

## Dokumentacija za paralelnu (i sekvencijalnu) verziju Pacman igrice

Za realizaciju paralelne igrice PacMan koriscena je prvo sekvencijalna implementacija koja je pomocu biblioteke OpenMP pretvorena u paralelnu. Kod je podeljen skoro potpuno u funkcije koje na kraju svaka nit posebno poziva u zavistni od toga koja je. Za obradu slike je koriscena biblioteka OpenCV, kao razvojno okruzenje je koriscen MicrosoftVisual Studio 2022 a operativni sistem je Windows.

Za pocetak su date dve identicne matrice, prva matrica je ona po kojoj se krece sam PacMan i ona se **ne menja**, 1 oznacava mogucnost prolaza a 0 zid. Druga matrica predstavlja lokaciju hrane i kada se hrana pokupi polje sa '1' se brise i menja sa '0'. Takodje pomocu ove matrice kreirana je mapa pomocu OpenCV funkcija (pokusano je sa screenshotom slike sa pdf ali je tu bilo mnogo problema i kalibrisanja pa je to zanemareno). Zatim su definisane inicijalne pozicije pacmana i duhova pomocu x i y koordinata i brojac koji je brojati skupljene poene.

- Funkcija **eat** dobija lokaciju i sliku map i ukoliko ima hrane na tom polju uklanja taj *food.png* sa pozicije.
- Funkcija **movement** obavlja pomeranje pacmana u zavisnosti od pritisnutog dugmeta (w, a, s, d; stelice iz nekog razloga na mom racunaru ne funkcionisu (cv\_waitKey mi nista ne vraca na stelice) na mom sistemu, probacu negde drugde).
- Funkcije **ghostMovement1** i **ghostMovement2** su identicne (samo pomeraju drugog duha) i one su po funkcionalnosti pomeranja iste sustinski kao i **movement** samo se duhovi pomeraju pomocu randomizacije.
- Funkcije **drawGhost1**, **drawGhost2** i **drawPacman** rade iscrtavanje karaktera. Desavalo mi se na pocetku realizacije da openCv pravi neke probleme zbog slika, pa u tom slucaju stoji dole komentaranis kod koji menja karaktere sa krugovima. (Ne bi trebalo da ima problema sa ovim u poslednjoj implementaciji.)
- **Victory** i **Defeat** su pomocne funkcije koje predstavljaju endgame mehanizam. Kada se skupi svih 210 poena igrac pobeđuje i igra se završava ili ako nas duh pojede igrac gubi.

Unutar **main** funkcije, se podseva broj niti i kreira se mapa sa hranom. While(1) petlja sadrzi podelu posla izmedju niti gde :

Nit 0 se bavi menjanjem mape i iscrtavanjem karaktera.

Nit 1 se bavi pomeranjem pacmana.

Nit 2 se bavi pomeranjem duha 1.

Nit 3 se bavi pomeranjem duha 2.

Van pragme su postavljeni *endgame mehanizmi* za koje nije potrebna paralelizacija. Nakon kraja igre je potrebno pritisnuti bilo koji karakter za izlazak iz igrice.

Obzirom da je radjeno na windows os potrebno je podesiti neke parametre u razvojenom okruzenju, konkretno MS Visual Studiu. Potrebno je imati ekstrahovan opencv folder I u *properties* naseg projekta podesiti sledece :

1. U *configuration properties* na polju *command arguments* podesiti 4 kao broj niti pomocu kojeg pokrecemo kod.

Debugging	
VC++ Directories	Command \$(TargetPath)
C/C++	Command Arguments 4

2. U polju *VC++ directories* podesiti sledeca podebljana polja:

General	Executable Directories	\$(VC_ExecutablePath_x64);\$(CommonExecutablePath)
Advanced	Include Directories	<b>D:\opencv\build\include;\$(IncludePath)</b>
Debugging	External Include Directories	\$(VC_IncludePath);\$(WindowsSDK_IncludePath);
VC++ Directories	Reference Directories	\$(VC_ReferencesPath_x64);
C/C++	Library Directories	<b>D:\opencv\build\x64\vc16\lib;\$(LibraryPath)</b>
General	Library WinRT Directories	\$(WindowsSDK_MetadataPath);
Optimization		

3. U polju *C/C++ -> Language* podesiti polje *OpenMP Support* na **Yes**.

4. U polju *Linker -> Input -> Additional Dependencies* napisati *opencv\_world490d.lib* (ako je verzija *opencv 4.9.0*)

Za kraj, na pocetku koda ukljuciti sledece biblioteke:

```

1  #include <opencv2/opencv.hpp>
2  #include <stdio.h>
3  #include <omp.h>
4  #include <iostream>
5
6  using namespace cv;
7  using namespace std;
8

```