

# KPG

## GAME OF LIFE

# Obsah

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Analýza úlohy</b>	<b>2</b>
2.1	Algoritmus Game of life . . . . .	2
<b>3</b>	<b>Uživatelská příručka</b>	<b>3</b>
<b>4</b>	<b>Závěr</b>	<b>4</b>

# 1 Zadání

Implementujte zobrazování vývoje buněk v Game of Life jako rovinnou animaci s možností volby libovolné počáteční konfigurace (konfigurace může buď být uložena číselně v souboru nebo se zadávat graficky interaktivně na obrazovce, podle vašich možností a znalostí).

- (Splnil) Za odevzdaný program v dvoubarevné verzi kresby s dokumentací získáte 3 body.
- (Splnil) Pokud váš program bude používat také barevné odlišení buněk, získáte další 2 body.
- (Nesplnil) Pokud váš program dovolí navíc přepnout do 3D zobrazení (místo 2D animace budou buňky dalších přibývat jako 3D buňky ve svislém směru), získáte dalších 5 bodů.

## 2 Analýza úlohy

Pro simulaci bude nejvhodnější zvolit si mapu obdélníkového typu, kvůli snadné reprezentaci maticí. Souřadnice každého bodu na mapě budou tedy odpovídat indexům ve dvourozměrné matici (korektně přepočteno z velikosti jedné buňky). Pro základní simulaci by stačilo ukládat v matici záznam pouze o tom zda buňka je živá či nikoliv. V našem případě jsem ovšem zvolil reprezentaci jedné buňky vlastní strukturou, především kvůli snadné rozšiřitelnosti základního algoritmu.

Ke vstupu bude nejsnazší zadávání kreslením žádaných konfigurací, vzhledem k tomu že by simulace měla být určena především k experimentování. Dále by měla být implementována možnost ovlivnění rychlosti simulace a pro zajímavost barevné odlišení déle žijících buněk.

### 2.1 Algoritmus Game of life

Simulace spočívá ve správné volbě pravidel pro chování buňky. Základní pravidla (viz dále) jsou zvolena tak, aby byl výsledek "zajímavý" z pohledu vývoje různých uskupení buněk. Pro ilustraci provedeme implementaci základních pravidel a dále různými jejich úpravami můžeme dostat velice odlišné

výsledky. Výsledek silně závisí na zvolených pravidlech a může být obtížné najít rovnováhu mezi přírůstkem a úbytkem nových buněk.

Při samotném zjišťování do jakého stavu buňka přejde z aktuálního cyklu do nového je nutné oddělit údaje o aktuálním cyklu a údaje o cyklu následujícím, aby nedocházelo k ovlivňování ještě nezpracovaných buněk průchodem matice.

### Pravidla základního algoritmu

- Pokud je buňka živá a má dva nebo tři živé sousedy, zůstává živou i v dalším cyklu. V opačném případě umírá.
- Pokud je buňka mrtvá a má tři živé sousedy buňka se stává živou.

## 3 Uživatelská příručka

Po spuštění programu je nejprve potřeba zadat horizontální a vertikální hranice simulace. Tyto rozměry jsou udávány v počtu buněk. Tlačítkem **Create Map** vytvoříme prázdnou mapu simulace. Mapa je ohraničena tenkou červenou čarou která zvýrazňuje hranici. Tato hranice nemusí být přes celou volnou plochu kvůli zachování poměru stran. Dále při zvolení příliš velkých rozměrů může aplikace běžet výrazně pomaleji.

Po vytvoření mapy se zpřístupní další nastavení. Při zvolené volbě **Default** se použije základní algoritmus (viz. 2.1) a uživatel má možnost klikáním případně tahem ožивovat buňky na mapě. Pravé tlačítko myši funguje jako guma ("zabití buňky").

Simulaci je možné odstartovat stisknutím tlačítka **Start / Continue** a pozastavit tlačítkem **Stop**. Tlačítko **Next Cycle** slouží ke krokování jednotlivých cyklů (aplikování pravidel na všechny buňky mapy). Konečně "slidem" **Speed** lze nastavit rychlost simulace podle potřeby.

Čistou simulaci lze začít opětovným vytvořením nové mapy (tlačítko **Create Map**).

Volba **Advanced** v sekci **Simulation parameters** přepne simulaci ze základních pravidel na experimentální. Dále je možné pomocí voleb **Blue** a **Red** kreslit různé druhy buněk. Tyto buňky mají trochu odlišná pravidla. Nicméně oba druhy mají nastavený větší přírůstek než úmrtnost. Při větších rozměrech mapy může docházet k zajímavým výsledkům ovšem na úkor výpočetní náročnosti aplikace.

## 4 Závěr

Program je určen pro experimentování s pravidly "Game of Life". Při praktickém zkoušení různých pravidel a konfigurací jsem ovšem došel k závěru, že nastavování různých parametrů pravidel z uživatelského rozhraní nemá žádný velký význam, při zachování jednoduchosti aplikace. Nevhodným zvolením výše zmíněných parametrů vznikají nezajímavé výsledky kvůli vysoké úmrtnosti nebo přírůstku buněk. Nastavování parametrů pravidel by mělo smysl při implementaci přidávání vlastních komplexních pravidel z uživatelského rozhraní a proto jsem toto nastavení z panelu vyřadil.

Při experimentování s pravidly a zařazením doby života buňky a její síly závislé na počtu sousedů při jejím narození jsem narazil na zajímavé konfigurace (pravidla **Advanced** jsou jednou z nich). Buňky jsou obarveny se světlostí závislou na jejich době života (světlejší buňky jako starší). Z dalších zajímavých konfigurací např. pravidla, která snadno udrží živé buňky při životě a nové buňky se rodí pouze při čtyřech živých sousedech, vznikají při větším počtu živých buněk jejich uskupení. Tyto uskupení se samovolně nerozšiřují ovšem udržují striktně tvar konvexního polygonu.

Modré buňky se při větším rozměru mapy chovají podobně jako například plíseň. Chápu tedy potenciál simulace rozmnožování jednoduchých živých buněk při správné volbě pravidel.

Pro zkoušení nových pravidel je tedy nutné upravovat kód aplikace, ale rozumné uživatelské rozhraní na žádané úpravy by vyžadovalo neúměrné úsilí vůči vlastní aplikaci.