Документација за 4snake

Играта 4snake e варијанта на добро познатата игра Snake, со неколку измени. 4snake поддржува на истата мапа да истовремено најмногу 4 змии да играат. Човечки играчи може да има 0, 1 или 2. Останатите змии се контролирани од страна на компјутерот водени според вештачка интелигенција. За разлика од стандардната игра Snake, во оваа може да избереме колку може да биде бројот на торти на мапата, кои кога ќе се изедат од некоја змија, доведува таа змија да се зголеми во должина. Истотака, на мапата може да се појават и камења на некои места, во кои, ако некоја змија се удри, ќе доведе тој играч да изгуби и змијата да биде отстранета од мапата. Друга значајна новост во играва е воведување на тајмери - часовници кои се појавуваат на мапата, за кои, кога ќе бидат изедени од некоја змија, предизвикува таа змија да продолжи да се движи со иста брзина, а противниците да бидат успорени за фактор од 2 до 5 пати. Фактор од 5 значи дека при секое едно поместување на забавените змии, змијата која го изела тајмерот за истото тоа време ќе се придвижи 5 полиња. т.е. ќе прави 5 пати повеќе движења во истиот временски период.

Исто така, 4snake е програмирана со Design Pattern начин на кодирање со кој многу лесно се прават промени во интерфејсот, со менување на многу малку код можеме да додадеме 5ти играч, менување на бројот на камења, или бројот на торти, или брзината на играта, бројот на тајмери и сл.

Постојат 3 начини на кои играчот може да изгуби :

1. Змијата да се удри од самата себе.

2. Змијата да се удри во некој камен.

3. Змијата да се удри во телото во друга змија- при што само едната се отстранува од играта, таа што удрила со главата, другата продолжува нормално со играта.

Ако две змии директно се удрат двете со главите, тогаш и двете се отстранети од играта.

На крајот победник е играчот чија змија собрала најмногу поени, или достигнала да изеде 50 торти.

Играта завршува кога ќе истече времето, или кога ниедна змија веќе нема да биде дел од играта.

Имплементација:

Секоја змија е претставена со следната класа

public class Snake

{

public int X { get; set; }

public int Y { get; set; }

public float Radius { get; set; }

public int nasoka { get; set; }

public bool otvorena { get; set; }

public SolidBrush br;

public int slednox { get; set; }

public int slednoy { get; set; }

За секоја змија ја чуваме нејзината моментална X,Y координата на главата, насоката кон која е насочена да се движи, X,Y координата каде што ќе се наоѓа во следниот момент - ова го користиме за имплементација на вештачката интелигенција- на другите змии ( подетално за ова понатаму). За секоја змија одделно чуваме нејзината боја, и состојбата дали и е отворена или затворена устата, кое служи за анимирано цртање.

Во главната класа за играта чуваме:

-координати на тајмерот

-информација дали се појавил тајмерот или не

-Колку време поминало откако бил изеден тајмерот ( за успорувањето на противничите змии да не трае засекогаш)

-кој го изел тајмерот

-која е веројатноста да се појави тајмерот- ова функционира на начин што, ако веројатноста е 3%, значи при 100 поместувања на змиите, само во 3 од тие 100 ќе се појави тајмер некаде на мапата.-

-вкупен број на играчи

-број на ботови (змии водени од вештачка интелигенција)

-која змија веќе не е во игра

-големина на мапата

-брзина на играта. должина на временски интервал за секој чекор

-матрица каде што е означено каде има торта

-матрица каде што се означени позициите на змиите. На позицијата на главата ставаме бројка колку што има изедено торти, понатаму по нејзиното дело, ставаме броеви за 1 помали, и при секој чекор вредностите низ целата матрица ги намалуваме за еден.

-број на изедени торти за секоја змија

-број на торти

-број на камења

-број на поени.

При генерирањето на позицијата на тортите, се внимава на таа позиција да нема некоја змија,

При генерирањето на позицијата на камењата, се внимава на таа позиција да нема некоја змија или торта.

При генерирање на позицијата на тајмерот, се внимава на таа позиција да нема некоја змија, торта или камен.

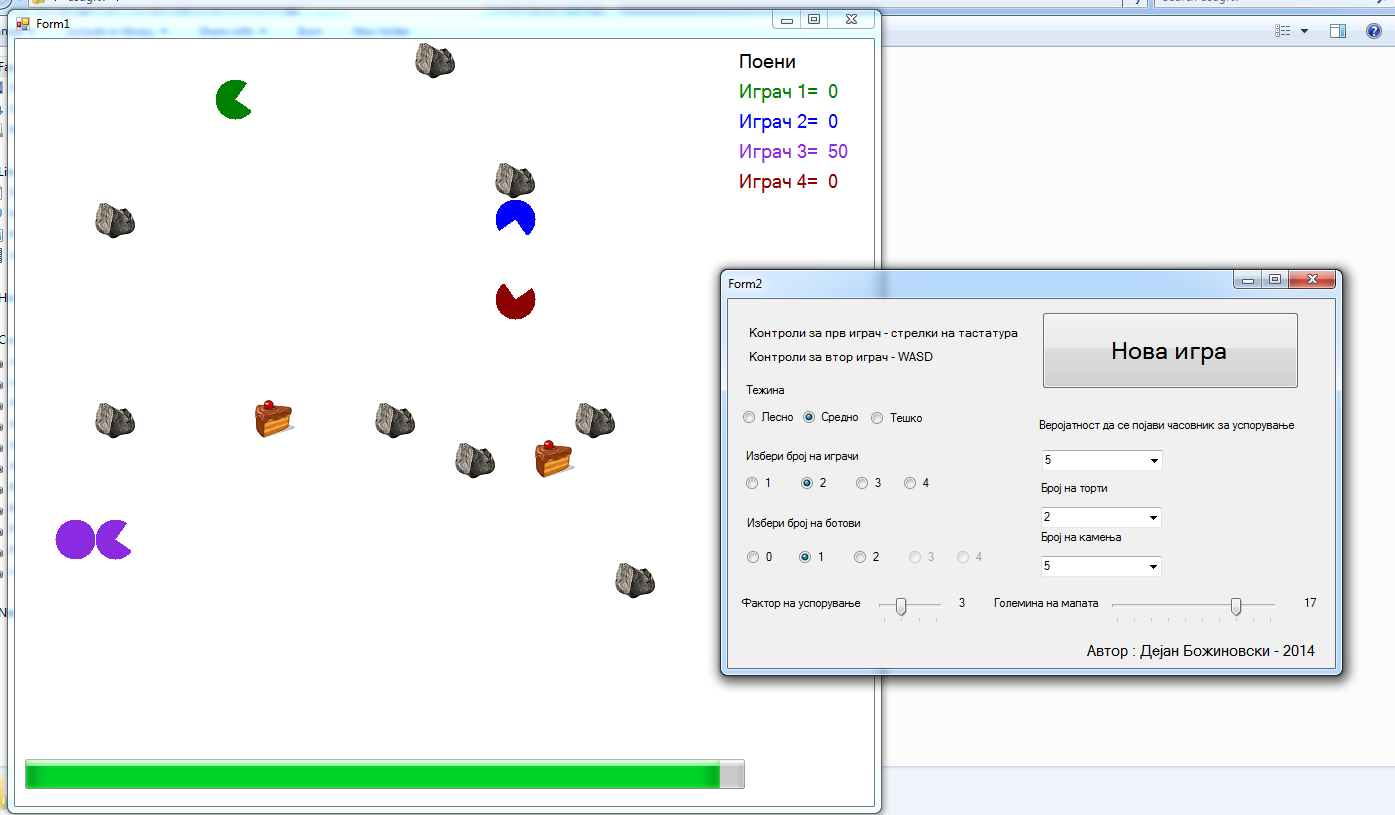
Ботовите го пресметуваат патот со вештачка интелигенција до тортите на следниов начин: почнувајќи од главата се пушта BFS алгоритам во 3 правци, не во 4 бидејќи змиите не можат да одат наназад. Од тие положби понатаму ги разгледува сите 4 правци. При ширењето на BFS, внимава да не се удри во себе, или да не се удри во друга змија. Со Design Pattern кодирање, ова е овозможено на многу лесен начин, сите податоци за змиите се ставаат во истата матрица, и една змија ги гледа телата на другите не разликувајќи ги која на која припаѓа. Ако налета на поле кадешто се наоѓа тело на некоја змија, не прави проверка дали тоа поле припаѓа на истата змија, од своето тело, или на некоја друга. Зафатено поле значи зафатено, и треба да се избегне. При пребарувањето низ мапата внимава и да не се удри во некои од камењата.

Сето ова работи добро, но не доволно добро. За да бидат поинтелигентни змиите, секоја змија одделно, ја предвидува наредната положба кај што ќе се наоѓа секоја друга змија, и се труди, ако стигне на време, да и се испречи на патот на другата, или па да не се случи во наредниот чекор двете да стапнат во исто место и да се удрат со главите.

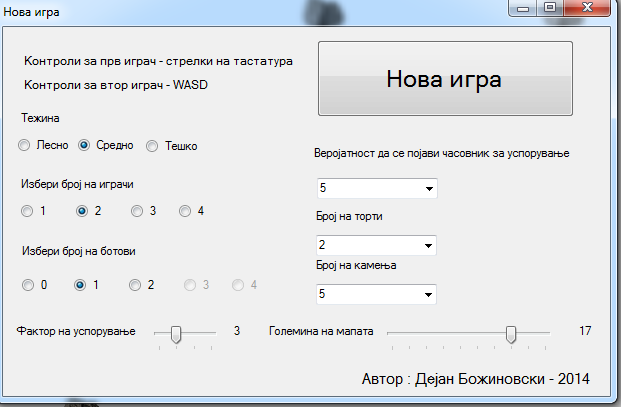
Друга паметна особина на змиите е што го наоѓаат патот до најкратката торта, ако ги има повеќе во еден момент на мапата.

Трета паметна особина е што змиите, при секој чекор, постојано одново го пресметуваат патот до тортите, за во случај ситуацијата на мапата да се сменила, на пример друга змија и се испречила на патот, или се појавила торта на некоја поблиска позиција. Овој факт овозможува за оптимизирање во имплементацијата, што не мора да го чуваме целиот пат по која ќе се движи, туку ни се потребни само 2 бита - да чуваме во кој од четирите правци треба понатаму да се движи.

Во момент кога ќе се појави тајмер, сите змии се нафрлаат кон него да го изедат. Тајмерот се поставува на најголем приоритет пред тортите, бидејќи тој има клучно значење за победата, да се соберат повеќе поени.



При стартувањето на играта, ни се појавуваат два прозорци. Едниот служи за избирање на конфигурациите за играта која ќе ја играме, а на другиот прозор, во позадина, можеме да видиме на поголема мапа како 4 змии водени од вештачка интелигенција играат една против друга. Оваа автоматска игра, кога ќе и дојде крај, нивото одново се рестартира од почеток, со нови random вредности за позициите, поради тоа секоја нова игра е различна од претходната. Ова автоматско повторување трае бесконечно се додека не започнеме ние нова игра од другиот прозорец.



Од прозорчето за нова игра, можеме да ги видиме контролите за двајцата играчи. Може да избереме тежина - ова се однесува на брзината со која ќе се одвива играта.

Бирање број на играчи може да е меѓу 1 и 4. Колку ќе бидат човечки, а колку вештачки од компјутерот зависи од избраниот број на ботови. Ако се избрани 2ца играчи, можноста да има повеќе од 2 играчи е оневозможено да се избере.

Исто така, ако избереме вкупно 3 или 4 играчи, не можеме да имаме 0 ботови, бидејќи моментално играта нема поддршка за повеќе од 2 човечки играчи. Поради тоа контролата е оневозможена да се избере 0. Меѓутоа, поради добриот начин на имплементирање на играта, користејќи Coding Design Pattern, со многу мала промена во кодот, брзо и лесно може да се додаде можност и за трет играч, но и за повеќе од 4 играчи.

Фактор на успорување- со оваа лента се избира колку придвижувања ќе биде успорувањето за змиите кога некоја змија ќе изеде тајмер.

Можеме да избереме веројатност колку често да се појавува тајмер на мапата, препорачлива вредност е 5%, иако може да е 0% - 30%.

Може да ја избереме димензијата на мапата, најмалку 10, за да има место да се движат змиите, и најмногу 19, бидејќи инаку прозорецот нема да го собира на екранот.

Најдоле можеме да видиме прогресбар, кој означува уште колку време е останато до крајот на играта.

Во горниот десен агол можеме да ја видиме моменталната ситуација со поени, со секоја изедена торта змијата добива +50 поени. Поените на секоја змија се во боја кои и соодветствуваат на бојата на змијата.

Кога ќе заврши играта, ни се појавува прозорче кое го означува крајот и на кое е напишано победникот во таа рунда- играчот со најголем број поени.

Со притискање ОК повторно ни се отвора прозорчето за избирање нова игра.