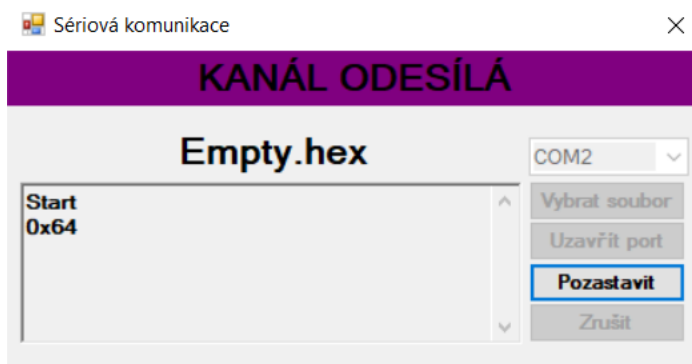
Obr. 3.41 Proces sériové komunikace (čeká)

Na druhé řádce rozhraní je text obsahující název animace, kterou si uživatel vybral k přeposlání do LED panelu. Pod tímto textem je konzole, která pravidelně vypisuje, co zrovna rozhraní dělá.

Vpravo rozhraní jsou prvky ovládající přenos přes sériový port. Horní bílá tabulka ‚Vybrat port‘ zobrazuje všechny dostupné sériové porty pro komunikaci. Tlačítko ‚Vybrat složku‘ slouží k vybírání animace. Po stisknutí tohoto tlačítka se otevře vyběrací okno ve složce ‚HexCode‘, kde jsou obsaženy animace ve formě .hex. Tlačítko ‚Otevřít port‘ se pokusí otevřít kanál mezi LED panelem a Ealedem a poté začne odesílat vybranou animaci. Tlačítko ‚Pozastavit‘ pozastaví odesílání.



Obr. 3.42 Proces sériové komunikace (otevřen)



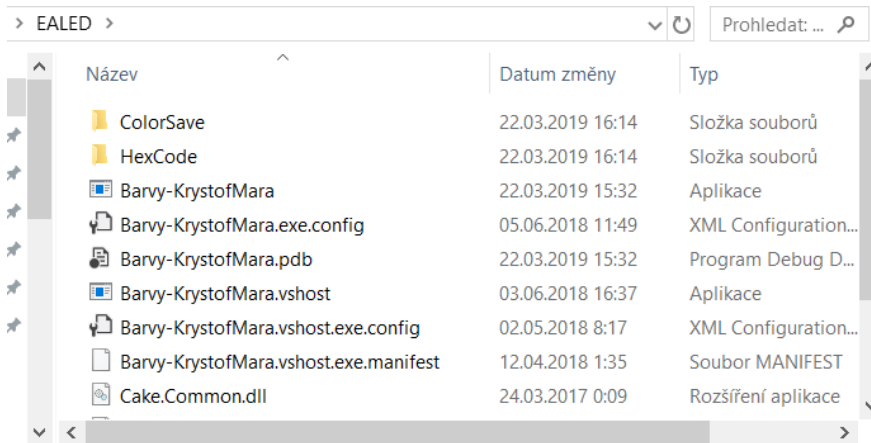
Obr. 3.43 Proces sériové komunikace (odesílání)

K úspěšné přípravě zaslání dat do LED panelu se musí dodržet pět kroků:

1. Před zahájením sériové komunikace s LED panelem musí být obě zařízení propojena komunikačním kanálem RS-252. Tento kanál umožňuje oboustrannou komunikaci mezi LED panelem a Ealedem.
2. Musí se vybrat port v zařízení, přes který se kanál otevře. K tomu slouží bílé okno v s nápisem ‚Vybrat Port‘, které umožňuje vybrat jeden z dostupných portů v zařízení
3. Vybere se animace ze složky ‚HexCode‘, která se má přeposlat pomocí tlačítka ‚Vybrat soubor‘.
4. Tlačítkem ‚Otevřít Port‘ se otevře kanál mezi Ealedem a LED panelem a Ealed otestuje, jestli spojení funguje.
5. Dále je sériová komunikace připravena a stačí dát povel tlačítkem ‚Odeslat‘.

3.4.3 ColorSave a HexCode

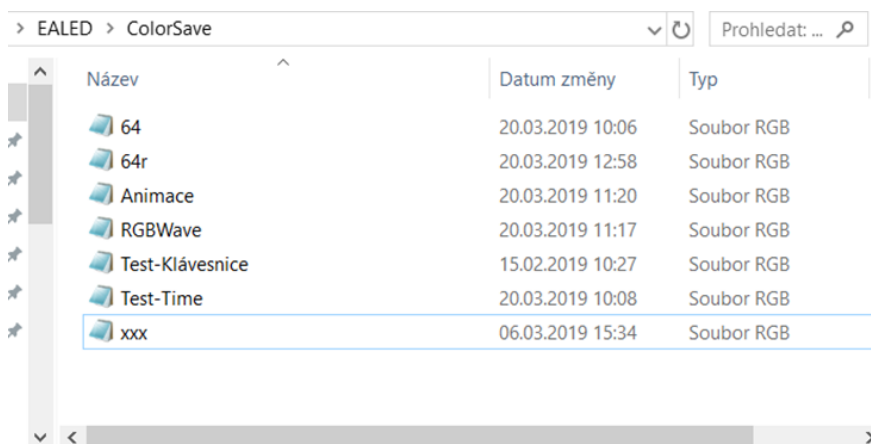
Samotné externí složky Ealedu ‚ColorSave‘ a ‚HexCode‘, mají také vlastní využití. Slouží převážně k ukládání, ale pokud si uživatel přeje uloženou animaci upravit bez použití Ealedu, je to možné.



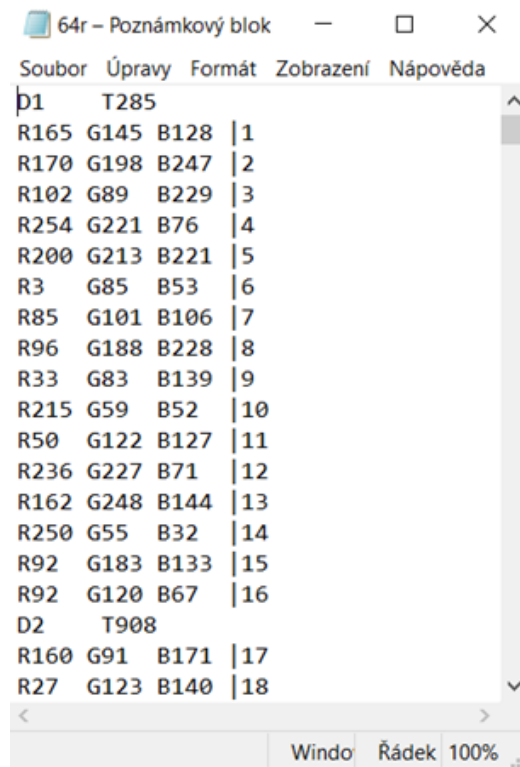
Obr. 3.44 Ukládací složky Ealedu

Složka ColorSave v sobě má uloženy všechny animace Ealedu. Každá animace je pojmenovaná podle toho, jaké jméno uživatel přiřadil této animaci a jsou uloženy v souboru typu .rgb, což je typ souboru, který Ealed dokáže přečíst a načíst z ní animace. Soubory typu .rgb, se ale mohou otevřít jednoduše jakýmkoliv textovým prohlížečem.

Soubor .rgb obsahuje data uložené animace, které Ealed čte řádek po řádku. Je zde informace o čísle kroku animace (D1, D2,... D25), barva a číslo řady v kroku animace, kde ‚R‘ značí jas červené a stejně tak ‚G‘ a ‚B‘ značí jas zelené a modré. A nakonec obsahuje údaj o délce prodlevy (T0, T1000,... T2000). Rozložení dat v souboru lze vidět na obrázku Obr. 3.46. Všechna tato data se dají změnit, ale pokud se nedodrží formát, animace se nemusí načíst správně.



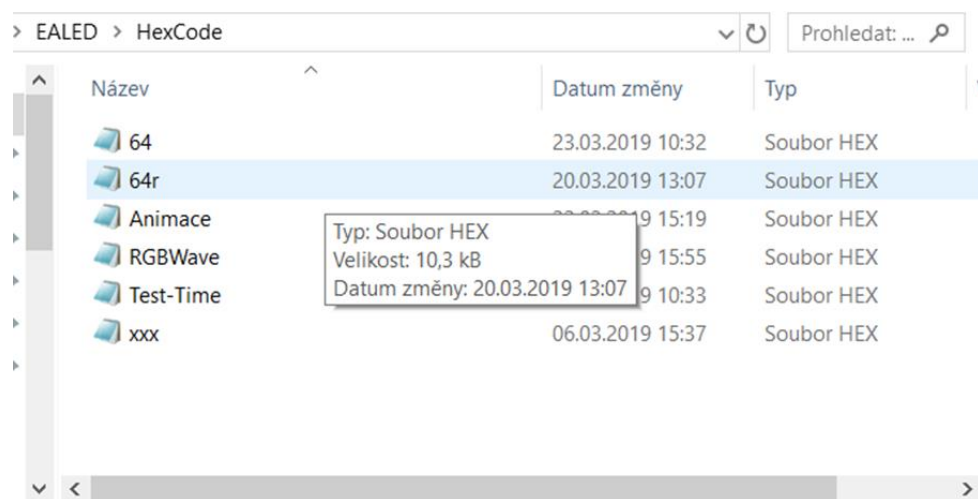
Obr. 3.45 Ukládací složka ColorSave



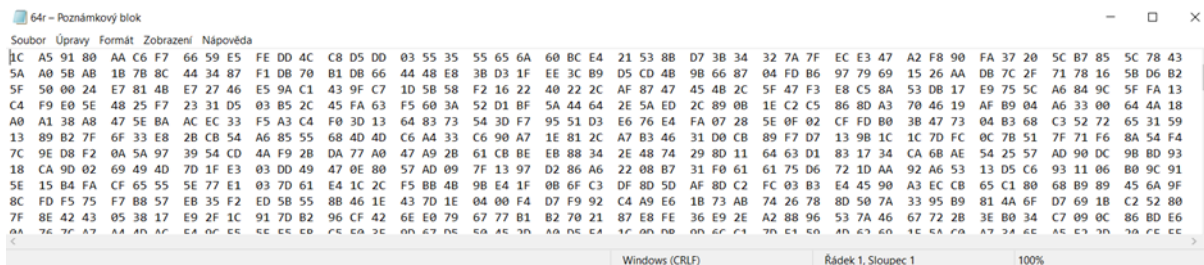
Obr. 3.46 Soubor .rgb ve složce ColorSave

Složka HexCode obsahuje soubory animace Ealedu, zakódované v hexadecimální podobě. Takto zakódované soubory mohou být odeslány do mikroprocesoru v LED panelu. Data v nich obsažená mohou být následně použita pro prezentaci vytvořené animace na LED panelu.

Soubory typu .hex je také možné otevřít jakýmkoli textovým prohlížečem, ale je mnohem těžší se v nich orientovat. Každá řada v .hex složce charakterizuje jeden krok animace. Na začátku řady jsou data o délce prodlevy (00, 9D,... FF) a dále tam jsou tři hexadecimální čísla s daty o jasu jednotlivých barev (00 00 00, 05 A8 EC,... FF FF FF).



Obr. 3.47 Ukládací složka HexCode



Obr. 3.48 Soubor .hex ve složce HexCode

4 ZÁVĚR

Prací na Ealedu jsem strávil hodně času. Při vytváření této práce jsem zjistil, že mě programování baví. Kdybych měl i více času, programování by určitě byl můj koníček.

Samotný Ealed je hotov. Poskytuje vše potřebné k vytvoření animace pro LED panel i pro přenos dat do LED panelu. Cíl práce byl tedy splněn. Program bude používán k návrhu světelných animací na VOŠ, SŠ, COP Sezimovo Ústí pro LED panel. Ten je určen pro prezentaci školy a náborové účely. Program lze využít i v hodinách fyziky k praktické ukázce vytváření různých barevných odstínů mícháním tří základních barevných složek.

Osobně si nemyslím, že můj program je něco světoborného. Ale na to, že Ealed je můj první opravdový program a mé znalosti programování byly na začátku malé, jsem s vytvořeným programem spokojen. Jsem také více než spokojen s tím, co jsem se při tvoření tohoto programu naučil.

Vytvořený program by se jistě dal ještě vylepšit. Stále obsahuje několik problémů s optimalizací, které by bylo vhodné odstranit. Zatím má pouze dva hotové vzory animací. Bylo by možné např. vytvořit vzor, který by vytvářel na LED panelu "vlny" nebo přechod celé plochy panelu z jednoho barevného odstínu plynule do jiného apod.

5 POUŽITÁ LITERATURA

- [1] Wikipedie, Java (programovací jazyk)
[https://cs.wikipedia.org/wiki/Java_\(programovac%C3%AD_jazyk\)](https://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk))
- [2] Obrázek z applitools.com/blog. Dostupné z: <https://applitools.com/blog/wp-content/uploads/2013/12/112866-java-eps-logo-99090.png>
- [3] Vyňato z wikipedie na adrese: <https://cs.wikipedia.org/wiki/C%2B%2B>
- [4] Wikipedie. Dostupné z:
https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO_C%2B%2B_Logo.svg/150px-ISO_C%2B%2B_Logo.svg.png
- [5] Vyňato z wikipedie na adrese: https://cs.wikipedia.org/wiki/C_Sharp
- [6] Imgix.net. Dostupné z: <https://pluralsight.imgix.net/paths/path-icons/csharp-e7b8fcd4ce.png>
- [7] Vyňato z wikipedie na adrese: https://cs.wikipedia.org/wiki/Microsoft_Visual_Studio
- [8] Wikipedie. Dostupné z:
https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Visual_Studio_2017_Logo.svg/1024px-Visual_Studio_2017_Logo.svg.png

6 SEZNAM OBRÁZKŮ A TABULEK

Obr. 2.1 Logo jazyka Java [2].....	7
Obr. 2.2 Logo jazyka C++ [4].....	7
Obr. 2.3 Logo jazyka C-Sharp (C#) [6]	8
Obr. 2.4 Logo programovacího rozhraní Visual Studio [8]	9
Obr. 3.1 Program EALED (jeden krok)	9
Obr. 3.2 Programu EALED (celá animace)	10
Obr. 3.3 Horní část přehledových panelů	11
Obr. 3.4 Horní část přehledových panelů	11
Obr. 3.5 Přehledový panel ANIM panel (s krokem animace).....	12
Obr. 3.6 Přehledový panelu ANIM panel (prázdný)	12
Obr. 3.7 Přehledový panel LIST panel (náhodně vyplněné listy)	13
Obr. 3.8 Přehledový panel LIST panel (nevyplněné listy).....	13
Obr. 3.9 List LIST panelu (vyplněný)	14
Obr. 3.10 List LIST panelu (prázdný).....	14
Obr. 3.11 Přehledový panel BOX panel	15
Obr. 3.12 Ovládací prvky RGB panelu.....	15
Obr. 3.13 Ovládací prvky TIME panelu	16
Obr. 3.14 Ukázka závislosti ovládacích prvků na přehledových prvcích (ANIM - RGB).....	16

Obr. 3.15 Ukázka závislosti ovládacích prvků na přehledových prvcích (BOX - TIME)	17
Obr. 3.16 Tlačítka k přepínání ovládacích prvků (TIME)	17
Obr. 3.17 Tlačítka k přepínání ovládacích prvků (RGB)	17
Obr. 3.18 Ovládací panel RGB panel (základní nastavení)	18
Obr. 3.19 Ovládací panel RGB panel (využití v kroku animace)	18
Obr. 3.20 Ovládací panel TIME panel (základní nastavení)	19
Obr. 3.21 Ovládací panel TIME panel (využití v kroku animace).....	19
Obr. 3.22 Extra panel DataGridView (příklad při vybraném kroku)	20
Obr. 3.23 Extra panelu DataGridView (ostatní prvky)	21
Obr. 3.24 Horní část DataGridView	21
Obr. 3.25 Vybírání buněk v DataGridView	22
Obr. 3.26 Extra panel Vzory animací (vzor animace 1).....	23
Obr. 3.27 Extra panel Vzory animací (vzor animace 2).....	23
Obr. 3.28 Funkce Levé dráhy pro první vzor animace.....	24
Obr. 3.29 Funkce Levé dráhy pro druhý vzor animace.....	24
Obr. 3.30 Funkce Pravé dráhy pro první vzor animace	25
Obr. 3.31 Funkce Pravé dráhy pro druhý vzor animace.....	25
Obr. 3.32 Ukázka přenesení vzoru na krok animace	25
Obr. 3.33 Výsledný vzhled po přenesení vzoru na krok animace	25
Obr. 3.34 Panel tlačítek externích funkcí.....	26
Obr. 3.35 Proces ukládání animace (dotaz k místu uložení)	27
Obr. 3.36 Proces ukládání animace (dotaz k jménu souboru).....	27
Obr. 3.37 Proces ukládání animace (výběr souboru k přepsání).....	27
Obr. 3.38 Proces načítání animace (dotaz, pokud Ealed animaci obsahuje)	28
Obr. 3.39 Proces ukládání animace (dotaz odkud načíst animaci).....	28
Obr. 3.40 Proces sériové komunikace (uzavřeno)	29
Obr. 3.41 Proces sériové komunikace (čeká).....	29
Obr. 3.42 Proces sériové komunikace (otevřen)	30
Obr. 3.43 Proces sériové komunikace (odesílání).....	30
Obr. 3.44 Ukládací složky Ealedu	31
Obr. 3.45 Ukládací složka ColorSave	31
Obr. 3.46 Soubor .rgb ve složce ColorSave	32
Obr. 3.47 Ukládací složka HexCode.....	32
Obr. 3.48 Soubor .hex ve složce HexCode	33

7 PŘÍLOHY NA CD

Příloha A Ealed – spustitelná verze programu