|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Милош Игњатовић RA95-2016

Слободан Костић RA139-2016

Арсеније Говедарица RA215-2016

**320x240 9-bit RGB VGA**

**Трка краљева**

ИСПИТНИ РАД

- Логичко пројектовање рачунарских система 2 -

Ментор: Милош Суботић

Нови Сад, 2019

**Садржај**

[1. Увод 4](#_Toc10133862)

[1.1 Трка краљева 4](#_Toc10133863)

[1.2 Платформа за развој пројекта 4](#_Toc10133864)

[2. Опис решења 6](#_Toc10133865)

[2.1 Иницијализација плоче 6](#_Toc10133866)

[2.2 Играчка логика и помоћне функције 6](#_Toc10133867)

[2.2.1 Функција findLegalMoves 6](#_Toc10133868)

[2.2.2 Функција isKingAttacked 7](#_Toc10133869)

[2.3 Графика и помоћне функције 7](#_Toc10133870)

[2.3.1 Функција drawMap 8](#_Toc10133871)

[2.3.2 Функција drawingCursor 8](#_Toc10133872)

[2.3.3 Функција drawTable 8](#_Toc10133873)

[2.3.4 Функција drawBackground 8](#_Toc10133874)

**Скраћенице**

**FPGA** - *Field Programming Gate Array*, Програмабилне секвенцијалне мреже

**USB** - *Universal Serial Bus*, Универзална серијска магистрала

**VGA** - *Video Graphics Array*, Видео графички излаз

**SDK** - *Software Development Kit*, Програмски пакет за развој софтвера

**VHDL** - *VHSIC Hardware Description Language*, Програмски језик за опис хардвера

**RGB** - *Red Green Blue*, Црвено-зелено-плави формат боја

**GIMP** - *GNU Image Manipulation Program*, GNU програм за обраду слика

# Увод

## Трка краљева

Игра „Трка краљева“ (енг. Racing Kings) је једна од најпознатијих шаховских игара. За разлику од оригиналног шаха, циљ ове игре није довести туђег краља у безизлазни положај, већ својим краљем први стићи до супротног краја табле. Оба играча крећу са исте стране табле, а њихове фигурице су распоређене у два реда, симетрично по вертикалној оси. Пошто је простор ограничен, а њихова улога не би имала смисла у кретању у истом смеру, пијуни нису део ове игре. Сваки играч на почетку игре на располагању има краља, краљицу и по два топа, ловца и скакача. С обзиром на то да је ова игра „тркачка“, потези којим би противнички краљ био нападнут нису дозвољени. Ову игру је осмислио енглески шахиста Вернон Рајландс Партон средином 20. века.

## Платформа за развој пројекта

Овај испитни рад је развијен на E2LP развојној FPGA плочи уз коришћење USB програматора. Развојна плоча, између осталих, поседује VGA излаз за приказ графике на екстерном монитору. Већина модерних монитора, па и оних који су били спојени на плочу приликом израде, имају 16:9 однос ширине и висине. Међутим, развојна плоча има подешен излаз са односом 4:3 који је раније био актуелан. Резолуција графичког излаза је подешена на 320 пиксела по ширини, док се по висини исцртава 240 пиксела. Четири тастера су искоришћена за кретање играча, док пети служи као мултифункционална команда.

Радно окружење за развој испитног рада припада програмском пакету Xilinx SDK, који има уграђену функцију директне комуникације са развојном плочом. Део кода везан за повезивање хардвера је писан у VHDL језику, док је програмски језик С коришћен за развој саме игре.

# Опис решења

## Иницијализација плоче

На самом почетку извршавања програма, позива се неколико функција за иницијализацију плоче. Коришћењем уграђене функције за упис у меморију, цела површина монитора се боји у светло сиву боју, која даје добар контраст другим елементима шаховске табле. Потом се иницијализује матрица са положајима фигурица и оне се исцртавају на шаховској табли. Коначно, позива се функција void move(), у којој су садржани позиви свих функција које су везане за саму логику играња (енг. gameplay). Матрице са положајима фигурица и дозвољеним потезима су реализоване као глобалне променљиве, због лакшег читања и писања у оквиру свих функција.

## Играчка логика и помоћне функције

У оквиру функције void move() се иницијализују координате курсора и променљиве, које садрже информације о играчу који је на потезу и треунтном моду игре. У бесконачној петљи се констатно проверава стање тастера, те се у складу са њима и помера курсор. У случају да је притиснут централни тастер SELECT, проверава се тренутно стање игре и позивају додатне функције за провере потеза. Уколико је играч започео потез, притиском на овај тастер преузима фигурицу са места курсора, уколико је његова. Поновним притиском на тастер, фигурица се спушта на жељено место. Ако жељени потез није валидан, не долази ни до каквих измена на табли и играчу се даје да поново одигра потез. Такође, уколико играч врати фигурицу на место одакле је преузео, сматра се да одустаје од померања исте. Стога, принцип „такнуто-макнуто“ се не примењује у овој реализацији.

### Функција findLegalMoves

Функција за рачунање валидних потеза као параметар прихвата тренутни положај курсора, на основу кога чита матрицу фигура и добавља информацију о којој фигурици је реч. У зависности од фигуре, испитује се могуће померање у свим правцима од тренутне координате. Ако се у току провере са табле очита пријатељска фигура, провера се не наставља. У случају да је на том месту непријатељска фигура, провера се обуставља закључно са тим местом.

Пошто ова тркачка игра забрањује напад на противничког краља, потребна је додатна провера сваког потеза. За свако место које одговара логици кретања фигурице се позива додатна функција, која прави копију шаховске табле са случајем да је одигран потез.

### Функција isKingAttacked

Ова функција има циљ да спречи изазивање шаха потезом играча. Она у исто време проверава да ли је извршен напад на противничког краља, али и проверава да ли ће потенцијални потез открити сопственог. Она као параметре прима координате фигурице пре и после потеза.

У оквиру функције се прави локална копија матрице шаховске табле. На месту одакле је започет потез се уписује празно место, док се на месту потенцијалног потеза уписује фигурица којом корисник намерава да изврши потез. Проласком кроз матрицу налазе се положаји црног и белог краља.

На основу ових позиција, у обрнутој логици се испитује напад. Наиме, провера креће од координата на којима се налазе краљеви и потом се проверавају координате са којих је могуће напасти краља. Провере се заустављају у случају наиласка на пријатељску, односно на непријатељску фигуру која неће изазвати напад. Повратна вредност ове функције је целобројна врредност 0 или 1, у зависности од тога да ли је неки од краљева нападнут.

У зависности од повратних вредности се модификује матрица легалних потеза, која се попуњава у функцији из поглавља 2.2.1.

## Графика и помоћне функције

Испис графике на VGA излаз из плоче је реализован уписивањем деветобитних вредности на координате пиксела. Вредности које се уписују садрже по три бита за црвену, зелену и плаву боју (RGB формат), док су функције за испис карактера онемогућене. Стога, потребно је нацртати сва слова и сличице у једном фајлу, одакле ће се читати вредности сваког појединачног пиксела.

Пошто помоћне функције за исцртавање пиксела подразумевају да се исцртава квадрат, а да шаховске фигурице и пратећи текстови немају ту форму, потребно је направити посебне карактере који ће бити игнорисани приликом исцртавања. Као и у филмској индустрији, фајл са свом потребном графиком је садржао зелену позадину, која се игнорише приликом уписа у меморију. Након детаљне обраде, искоришћена је опција програма GIMP за извоз bitmap слике у матрицу C програмског језика. Из тог фајла се индексирају пиксели и уписују у меморију плоче. Фајл sprites\_green је величине 95\*95 пиксела и садржи све фигурице и текстове за испис назива игрице, редова и колона на табли, као и исписа играча на потезу.

Због разлика у односима ширине и висине на развојној плочи и монитору, све сличице су сужене у самом фајлу, како би се добио правилан облик након развлачења до односа 16:9.

### Функција drawMap

Најбитнија функција за сам испис графике прима шест параметара. У њој је потребно индексирати почетне пикселе сличица у извезеном фајлу као и координате на екрану. Додатни параметри су ширина и висина сличице која се исцртава на екран. Функција нема могућност скалирања сличице, па је размера пиксела у фајлу једнака размери пиксела на екрану.

Функција из фајла преузима вредности црвене, зелене и плаве боје. По стандарду се заузима по осам бита за сваку боју, али меморија развојне плоче подразумева укупно девет бита, односно по три за сваку компоненту. Због тога се врши померање бита удесно, након чега се компоненте боја спајају у једну целину.

Због ефекта „зеленог платна“, пре самог уписа у меморију се проверава и да ли је очитани пиксел искључиво зелене боје. Уколико јесте, његов испис се игнорише, па на екрану остаје претходна позадина.

### Функција drawingCursor

Проширена функција за исцртавање квадратног курсора служи за обележавање тренутног кретања играча, као и дозвољених потеза у случају одабирања фигурице. Као параметар прима три различите вредности, а у зависности од њих исцртава курсоре различитих боја. Кретање корисника кроз таблу је обележено црном бојом, дозвољени потези зеленом а обележавање жељеног потеза, који је дозвољен, црвеном бојом.

### Функција drawTable

Исцртавање фигурица на екран се врши након сваког потеза и служи да кориснику прикаже тренутно стање на табли. Она чита глобалну матрицу шаховске табле и за сваку фигурицу позива функцију из поглавља 2.3.1. Празна поља се игноришу током извршавања ове функције, с обзиром на то да се у програму прво исцртава позадина, па тек онда фигурице.

### Функција drawBackground

Ова програмска функција је најбитнија за оно што сам корисник види. Она представља комбинацију наведених графичких функција и заслужна је за координацију исписа на целом екрану. Она се позива након сваког потеза и исписује битне корисничке информације. У свакој итерацији она исписује називе редова и колона, назив игрице као и играча који је тренутно на потезу. На крају ове функције позива се функција из поглавља 2.3.3, која је заслужна за испис свих фигурица на табли.