

# **Vysoké učení technické v Brně Fakulta informačních technologií**

IMP projekt - dokumentace

## **Hra HAD**

Pavel Hurdálek (xhurda01)

## **Obsah**

<b>Úvod</b>	<b>1</b>
<b>Návrh programu</b>	<b>1</b>
<b>Popis implementace</b>	<b>2</b>
<b>Základní vlastnosti programu</b>	<b>3</b>
<b>Závěr</b>	<b>3</b>
<b>Odkazy</b>	<b>4</b>

*10. prosince 2022*

# Úvod

Cílem projektu je navrhnout a implementovat zjednodušenou verzi hry Had na platformě FitKit3 osazenou mikrokontrolérem Kinetis K60. Zobrazování se provádí na maticovém displeji.

Pro splnění projektu není nutné generovat na herní ploše ovoce, či měnit délku hada. [2]

Projekt je vytvářen ve vývojovém prostředí Kinetis Design Studio (KDS).

## Demonstrace funkcionality

<https://youtube.com/shorts/ZNJnc8lxUIE?feature=share>

## Zapojení

Externí maticový displej byl k desce FitKit3 zapojen na portu P1 dle prezentace k projektu. [1]

## Návrh programu

Program je implementován modulárně v jazyce C.

V projektu jsou vytvořeny dva moduly:

- common - nadefinování společných datových typů (bod a směr, viz níže)
- snake - datové typy hada a veškeré operace nad nimi

Pro jednodušší práci a čitelnější kód je vytvořeno několik datových typů:

- Bod - reprezentuje souřadnice v rámci maticového displeje
- Směr - určuje směr pohybu hada
- Část hada - jeden článek těla hada
- Had - seskupuje části hada do jednoho objektu

Had je reprezentován cyklickým lineárním seznamem jednotlivých jeho částí. Tento způsob byl zvolen zejména kvůli jednoduchosti následovné implementace pohybu hada a možnému rozšíření na plnohodnotnou hru Had, ve které se délka hada dynamicky mění.

## Popis implementace

Implementace se nachází v adresáři **source**, ostatní soubory byly vygenerovány automaticky při vytváření projektu ve vývojovém studiu.

Vstupním bodem programu je funkce **main** v souboru *main.c*.

### Datové typy

Bod (**point\_t**) je reprezentován dvojicí integerů *x* a *y*, v této části není ošetřeno přetečení mimo maticový displej a předpokládá se tedy správné použití.

Směr (**direction\_t**) je enumerátor obsahující výčet všech směrů.

Část hada (**snake\_part\_t**) je jedním prvkem seznamu. Obsahuje bod, tedy umístění v rámci displeje, a ukazatel na následující část hada. Jelikož se jedná o cyklický lineární seznam ukazuje první prvek (hlava hada) na poslední (ocas).

Had (**snake\_t**) je samotná reprezentace hada, zahrnuje ukazatel na hlavu hada, ukazatel na část hada určenou pro aktuální výpis a aktuální směr.

### Specifikace hada

Změna směru hada je omezena, aby se had nemohl na místě otočit do protisměru.

Délka hada se v programu nastaví podle hodnoty v makru **LENGTH** v souboru *main.c*. Pro validní chování je nutné zadat délku od 1 do 8. V pro odevzdávané verzi byla zvolena délka 5.

Z důvodu délky vyšší než 4 je nutné řešit kolize sama se sebou. Kontrola se provádí při každém posunu (tato kontrola je zapouzdřena v rámci volání funkce pro posun z modulu *snake*). V případě kolize je provedena animace a hra resetována.

Pozice v rámci displeje je udržována pomocí operace modulo, díky čemuž je dosaženo požadovaného chování při přechodu mimo displej (při překročení hranice z jedné strany přichází z druhé).

### Inicializace

Program není řešen pomocí dynamické alokace, na začátku se tedy vytvoří globální pole částí hada, které jsou naplněny a navázány na sebe do cyklického lineárního seznamu.

Zároveň se vytvoří proměnná reprezentující běh hry (**game**), had se pohybuje pouze v případě, že je nastavena na 1.

Ve funkci **SystemConfig** jsou nastaveny všechny potřebné porty a periferie. Pro displej je využit **PORTA**, tlačítka jsou na **PORTE** a časovač **PIT**.

### Pohyb hada

Pro pohyb je využito přerušení způsobované časovačem (**PIT** - Periodic Interrupt Timer) vyvolané dvakrát za sekundu. V obslužné rutině se vynulují příznaky přerušení, zkontroluje se zda probíhá hra a případně se provede posun hada.

Posun hada se provádí překopírováním souřadnic hlavy do ocasu s modifikací podle aktuálního směru pohybu. Následně se změní ukazatel na hlavu, aby nyní referencoval aktuální první článek hada, tedy dřívější ocas.

Ke změně směru dojde při přerušení z tlačítek (**PORTE**), které slouží pro ovládání hada.

## Zobrazení hada

Zobrazení na displeji probíhá v nekonečné smyčce ve funkci **main**. Pokud je hra probíhá, získá se pomocí metody **snake\_get\_point** bod pro výpis (v rámci tohoto volání se modifikuje ukazatel pro výpis v hadovi, aby příště byl vrácen bod příslušející další části) a poté se se rozsvítí ledka na pozici dané souřadnicemi bodu (funkce **print\_point**). Zobrazování tedy probíhá neustále po dobu hry.

Funkce **print\_point** po nastavení příznaků a rozsvícení dané ledky čeká krátkou dobu za účelem zvýšení jasů, až poté vynuluje registry, aby bylo možné pokračovat ve vykreslování.

## Základní vlastnosti programu

Po inicializaci globálních proměnných a nastavení hardware se provede animace, při které se rozsvítí celý displej. Poté se přechází do stavu hra, kde se pohybuje had po herní ploše. Při kolizi hada sama se sebou dochází k animaci postupného rozsvěcování sloupců, což signalizuje konec hry. Poté se vše uvede do původního stavu a hra začíná od znovu.

Jediným způsobem jak hru restartovat (prohrát) je výše zmíněná kolize.

## Závěr

Projekt splňuje všechny body stanovené v zadání.

Jsou zde také ošetřeny nespecifikované stavy. Například:

- přechody mimo hrací plochu
- kolize hada sama se sebou

Implementace je zvolena tak, aby bylo možné případné rozšíření na plnohodnotnou hru Had s generováním ovoce a dynamickou délkou.

## Odkazy

- [1] Ing. Václav Šimek. *IMP – projekt „HAD“*. URL: [https://www.fit.vutbr.cz/~simekv/IMP\\_projekt\\_had.pdf](https://www.fit.vutbr.cz/~simekv/IMP_projekt_had.pdf). (10.12.2022).
- [2] Ing. Václav Šimek. *Š - ARM-FITkit3 či jiný HW: Hra HAD*. URL: [https://www.vut.cz/studis/student.phtml?script\\_name=zadani\\_detail&apid=230998&zid=51286](https://www.vut.cz/studis/student.phtml?script_name=zadani_detail&apid=230998&zid=51286). (10.12.2022).