

Metronom

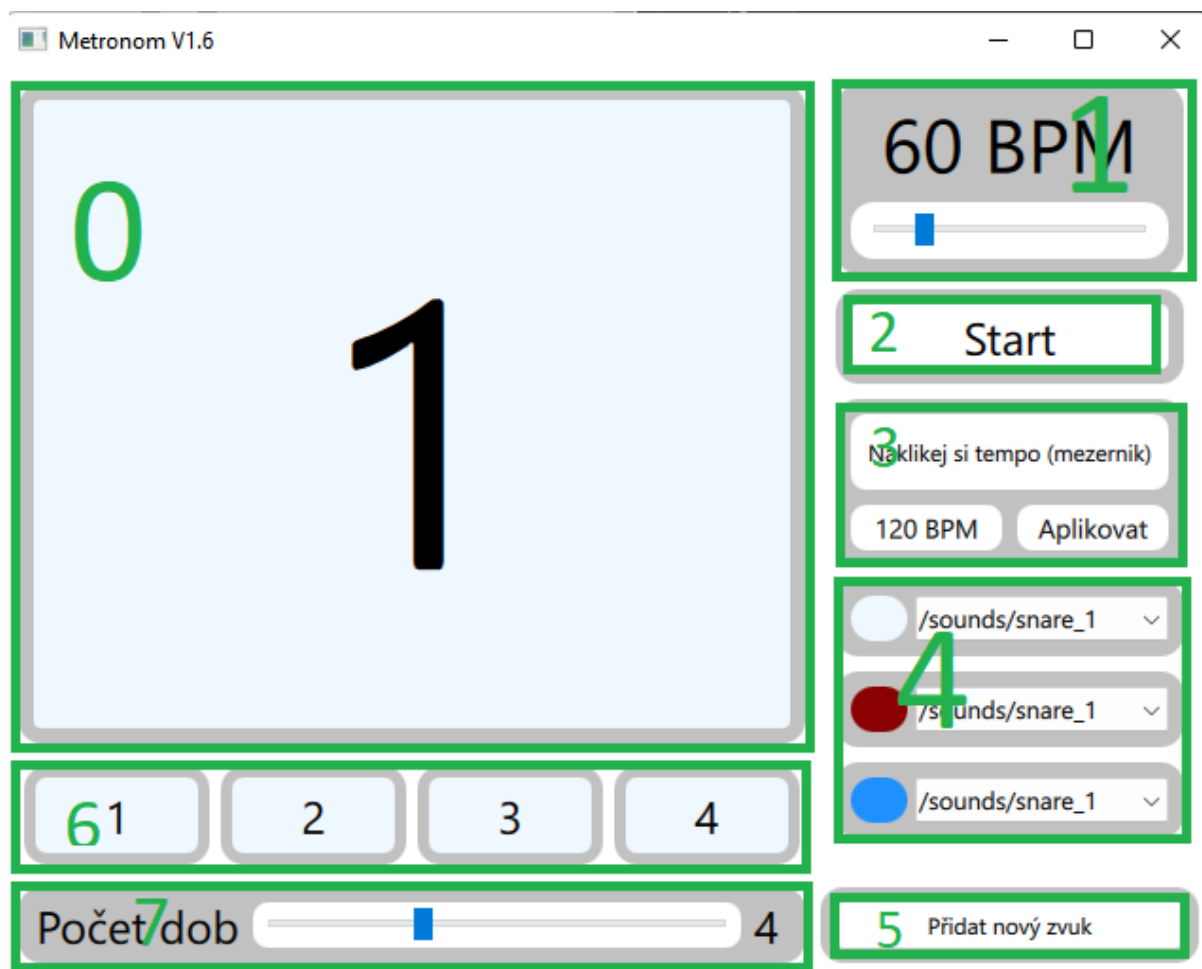
Motivace pro tvorbu metronomu

Moje první téma semestrální práce bylo propojení Home Assistantu s LoRaWAN, bohužel mi v raných fázích programování, během měnění konfigurace HW přestal fungovat jeden z modulů. Protože se mi nepodařilo včas sehnat náhradu a zároveň to nebyla taková ztráta času jelikož jsem teprve začínal s programování, rozhodl jsem se změnit téma a vybrat si jedno ze seznamu na CourseWare. Z těchto témat mi přišlo nejzajímavější právě metronom. Při hledání na internetu jsem zároveň zjistil, že je velice málo metronomů nabízí kompletní sadu funkcí. Zkusil jsem tedy naprogramovat právě takový program, který bude mít v sobě všechny možné funkce. Což se mi myslím povedlo až na jednu funkci a to možnost zvukové signalizace i na díly dob.

Seznam funkcí

1. Klasický metronom s rychlostí udávanou v BPM
2. Plná podpora dotykového ovládání
3. Možnost měnit počet dob po které se bude opakovat
4. Možnost si pro každou dobu nastavit jeden ze 3 párů barvy a zvuku
5. Tyto páry barva + zvuk lze libovolně nastavit a upravit
6. Možnost si "naklikat" vlastní tempo
7. Design je připravený pro různé velikosti zařízení

Popis ovládacích prvků (podle obrázku)



0. Ukazatel aktuální doby, není možné s ním interagovat.
1. Ukazatel tempa - zde se ukazuje aktuální tempo metronomu v jednotkách Beats Per Minute
2. Tlačítko pro zapnutí a vypnutí metronomu - nápis na tlačítku se mění podle stavu metronomu
3. Modul pro "naklikání si tempa" - pokud neznám přesnou hodnotu tempa v písničce, mohu si klikáním myši na tlačítko, či mačkáním **mezerníku** v tomto tempu zjistit přesnou rychlost.
Výpočet tempa probíhá pomocí měření času mezi kliknutím, poté jsou poslední 3 hodnoty jsou průměrovány podle následujícího vzorce: $2 \times \text{poslední hodnota} + \text{předposlední hodnota} + 0,5 \times \text{nejstarší ze 3 hodnot}$. Poté je tento průměr převeden na BPM a zaokrouhlen na celá čísla.
Tento modul má 3 části:
 - a. Tlačítko pro naklikání tempa
 - b. Ukazatel tempa ve kterém se aktuálně tlačítko (a) mačká
 - c. Tlačítko pro aplikování mého tempa do hlavních hodin
4. Sekce pro nastavení zvuku odpovídající jedné ze 3 možných barev indikátoru a změny barvy. Kliknutím na barvu se zobrazí vyskakovací okno, kde lze využít RGB posuvníků k nastavení libovolné barvy. Po kliknutí na OK se barva aplikuje. Pro změnu zvuku stačí kliknout na rozbalovací menu vedle barevného indikátoru, nebo

na něj najet myší a točit kolečkem.

Metronom obsahuje 3 barvy a k nim odpovídající zvuky mezi kterými je možné procházet v sekci 6.

5. Tlačítko pro přidání si vlastního zvuku. Stačí na něj kliknout a vybrat si jeden .wav soubor se zvukem, po stisknutí OPEN bude tento zvuk k dispozici v rozbalovacím menu
6. Tlačítka pro nastavení barev a zvuku jednotlivých dob. Kliknutím na určité tlačítko s číslem se barva a zvuk posune o 1 na pomyslném kole barev. Dohromady jsou 3 možnosti barev
7. Posuvník pro nastavení počtu dob. Je možné si zvolit 2 až 8 dob Po posunutí se metronom z důvodu předcházení chyb vrátí vždy na 1. dobu.

Vnitřní implementace programu

Program využívá knihovnu Qt.quick a je psaný v qml s použitím ECMAScriptových funkcí. Program se skládá z hlavního okna, které obsahuje veškeré funkční prvky, zároveň také umí vyvolat 2 dialogové okna pro nastavení určitých parametrů. Kvůli velké délce main.qml jsem se snažil program rozdělit na více částí. Né vždycky se mi to podařilo a například dialogové okno pro výběr barev je zakomponované uvnitř main kvůli problému s předáváním si proměnných mezi dialogem a hlavním programem. Povedlo se mi, ale například modul pro “naklikání” vlastního bpm a všechny dynamicky alokované prvky se tvoří z komponentů v samostatných souborech. Dynamickou alokaci jsem používal u prvků kde to dávalo smysl jak z hlediska délky kódu, tak z funkční stránky aplikace. V popisu ovládacích prvků jsou popsány 2 dynamické prvky. 1) sekce 4 - modul pro změnu barvy a zvuku. Zde dynamická alokace dávala smysl, protože každá dvojice zvuk a barva odpovídá jednomu prvku pole barev. Tím pádem se dá snadno generovat vše podle těchto indexů. Každý modul má v sobě i vlastní přehrávač zvuků, protože v případě změny přehrávaného zvuku trvá pár desítek milisekund než je možné nový zvuk použít, nebylo tedy možné využít jeden přehrávač pro všechny 3 páry.

V případě sekce 6 je dynamická alokace prakticky nutná pro správný chod programu, jelikož je potřeba prvky přidávat nebo odebírat podle toho kolik si uživatel zvolí dob. Zde jsem si dával pozor na to, aby při změně počtu prvků nedošlo ke změně nastavení na prvcích, kterých se to netýkalo (např. při změně z 8 dob na 4 zůstanou první 4 jako stejné objekty)

Díky použití Qt quick by mělo být relativně snadné aplikaci upravit pro využití na jiných platformách (např. telefon)

Možnosti dalšího vývoje

1. Připravit program na jiné zařízení než desktop
2. Využít přesnější časovač než QTimer
3. Přidat funkci zvukové signalizace mezi dobami
4. Vytvořit kompletně dynamický layout, který by se měnil v závislosti na rozlišení a DPI monitoru
5. Opravit chybu kde při zapnutém metronomu na velké BPM není možné použít mezerník k “naklikání” vlastního BPM (k opravě je zapotřebí vytvořit separátní proces pro “naklikávací” modul)