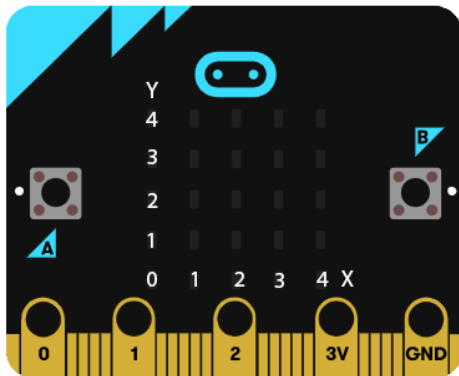


Název úlohy	LED mozaika
Třída	7. třída
Úloha splňuje RVP rámce	<ul style="list-style-type: none"> • ALGORITMIZACE A PROGRAMOVÁNÍ – řešení problému krokováním, programování, kontrola řešení
Propojení s RVP výstupy	<ul style="list-style-type: none"> • I-9-2-05 - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za něj; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné • I-9-2-06 - Žákyně/žák ověří správnost postupu, najde a opraví v něm případnou chybu
Propojení s ŠVP výstupy	<ul style="list-style-type: none"> • Žákyně/žák v blokově orientovaném programovacím jazyce sestaví přehledný program k vyřešení problému • Žákyně/žák používá podmínky pro ukončení opakování, rozezná, kdy je podmínka splněna
Časová náročnost	45 minut (jedna vyučovací hodina)
Stručný popis úlohy	Žáci si naprogramují micro:bit tak, aby mohli rozsvěcet libovolné LED na jeho poli.
Odkaz na rozšíření	https://github.com/microbit-cz/pxt-drawing-extension
Odkaz na řešení	https://github.com/microbit-cz/pxt-drawing-demo-hard

LED mozaika

Začátek

Ve variantě pro vyšší ročník budou mít žáci souřadnice již ve své vlastní režii. Nejzajímavější je to, jakým způsobem zajistit, že se kurzor přesune například na začátek řádku, pokud z řádku vyjedeme. Úkolem žáků tedy bude postarat se o „přetékání“ kurzoru. K tomu je potřeba pochopit souřadnice na microbitu, které jsou vidět na následujícím obrázku:



Co budete potřebovat

- PC s přístupem k [MakeCode](#)
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit

Rozšíření

Popis rozšíření

Překreslit bod	<ul style="list-style-type: none">• Překreslí aktuální bod• Bez parametrů• Bez návratové hodnoty
Vymazat kresbu	<ul style="list-style-type: none">• Vypne všechny LED (uvedení do výchozího stavu)• Bez parametrů• Bez návratové hodnoty
Přepnout kurzor	<ul style="list-style-type: none">• Zapne/vypne blikání kurzoru (pokud si bude někdo chtít kresbu prohlédnout, mohl by kurzor překážet)• Parametry:<ul style="list-style-type: none">○ Souřadnice X (číslo)○ Souřadnice Y (číslo)• Bez návratové hodnoty
Blikat kurzorem	<ul style="list-style-type: none">• Problikne kurzorem (umístí se do smyčky „opakuj stále“)

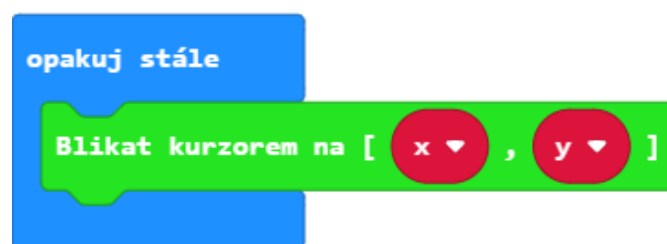
	<ul style="list-style-type: none"> • Parametry: <ul style="list-style-type: none"> ○ Souřadnice X (číslo) ○ Souřadnice Y (číslo) • Bez návratové hodnoty
Pohyb dolů/nahoru/doleva/doprava	<ul style="list-style-type: none"> • Pohne kurzorem na danou stranu • Parametry: <ul style="list-style-type: none"> ○ Souřadnice X (číslo) ○ Souřadnice Y (číslo) • Bez návratové hodnoty

Možný postup v úloze

1. Nejdříve si v bloku „při startu“ vytvoříme proměnné x a y, které v sobě budou obsahovat aktuální souřadnice.

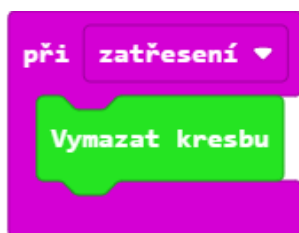


2. Nyní si vezmeme blok „opakuj stále“, ve kterém realizujeme blikání kurzoru.

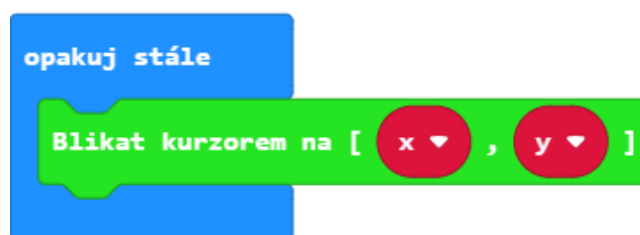


3. Nyní si zvolíme vstupy pro pohybování kurzorem, vypnutí/zapnutí kurzoru, vymazání kresby a přepnutí „pixelu“ (rozsvícení/zhasnutí LED).

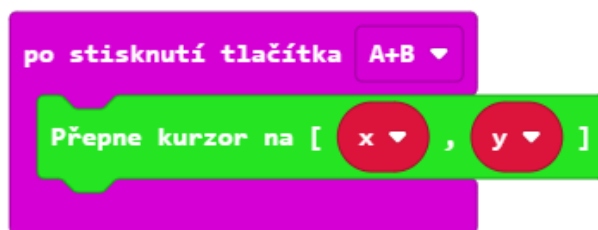
Smazání kresby:



Zapnutí/vypnutí blikání kurzoru:



Překreslení bodu (rozsvícení/zhasnutí):



4. Nyní jsou na řadě vstupy pro pohyb s kurzorem.

Nejdříve začneme volbou vstupů, které budeme používat („natahání“ do vývojového prostředí).

Učitel nechá žáky zamyslet se, jak se budou souřadnice měnit, když půjdeme doprava (x se zvýší o 1), doleva (x se sníží o 1), nahoru ($y + 1$) a dolů ($y - 1$). Když na tohle žáci přijdou, můžou toto naprogramovat.

V každém vstupu adekvátně upraví souřadnice a pak je vždy obě předají metodě *pohyb*. Při zkoušení programu by žáci měli přijít na to, že když vyjedou na konec pole a půjdou dále, kurzor zmizí. Zde by se učitel mohl zeptat, čím to je (odpověď by byla, že je souřadnice moc velká/malá => pokud jsme úplně vpravo a půjdeme opět doprava, souřadnice X bude 5, což je mimo pole). Dalším úkolem tedy bude zajistit, že pokud vyjedeme například z pole dolní stranou, kurzor se objeví na té horní.

Pro pohyb nahoru musíme zvýšit proměnnou Y o 1. Je ovšem Y 4, tak jsme na horní hraně a další pohyb nahoru by měl způsobit, že se kurzor přesune opět dolů. Tuto funkcionalitu

zajistíme jednoduchou podmínkou if(když). Y tedy zvýšíme o 1 a zkontrolujeme, že Y není větší než 4 (máme pole 5×5, ale vše číslujeme od nuly => 4 je maximum). Pokud je Y větší než 4 (jsme za hranou), tak Y nastavíme na 0 a tím pádem se vrátíme zpět dolů. Poté již můžeme zavolat blok z rozšíření.



Zbytek funguje na stejném principu. U pohybu vpravo, u kterého se zvyšuje X, je potřeba kontrolovat, zdali není X větší než 4. Pokud je, tak se nastaví X na 0.



Pohyb dolů funguje podobně, jako pohyb nahoru, ale mezní případ zde je, když je Y menší než 0. V tomto případě musí kurzor skočit nahoru a tím pádem nastavíme Y na 4.



No a u pohybu vlevo dojde k přeskočení na opačnou stranu v případě, že je X menší než 0. Když se toto stane, tak se X nastaví na 4. Poté opět můžeme volat metodu pro pohyb.

