**Обектно Ориентирано Програмиране със C#.NET**

**Курсов проект:**  **Snake game**

***Илия Ангелов Велев фн-44619 ,Информатика***

Подходът към решаването на проблема е обектно ориентиран.Решението на проблема обособява четири класа представящи отделните компоненти на Snake game.

**Клас Food**- реализира храната на змията, той съдържа като член дънна елипса , чийто цвят и размери могат да бъдат зададени, като параметри на конструкторът на класа или чрез property за достъп.Храната бива три вида , всеки от които носи определен брои точки. Броят точки е според цветът на храната .

**Клас Obstacle**- всички препятствия които се появяват по екрана , са инстанции от този клас .Самия клас съдържа в себе си като член данна правоъгълник , чийто свойства могат да бъдат зададени с предоставените properties .Класът има метод

InObstacle, който ни дава възможност да проверяваме за принадлежност на точка в препятствие, за тази цял пазим координатите на препятствието върху игралното поле.

**Клас Snake** – притежава две член данни които обособяват змията, като съставена от две части .Първата част, това е главата на змията , тя е елипса с определян диаметър и цвят , като са предоставени properties за достъп .Втората част от змията, това е нейното тяло , то е представено с Polyline, който съдържа точките от екрана , които влизат в тялото на змията. Клас Snake предоставя опции за задаване на ширина на змията .

**Клас FeedTheSnakeGame** – събира функционалноcтта на останалите класове като пази по една член данна от всички . Класа съдържа Canvas, който се подава, като параметър на конструктора. Независимо от размерите и позицията на полето, играта се играе върху него, това я прави до голяма степен преносима между различни приложения.

Освен Canvas като параметър на конструкторът се подава Label, който служи за изобразяване на резултата на потребителя в точки, като той може да бъде на произволна позиция, което дава възможност за подобряване на интерфейса на приложението. Играта започва със извикване на метода GameStart, неговата задача е да изчисти полето, да добави на полето Label (gameState) , както и да накара системата да следи, дали курсора на мишката се намира върху този Label .Това се налага понеже е добре да започнем играта , когато курсора на мишката е в центъра на игралното поле .

Действително играта започва, когато курсора попадне върху Label(gameState) , като се добавят елементите на играта и събитието MouseMove за нашият Canvas.

Всеки път когато движим курсора, ние взимаме текущата му позиция на курсора, изобразяваме змията спрямо нея, проверяваме за край на играта , проверяваме, дали змията е изяла някоя храна и ако да я добавяме към нейната дължина, според броят точки, който тя дава .

Метода SnakeMove реализира движението на змията като добавя текущата точка към началото на змията и премахва последната.

Играта завършва с изсекването на метода GameOver, който освобождава заетите ресурси и премахва събитиетo MouseMove от полето.

**Други по съществени методи на класа**:

**SnakeCrossesItsTail** – връща истина ако змията е пресякла нейното тяло ,проверява се дали, главата на змията е попаднала, в достатъчно близка околност до някоя от точките, които участват в тялото на змията .

**MoveFood** – задава произволни координати на храната на змията като също така определя и нейния вид (брои точки ,цвят).

**CheckForObstacles** - извършва проверка дали змията се е ударила в препятствие (използва се също, за да предотврати попадане на храна върху препятствие).

**AddObstacles** – определя произволна позиция на всяко препятствие, стреми се препятствията да се засичан възможно най малко.

**Предимства и недостатъци на реализацията** : Съществен недостатък на реализацията е това, че събитието MouseMove следи движението на мишката, при определен интервал, това предизвиква неприятности при определяне на края на играта, също така създава усещане за разтегляне на тялото на змията и затруднява проверката дали змията е пресякла тялото си.