



Софийски университет „Св. Климент Охридски“  
Факултет по математика и информатика

# Примерни теми за проекти

курс Функционално програмиране  
за специалности Информатика и Компютърни науки (2-ри поток)  
зимен семестър 2020/21 г.

Важно: Освен ако в задачата не е указано на кой език да се реши, допустимо е да се използва Haskell или Scheme (R5RS или Racket).

## Общи указания

Темите за проектите са описани в по-общи рамки. За редица аспекти от решението е оставена възможност да изберете как да подходите към тях и как да ги реализирате.

След като изберете тема за своя проект, опишете накратко какво планирате да направите в документ в Google Drive и го споделете с екипа на курса. В документа с описанието посочете коя тема сте избрали и как конкретно смятате да я реализирате.

Решенията трябва да се качат в съответната форма на страницата на курса в Moodle, в срок до 23:55 на 24 януари 2021 г.

## Тема 1

Напишете собствен интерпретатор на Scheme, който използва модела на средите. Използвайте C, C++, C# или Java.

## Тема 2

Нека е даден [графичен файл в PBM, PGM или PPM формат](#).

Напишете програма, която получава входен файл в някой от горните формати; разпознава го по съдържанието му; и след това прилага някой от Dithering алгоритмите, които са описани в следната статия (потребителят трябва да може да укаже кой

алгоритъм иска да се приложи; програмата ви трябва да реализира всички 11 алгоритъма от статията):

Tanner Helland (2012) Image Dithering: Eleven Algorithms and Source Code  
<https://tannerhelland.com/2012/12/28/dithering-eleven-algorithms-source-code.html>

Програмата трябва да запише резултата от работата си в изходен файл указан от потребителя.

Програмата ви трябва да може да валидира входа си и да обработва адекватно възникнали грешки (например ако бъде подаден входен файл с некоректно съдържание, ако се подаде неподдържана операция, ако програмата не може да запише данните в изходния файл и т.н.)

## Тема 3

В този проект трябва да реализирате проста CRPG игра.

Играта трябва да се развива в свят, който се състои от множество зони. Зоните трябва да са организирани в насочен граф, който указва от коя зона в коя може да се отиде. Някои от преходите може да изискват да бъде изпълнено някакво условие. Например преди това да сте намерили даден ключ.

Във всяка зона може да има различни предмети, които можете да събирате.

Във всяка от зоните може да има и чудовища. Ако влезете в такава зона, героят ви ще трябва да може да се справи с чудовището преди да премине нататък. Сами можете да изберете как точно да реализирате това (дали при влизане в такава зона задължително да се започне битка или да има вариант това да се избегне по някакъв начин).

Сами преценете каква механика да реализирате за битките, какви умения да има героят ви, а също и дали чудовищата да са случайно генерирани или да са фиксирани за зоните. Също, преценете дали да поддържате екипировка (т.е. дали някои от предметите да могат да се използват от героя -- оръжия, брони и т.н.), дали да се поддържат различни видове герои (воин/магьосник/крадец и т.н.).

Информацията за играта (зони, чудовища и т.н.) трябва да се съхранява в един или повече конфигурационни файлове, които играта ви трябва да зареди при своето стартиране.

В играта трябва да има крайна цел, като сами можете да изберете дали в играта да се фиксира едно от следните или това да може да се указва в конфигурационните файлове:

- Да се достигне дадена финална зона;
- Да се вземе конкретен предмет;
- Да се победи конкретно чудовище.

Интерфейсът на играта може да бъде текстов. Сами можете да изберете как да го организирате -- rogue-like, изцяло текстово и т.н. Например при влизане в дадена зона може да се извежда текст, който описва зоната. След това потребителят може да въвеждат различни команди, например:

- LOOK # за оглеждане наоколо и извеждане на информация за зоната
- TAKE <item> # за взимане на даден предмет
- EQUIP <item> # за екипиране на предмет (ако поддържате това)
- GO <direction> # за преминаване от една зона в друга
- и т.н.

За може играта да не бъде просто пасивно обхождане на зоните, добавете и възможност компютърът да контролира други същества, които се движат между зоните. Това могат да бъдат например:

- Чудовища, които се опитват да достигнат до вашия герой;
- Контролиран от компютъра герой, който също се опитва да достигне до крайната цел преди вас.

За тази цел играта трябва да бъде организирана на ходове. Сами изберете подходяща схема за това, която да е съобразена с начина, по който сте организирали играта. Например ако играта е изцяло текстово базирана, един ход може да изтича когато правите преход от една зона в друга. Ако е rogue-like, може всеки път, когато се преместите от едно квадратче в съседно, да се придвижват и останалите същества и т.н.