

# Snake klon

Jorge Zuñiga  
ČVUT–FIT  
zunigjor@fit.cvut.cz  
3. ledna 2020

## Úvod

Semestrální práce je klonem hry snake.

([https://en.wikipedia.org/wiki/Snake\\_\(video\\_game\\_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre)))

## Hlavní funkce

Hra nabízí grafické rozhraní vytvořené pomocí knihovny **Pygame**. Nabízí hru pro jednoho a nula hráčů. Hra počítače využívá knihovnu **numPy**.

Ve hře jednoho hráče je možné hru pozastavit, a hra si pamatuje nejvyšší dosažené skóre.

Hra počítače vyhraje hru snake (zaplní celé herní pole).

## Metody/postupy/algoritmy

Program je napsán se zaměřením na postupy OOP. To je nejvíce vidět v metodě `draw()`, kterou má každý objekt hry. Ta se stará o vykreslení objektu na obrazovku.

Co se samotného běhu programu týče, tak ten běží v nekonečné smyčce díky které zůstává spuštěn dokud si uživatel nepřeje jeho ukončení. Tato myšlenka je základem každé části hry, od hlavního menu až po samotný průběh hry snake. Celý program by mohl vzdáleně připomínat konečný automat.

Algoritmicky nejzajímavější částí je hra počítače. Ta je řízena cestou která prochází všechna políčka a had tak projde celé pole, aniž by do sebe narazil. Cestu jsem vybral já. Toto samo o sobě tak zajímavě nezní a tak jsem přidal možnost zvýšit rychlost hry počítače. Bohužel časový tlak mi neumožnil implementovat zajímavější algoritmus, ačkoli jsem si o něm načetl hodně materiálů.

Hlavní myšlenkou pro “zajímavější” algoritmus je nalezení náhodné Hamiltonovské kružnice na grafu. ([https://en.wikipedia.org/wiki/Hamiltonian\\_path](https://en.wikipedia.org/wiki/Hamiltonian_path)). Tím se cesta, po které se had hýbe, stane zajímavější.

Nalezení takovéto cesty na jakémkoliv grafu patří do třídy problémů NP-complete. Samotné nalezení takového cyklu se dá provést více způsoby. Jedním z nich je vygenerování náhodného labyrintu na polovině výšky i šířky hracího pole.

([https://en.wikipedia.org/wiki/Maze\\_generation\\_algorithm](https://en.wikipedia.org/wiki/Maze_generation_algorithm))

Poté stačí labyrint projít tak, že budeme vždy, kdy to bude možné odbočovat doleva. Tím vygenerujeme Hamiltonovskou kružnici. Tím se učí cesta po které se had bude pohybovat.

Dalším krokem je přidání možnosti přeskakovat některé části této kružnice. Zde už záleží na faktorech jako je vzdálenost k jídlu nebo zaplněnost hracího pole (protože při velkém zaplnění se spíše had zasekne)

## Výsledky

Ačkoli algoritmus při hře počítače není příliš zajímavý. Dalo by se říct, že splní to co má za úkol (vyhrát hru).

Program plně implementuje grafické rozhraní i hru pro jednoho hráče. Z tohoto pohledu jsem velmi spokojen s dosaženými výsledky.

## Závěr

Práce na tomto projektu mě bavila a mám motivaci se k němu vrátit, hlavně co se týče vylepšení algoritmu pro hru počítače.

Knihovna pygame je postavená na knihovně SDL pro jazyky C/C++, díky čemu mi nebyla úplně cizí.

Dalším aspektem, který by se dal v budoucnu zlepšit, je využití hezčích grafik pro výkres hada a jídla (i když po dnech práce se mi ten kostičkovaný had docela líbí).

## Reference

[https://en.wikipedia.org/wiki/Snake\\_\(video\\_game\\_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))

[https://en.wikipedia.org/wiki/Hamiltonian\\_path](https://en.wikipedia.org/wiki/Hamiltonian_path)

[https://en.wikipedia.org/wiki/Maze\\_generation\\_algorithm](https://en.wikipedia.org/wiki/Maze_generation_algorithm)

[https://en.wikipedia.org/wiki/Prim%27s\\_algorithm](https://en.wikipedia.org/wiki/Prim%27s_algorithm)

[https://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm)

<https://codeforces.com/blog/entry/79788>

<https://stackoverflow.com/questions/7371227/algorithm-to-find-a-random-hamiltonian-path-in-a-grid/20056736>

<https://johnflux.com/2015/05/02/nokia-6110-part-3-algorithms/>

*"Secondary structures in long compact polymers"* by Richard Oberdorf, Allison Ferguson, Jesper L. Jacobsen and Jané Kondev, Phys. Rev. E 74, 051801 (2006)