

**ВЕЛИКОТЪРНОВСКИ УНИВЕРСИТЕТ**  
**„СВ. СВ. КИРИЛ И МЕТОДИЙ“**



Курсова работа  
по  
Събитийно програмиране  
на тема:

**Змията (Snake game) написана на JavaScript**  
**използвайки Event Driven Design**

Изготвил: Димитър Лъчезаров Захариев

Специалност: Софтуерно инженерство

Фак. №: 20023

Курс: Втори

Проверил: доц. д-р Тодор Йорданов Тодоров

При зареждането на страницата се инициализира масив с `emptySpaces` – (променлива зад която седи число и указва, че на тази позиция няма нищо) и самата змия която е имплементирана като двойно свързан списък. При започване на играта се извиква event handler-a `StartGame(event)`. Прави се проверка дали натиснатият клавиш съществува като такава опция (`IsEventKeyValid(event)`). При успешно преминато условие се извиква `setInterval(TriggerSpawnApple, 8000);`. Подава ѝ се функцията `TriggerSpawnApple` която да се изпълнява през определено време за да хвърля event-a “spawn-apple” който при обработка от `SpawnApple listener`-а пуска ябълка. След изпълнението на handler-a се вика друг event handler от вече натиснатият бутон - `KeydownPress(event)`. Отново след проверка дали бутонът съществува във възможните опции се прави проверка дали подадената посока в която змията ще върви не е противоположна на тази която е подадена (при противоположна подадена посока се използва предишната валидна посока) и се извиква функцията `MakeNextMove(direction)`. В `MakeNextMove` първо се търси къде се намира главата на змията, след това се проверяват дали новите координати (които ще са координатите на новата глава) са валидни. При невалидни координати се хвърля custom event “dead-event”. (Който бива обработен от `GameOver event handler`-а който откача `KeydownPress` и `SpawnApple event handler`-а, и се занулява променливата `stopSpawningApplesId` която пази Id-то за `TriggerSpawnApple` функцията (за да не може да продължи изпълнението на играта след като играчът вече е умрял)). Прави се проверка дали на новата позиция на която ще се настани главата няма никакъв item чрез `SpecialItemChecker()` (ако има такъв, като например ябълка се хвърля event “apple-eaten”) който бива обработен от `UpdatePoints()` и `GrowSnake()` handler-ите.) (`UpdatePoints()` увеличава точките и ги обновява в HTML-a. `GrowSnake()` от своя страна извиква функцията `AddNewTail()` която намира мястото на опашката в даденият момент и добавя нова точка към опашката на змията.). `MoveSnake(position)` премахва опашката (най-крайната точка от змията) и добавя нова точка - „глава“ на змията (в зависимост от посоката в която се движи змията).

`RenderScreen()` се грижи за изрисоването на екрана. При извикването ѝ се извиква друга функция - `ClearScreen()`; която изтрива нарисованото от преди това. Целият масив `map` се обикаля с два вложени цикъла и с помощта на две други функции `IsSnakeBodyPoint(row, column)`, `IsApplePoint(row, column)` се променя видът на div-овете в зависимост от това какъв е даденият елемент в масива (`emptySpace`, `apple`, `snakeDot`).

След изчертаването на екрана се вика отново същата функция `MakeNextMove - setTimeout(MakeNextMove, 500, direction);` и се запазва нейното `Id` - на `timeout`-а в глобалната променлива `shouldMakeAnotherMoveId` (за да може когато `user`-а отново натисне някой бутон и се задейства `handler`-а `KeydownPress(event)` да може да се изчисти `timeout`-а чрез `clearTimeout(shouldMakeAnotherMoveId);` (за да не продължи извикването на `MakeNextMove(direction)` със старият подаден `direction`).