

# Fotbalový zápas

Autor: Ondřej Kořínek

|                                   |  |
|-----------------------------------|--|
| Oblast z RVP<br>Doporučený stupeň | Algoritmizace a programování<br>2. stupeň ZŠ   |
| Naplněvané výstupy RVP            | <ul style="list-style-type: none"><li>• (I-9-2-01) po přečtení jednotlivých kroků algoritmu nebo programu vysvětlí celý postup; určí problém, který je daným algoritmem řešen</li><li>• (I-9-2-02) rozdělí problém na jednotlivě řešitelné části a navrhne a popíše kroky k jejich řešení</li><li>• (I-9-2-03) vybere z více možností vhodný algoritmus pro řešení problém a svůj výběr zdůvodní; upraví daný algoritmus pro jiné problémy, navrhne různé algoritmy pro řešení problému</li><li>• (I-9-2-05) v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné</li><li>• (I-9-2-06) ověří správnost postupu, najde a opraví v něm případnou chybu</li></ul> |
| Vstupní požadavky na žáky         | <ul style="list-style-type: none"><li>• Žák umí pracovat běžným způsobem s prohlížečem, orientuje se na webové stránce a umí vyhledat a zvolit možnost podle požadavků</li><li>• Žák ví, co je algoritmus, umí sestavit algoritmus, resp. vývojový diagram podle požadavků zadání.</li><li>• Žák používá opakování a větvení programu</li></ul>  |
| Cíl aktivity                      | <ul style="list-style-type: none"><li>• Žák se orientuje v novém prostředí v blokově orientovaném programovacím jazyce</li><li>• Žák ze zadání úlohy ověří, jestli potřebuje využít všechny bloky v daném blokově orientovaném programovacím jazyce</li><li>• Žák navrhne/vytvoří algoritmus pomocí vývojového diagramu ze zadání</li><li>• Žák z vývojového diagramu sestaví program v blokově orientovaném programovacím jazyce</li><li>• Žák srovná mezi sebou různé vývojové diagramy a sdělí, v čem se jednotlivá řešení liší</li><li>• Žák využívá v návrhu vývojového diagramu základní řídicí konstrukce – sekvenci, větvení a cyklus s neznámým počtem opakování</li></ul>  |
| Rozvíjené kompetence              | Klíčové kompetence prolínající se s rozvojem digitální gramotnosti žáků:<br><b>Kompetence k učení</b> <ul style="list-style-type: none"><li>• samostatně zpracovává zadaný projekt</li><li>• hodnotí vlastní řešení s předloženým řešením a navrhuje další možné zlepšování řešení</li><li>• využívá různé zdroje informací (algoritmus, vývojový diagram, řídicí konstrukce)</li></ul> <b>Kompetence k řešení problémů</b>  |

|                   |   |
|-------------------|---|
|                   | <ul style="list-style-type: none"> <li>• navrhuje algoritmus ze zadání</li> <li>• zjišťuje, že návrh algoritmu vede ke správnějšímu sestavení programu</li> <li>• rozpoznává chyby v navrženém algoritmu</li> <li>• identifikuje, s kterými kroky algoritmu by mohl být problém</li> <li>• aplikuje sestavený algoritmus v návrh programu</li> </ul> <p><b>Kompetence komunikativní</b></p> <ul style="list-style-type: none"> <li>• vysvětlí navržený algoritmus a graficky znázorní pomocí vývojového diagramu</li> </ul> <p><b>Kompetence sociální a personální</b></p> <ul style="list-style-type: none"> <li>• při vytvoření vývojového diagramu spolupracuje s ostatními žáky ve skupině</li> </ul> |
| Potřebné vybavení | PC, internet, webový prohlížeč, případně papír a tužka<br>K blokově orientovanému prostředí je využit projekt ze stránky <a href="https://code.org/">https://code.org/</a><br>Vývojové diagramy lze online vytvořit např. na stránce <a href="https://online.visual-paradigm.com/diagrams/features/flowchart-tool/">https://online.visual-paradigm.com/diagrams/features/flowchart-tool/</a> (jako konkrétně v tomto příspěvku)   |
| Časová dotace     | 4 vyučovací hodiny  |

### Průběh výuky

Příspěvek uvádí možnost uvedení do problematiky v prostředí blokově orientovaném programovacím jazyku. Žáci by již měli mít zkušenosti s vytvářením např. vývojových diagramů nebo by již mohli mít zkušenosti s daným blokovým prostředím, ale to není úplně nutné. Žáci by se případně měli v daném prostředí lehce zorientovat sami. Daný příklad má za cíl procvičit následující řídicí konstrukce: sekvenci, větvení a cyklus s neznámým počtem opakování. Další informace pro vyučujícího, zadání pro žáky a ukázky řešení jsou uvedeny dále v příspěvku.

**Zadání:** Vytvořte program „Fotbalový zápas“ v prostředí Code.org®, které naleznete na stránce <https://code.org/> v sekci Vytvoř: Ukázat všechny projekty. Dále v sekci Začít nový projekt zvolte View full list a vyberte správný projekt. Který to bude?

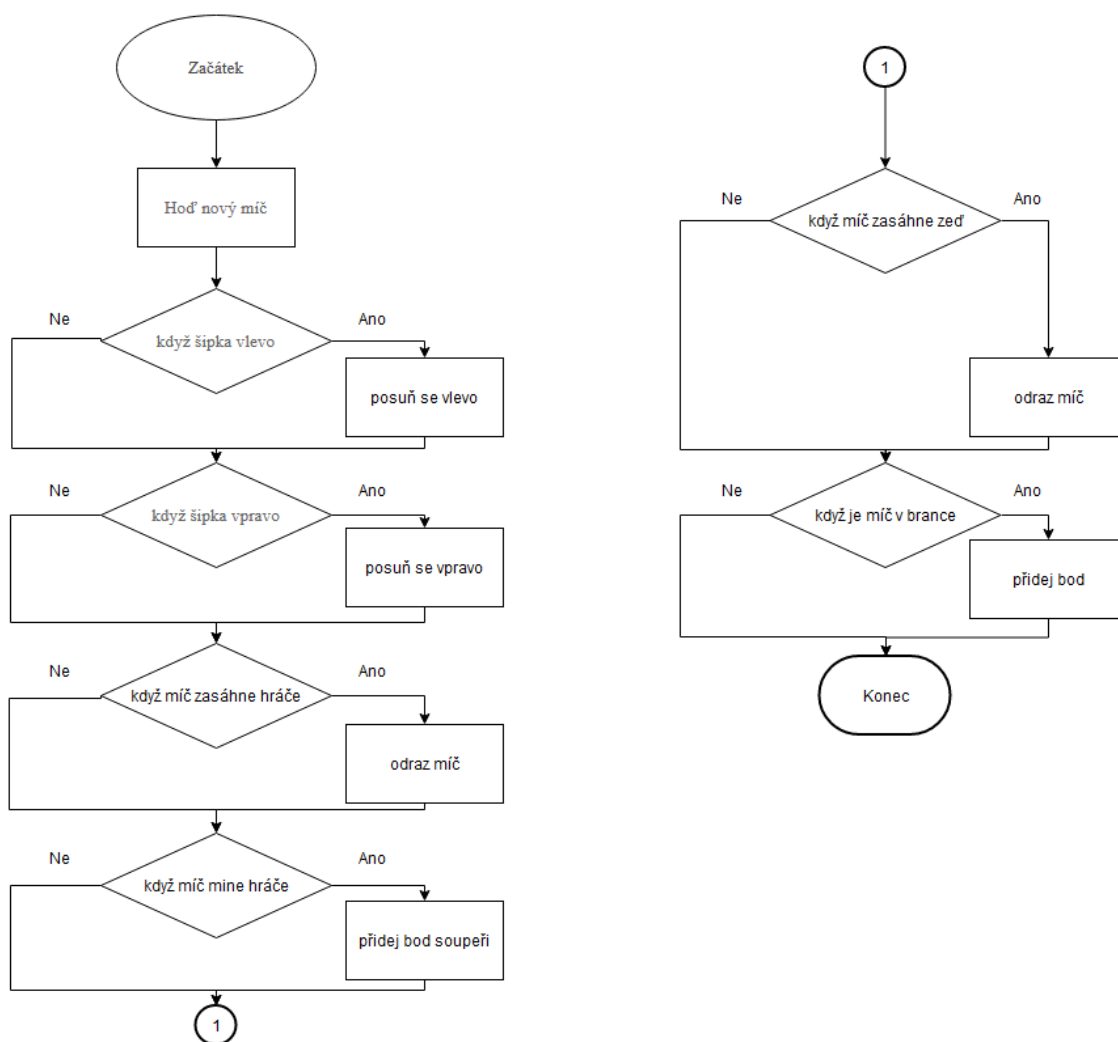
### Požadavky na „Fotbalový zápas“

- Hráč se bude moci ovládat šipkami vlevo a vpravo.
- Míč se bude odrážet od zdi a od hráče.
- Pokud míč skončí v bráně, připíše se bod hráči.
- Pokud hráč mine míč, bod se připíše protihráči.
- Rychlost hráče i míče bude náhodná.
- Pokud některý z hráčů získá bod, tak se v zápasu pokračuje. Počet gólů v zápasu není omezen.
- Dobrovolný úkol: Přidejte příslušné zvukové efekty.

### Otázky k „Fotbalovému zápasu“, v závorce jsou uvedeny označené výstupy:

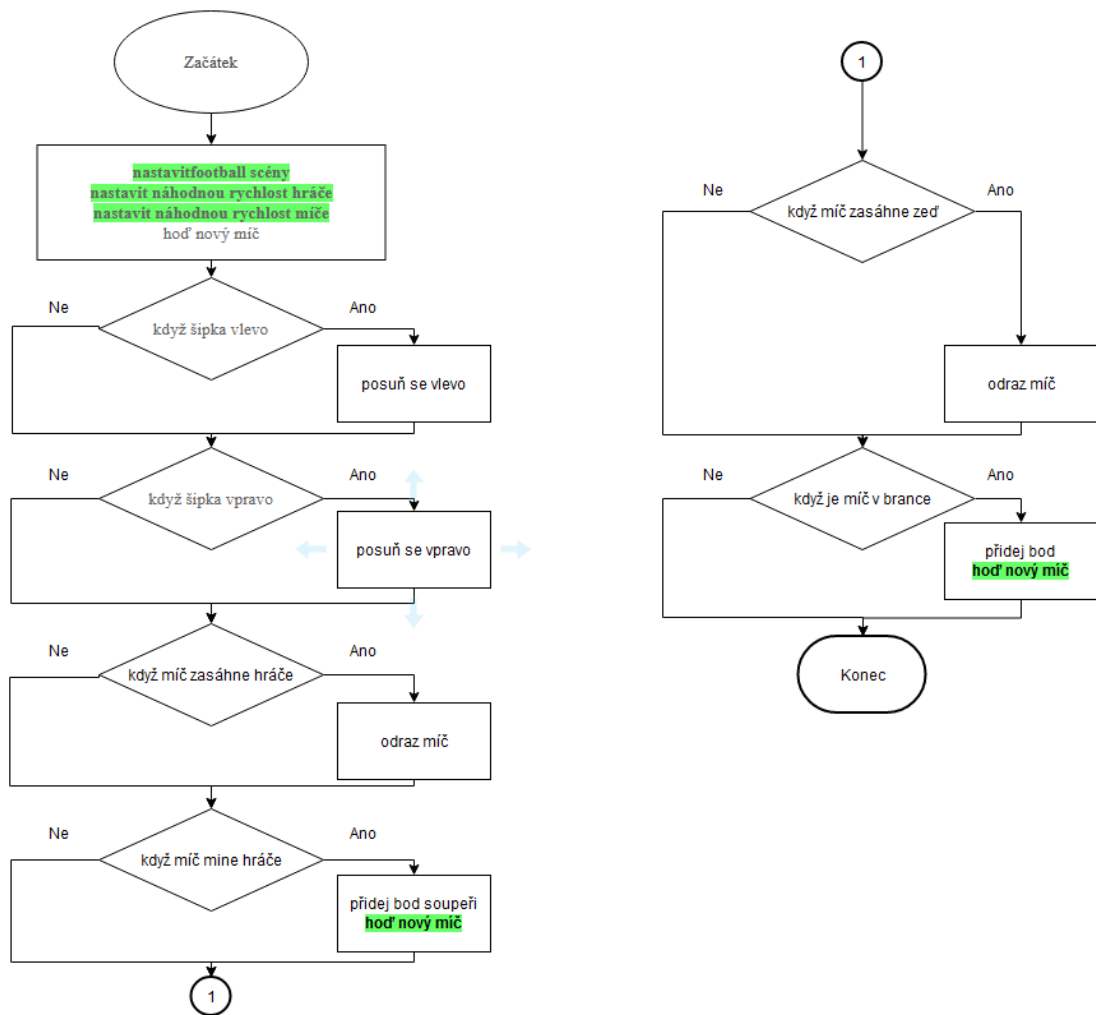
1. Na základě požadavků bude nutné použít všechny „možnosti“, které jsou v Pracovním prostoru? Pokud ne, které „možnosti“ nebudou potřeba? (I-9-2-01)

- Podle požadavků na „Fotbalový zápas“ se pokuste nakreslit vývojový diagram a podle tohoto vývojového diagramu sestavte pomocí příslušných bloků funkční projekt v prostředí Code.org. (I-9-2-02, I-9-2-05)
- Níže je uvedený obrázek 1 vývojového diagramu, který by měl odpovídat požadavkům zápasu. Jakým způsobem se liší vaše řešení z předchozího bodu od níže navrženého? Je daným algoritmem znázorněným vývojovým diagramem řešen problém zápasu, tj. jeho požadavky? Pokud ne, v čem je v daném vývojovém diagramu problém? (I-9-2-01, I-9-2-03, I-9-2-06)



Obrázek 1 – Vývojový diagram 1

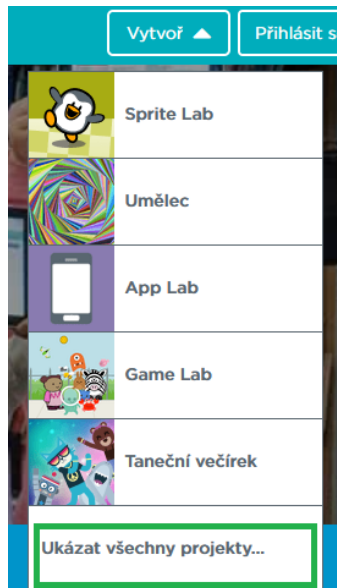
- Na obrázku 2 je uveden vývojový diagram, který vylepšuje předchozí verzi. Splňuje druhý vývojový diagram požadavky fotbalového zápasu, tj. je již správně řešen problém zápasu? Změny oproti předchozímu obrázku jsou vyznačeny zeleně. (I-9-2-03, I-9-2-06)
- Nakreslete vývojový diagram, který by řešil beze zbytku požadavky fotbalového zápasu. Jakou další řídicí konstrukci je nutné do vývojového diagramu přidat a proč? Neuvažujeme, že bychom museli stále ošetřovat ovládání (šipka vlevo a šipka vpravo) a zasažení míče. Tento úkol je náročnější, může být vyřešen žáky při skupinové práci ve skupinkách. (I-9-2-03)



Obrázek 2 - Vývojový diagram 2

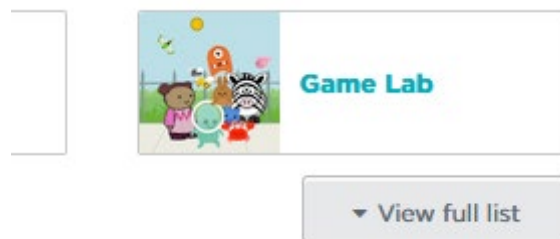
### Příprava na výuku

- Všechny úkoly je možné si vyzkoušet předem na stránce <https://code.org/>, kde se zvolí možnost Vytvoř: Ukázat všechny projekty, viz obrázek 3.



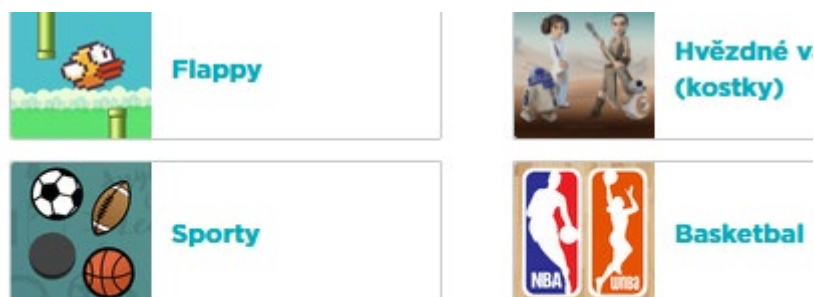
Obrázek 3 – Vytvoření projektu

- Na nové stránce se zvolí možnost View full list, viz obrázek 4.



Obrázek 4 – Zvolení možnosti View full list

- Vybere se kategorie Hry s událostmi s možností Sporty, viz obrázek 5.

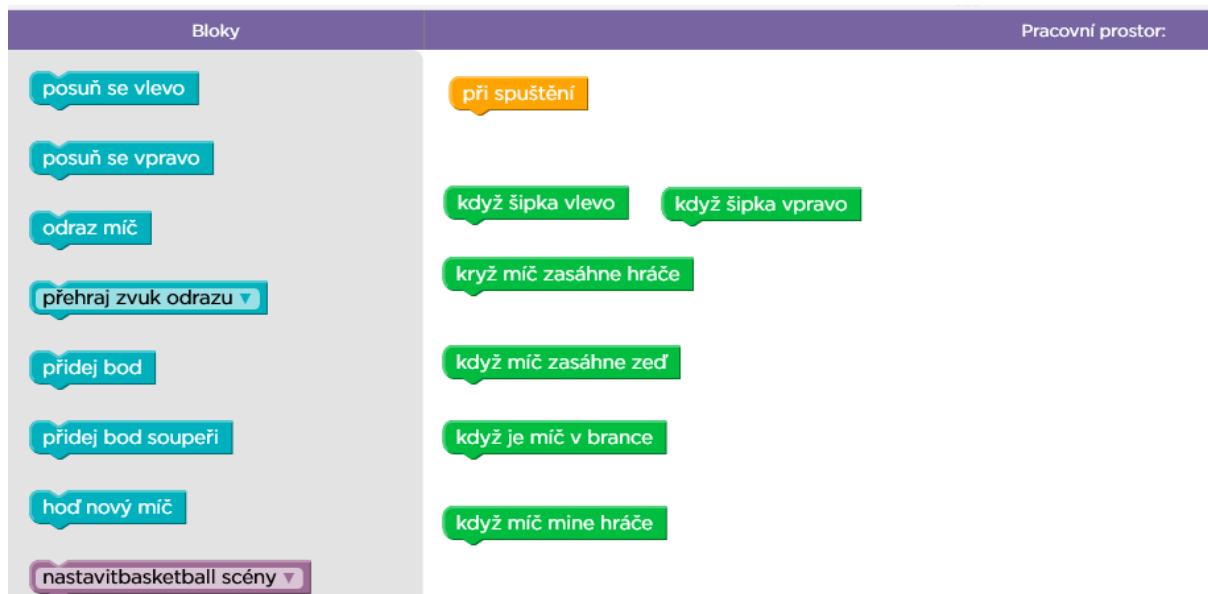


Obrázek 5 – Vybrání projektu Sporty

### Organizace práce

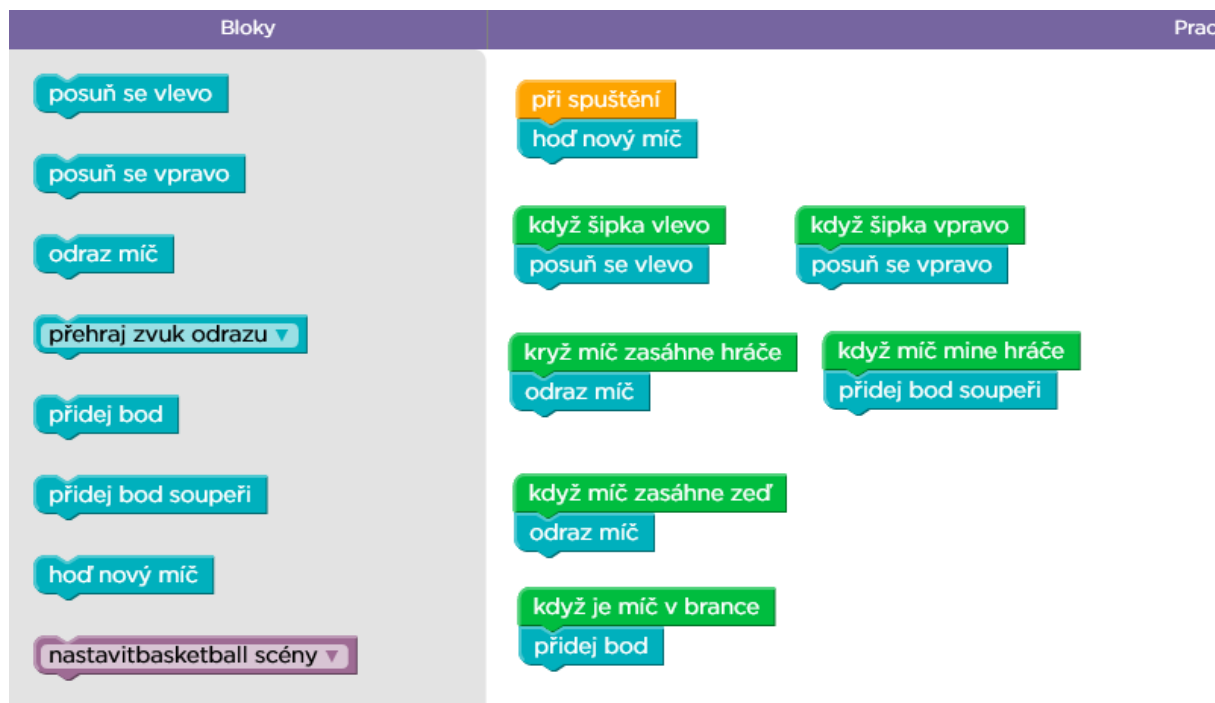
1. Žáci si podle návodu zkusí samostatně otevřít projekt. Po přečtení a pochopení zadání se budou snažit odpovědět na první otázku v předcházející kapitole Průběh výuky v podkapitole **Otázky k „Fotbalovému zápasu“**. Základní ukázka projektu po jeho vybrání je na obrázku 6. Správná

odpověď na zadanou otázku je, že je potřeba použít všechny možnosti (zelené bloky v obrázku 6 vpravo).



Obrázek 6 – Výřez prázdného projektu Sporty

2. Žáci by měli nejdříve navrhnout řešení pomocí vývojového diagramu, případně mohou obdržet Vývojový diagram 1 a podle něho mohou zkusit sestavit program. Řešení obrázku 1 v projektu Sporty je uvedeno na obrázku 7.

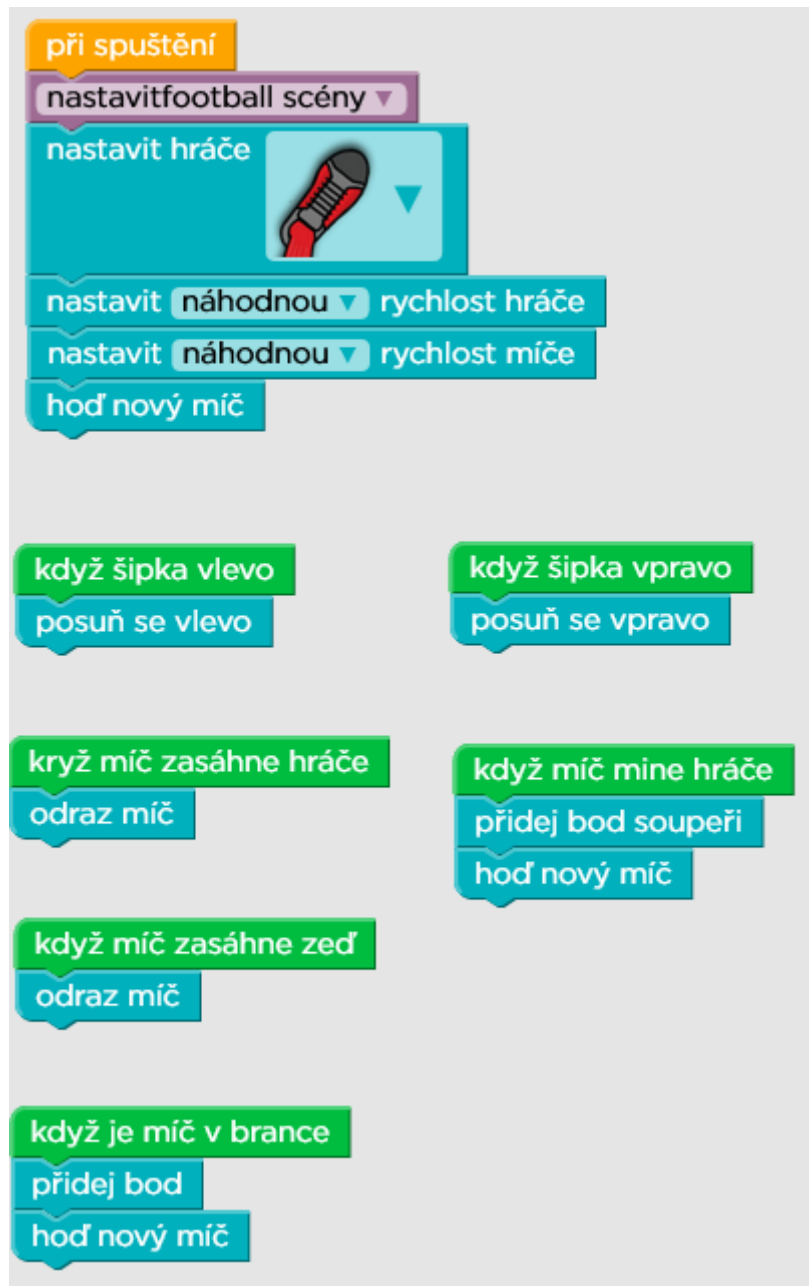


Obrázek 7 – Řešení vývojového diagramu 1 v prostředí Code.org v projektu Sporty

3. Pokud žáci dostali k dispozici v předchozím bodě vývojový diagram, tak v tomto bodu mohou odpovědět na otázky, jestli sestavený program odpovídá požadavkům zápasu.

**Správné odpovědi:**

- V daném vývojovém programu chybí správné vybrání prostředí scény
- Chybí správné nastavení hráče
- Chybí správné nastavení rychlosti míče i hráče
- V možnostech **když míč mine hráče** a **když je míč v brance** chybí, **hod' nový míč**, aby se program stále opakoval. Tyto možnosti mohou zkusit žáci doplnit samostatně a potom si zkontrolovat s řešením na obrázku 2 – vývojový diagram 2. Řešení tohoto bodu 3 je uvedeno na obrázku 8.



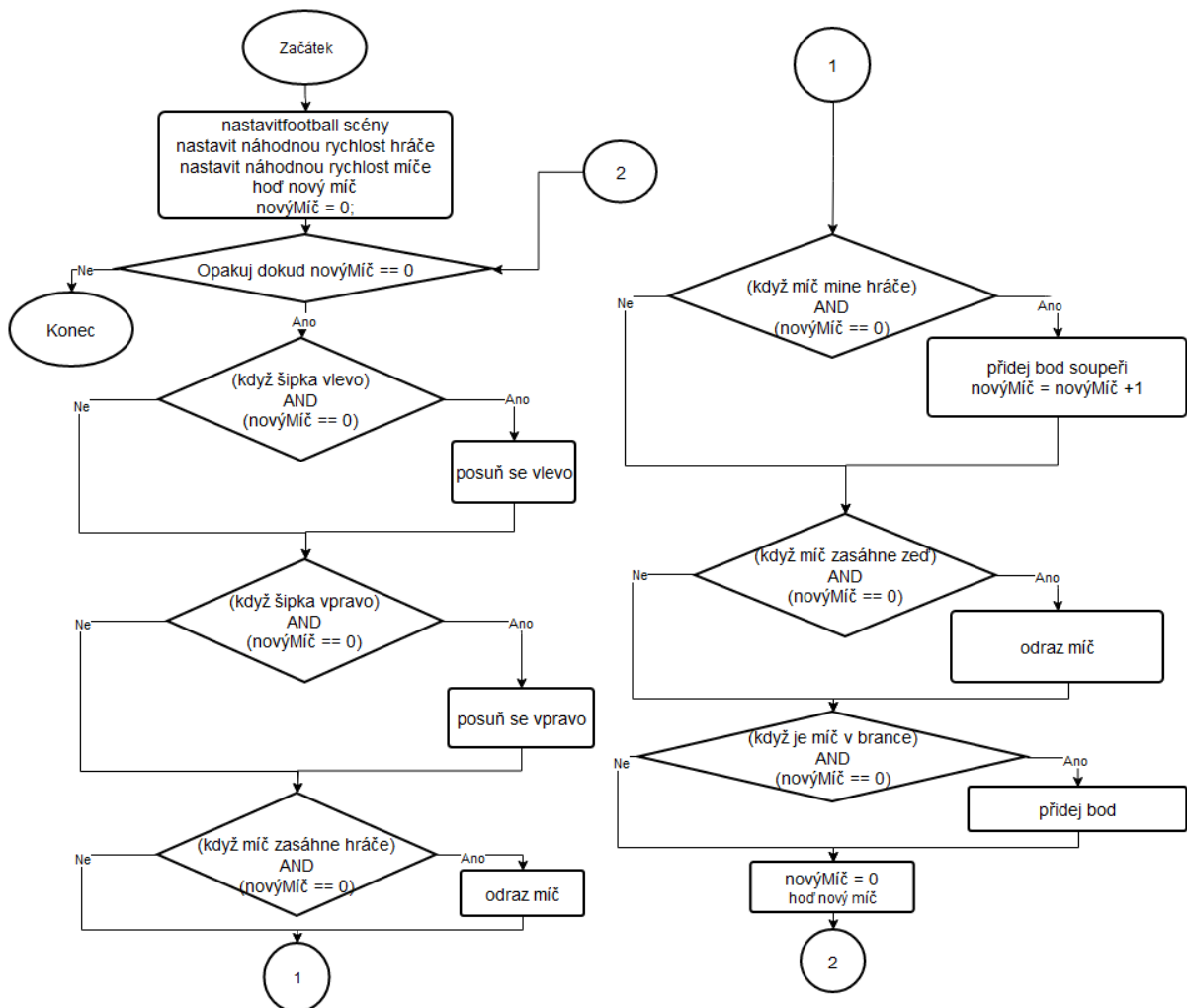
Obrázek 8 - Řešení vývojového diagramu 2 v prostředí Code.org v projektu Sporty

4. Ve druhém vývojovém diagramu není řešeno opakování jednotlivých částí po hození nového míče, tj. na rozdíl od vývojového diagramu, ve kterém není řešeno opakování, je v projektu Sporty řešení opakování pomocí cyklu, které není na první pohled zřejmé. Zde by si žáci měli uvědomit, k čemu je dobré opakování některé činnosti

5. Vývojový diagram, kdyby se měly stále opakovat jednotlivé části, by byl složitý, proto bude ukázka řešení pomocí vývojového diagramu trochu zjednodušená. Opakování je bráno pouze v úvahu, když má nějaký hráč získat bod, v jiných případech ne.

### Správné řešení:

- Do vývojového diagramu je přidána proměnná novýMíč, jejíž hodnota je nastavená na hodnotu 0.
- Pokud se míč dostane do branky, nebo když hráč míč mine, tak je hodnota proměnné novýMíč zvýšena o hodnotu 1 (je možné žákům vysvětlit) a ostatní podmíněné příkazy v daném průchodu cyklem se již neprovedou.
- Když se vše provede a jsme na konci cyklu, aby se cyklus zopakoval, hodnota proměnné novýMíč je nastavená na hodnotu 0 a přidá se nový míč a cyklus se stále opakuje.
- Proměnná novýMíč je přidána pouze do vývojového diagramu, v sestaveném programu to je řešeno způsobem, který není patrný a je součástí projektu.



Obrázek 9 – Řešení vývojového diagramu s cyklem s podmínkou na začátku

### Na co si dát pozor

- Nejdříve by žáci měli navrhnout algoritmus řešení ve formě vývojového diagramu, tj. měli by si nejdříve možné řešení rozmyslet. I když je příklad jednoduchý, žáci by měli mít návyk



si nejdříve vše důkladně rozmyslet, navrhnout řešení a teprve potom ho realizovat, aby dokázali dále programovat v nějakém vyšším programovacím jazyku.

- Vývojové diagramy neuvažují možnost použití cyklů, protože jednotlivé činnosti by se měly stále opakovat, kromě vývojového diagramu na obrázku 9. Vývojové diagramy by byly složité a nepřehledné.
- Vývojové diagramy odpovídají řešení, které je uvedeno na obrázku 8, kde jednotlivé zelené bloky obsahují slovíčko když, tj. podmíněný příkaz.
- Pokud budou žáci vytvářet vývojový diagram, vhodné je i jeho nakreslení, vytvoření v on-line nástroji pro ně může být složité nebo s ním již musejí mít nějaké zkušenosti.

#### *Alternativní řešení*

- Daný příklad je vhodnější pro chlapce, dívky mohou vyzkoušet jiný sport, než je fotbal, případně mohou zvolit jiný, podobný projekt, např. Flappy apod.

#### *Literatura a zdroje*

- Vývojové diagramy jsou vytvořeny v on-line nástroji, který je dostupný na adrese: <https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new=Flowchart>
- Zvolený projekt je on-line veřejně dostupný na adrese: <https://studio.code.org/projects/public>
- Screeny obrázků, slouží pouze pro rychlejší vyhledání zvoleného projektu, jsou vytvořeny z výběru projektu na stránce <https://studio.code.org/>
- Jednotlivá řešení jsou vytvořena autorem příspěvku.