**УЧЕНИЧЕСКИ ИНСТИТУТ ПО МАТЕМАТИКА**

**И ИНФОРМАТИКА**

**ПЕТНАДЕСЕТА УЧЕНИЧЕСКА КОНФЕРЕНЦИЯ**

**УК’15**

**ТЕМА НА ПРОЕКТА**

**"Шаблонът" – триизмерна логическа развлекателна игра**

**Автори:**

**Атанас Великов Орманов, ПМГ „Иван Вазов”, Димитровград, 10 клас**

**Никола Христов Тодев, ПМГ „Иван Вазов”, Димитровград, 10 клас**

**Научен ръководител (консултант):**

**Снежа Делчева Димитрова, ст. Учител по информатика и ИТ, ПМГ „Иван Вазов”**

1. **Увод**

Много голяма част от игрите в днешно време са с чисто развлекателен характер. Друга част са с изцяло образователна цел. Целта на нашия проект е да съчетаем тези две крайности в игра, която е развлекателна, но същевременно включва мисловна дейност. Приложните умения в играта ни са:

* развиване на логическото и разумно мислене у играча
* подтикването на играча да мисли нестандартно, когато се открие в задънена улица

1. **Изложение**

The Pattern е логическа, развлекателна игра, чиято цел е едновременно да развие последователното мислене и да накара играча да мисли нестандартно. Избрали сме действието да се развива в случайно генериращ се лабиринт. Перспективата в играта е от първо лице, защото по този начин играещия е въвлечен в случващото се и, съответно, му е най-интересно. За разработването на проекта сме избрали програмите Unity за цялостния дизайн и механики на играта и Blender за създаване на модели и текстури. В бъдеще мислим да разширим възможностите на играта и да добавим още логически задачи.

Вдъхновение за играта черпим от поредицата фантастични новели “Хрониките на Амбър” от американския писател Роджър Зелазни. В книгите част от действието се развива в мрежовиден лабиринт на име “The Pattern”, който позволява на героя да се движи между светове. Също така идеята за решаване на логически задачи с цел да помогнат на играча да мисли нестандартно идва от играта “Antichamber”. Зя цялостния дизайн са помогнали игри като “Tower of Guns”, “Paranautical Activity” и “Ziggurat”.

1. **Основни етапи в реализирането на проекта**

Първо създаваме програмния код, след което го имплементираме в играта. След като сме се уверили, че работи, оформяме графичния дизайн.

1. **Функционално развитие**

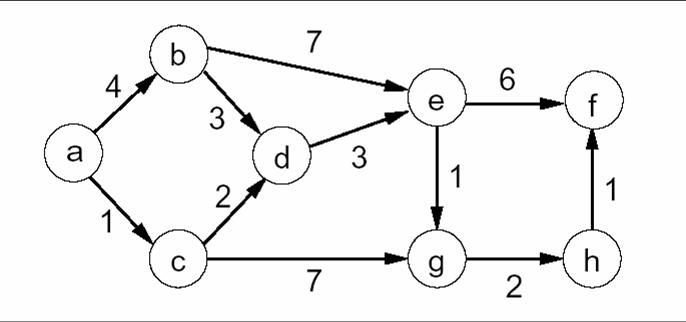
**Играта е съставена следните части:**

* графична – тази част включва 3D модели, текстури, материали и фоневе.
* програмна – тази част включва C# и JavaScript.
* game engine – в тази част се редактират обектите и тяхното позициониране в играта и се добавят текстурите им, също така се добавят кодовете, изпълнаващи различни действия, свързани с тях.

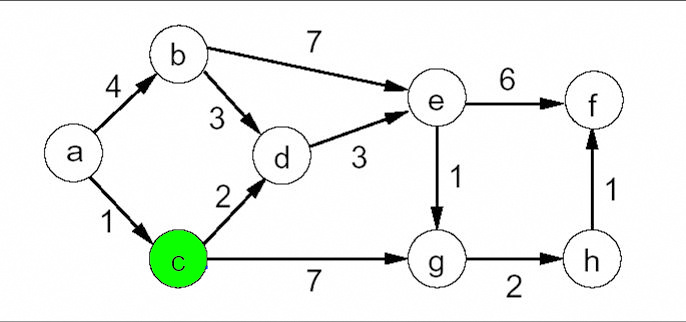
**Генериране на лабиринта:**

Създаването на лабиринта става чрез алгоритма на Прим, който работи по следния начин:

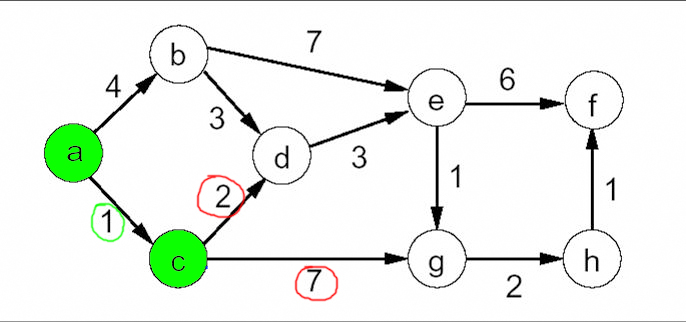
Първоначлно имаме графика от точки с пътища помежду си, като съответно всеки път си има дължина(стойност).

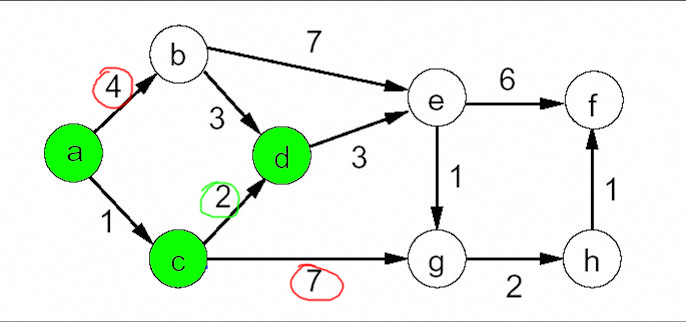


Целта на алгоритъма е да премине през всички точки по възможно най-краткия път без да преминава повторно през точките. Първо избира една точка от която да започне, например „c“. Добавя я към групата N. Когато всички точки участват в N пътя ще е завършен.

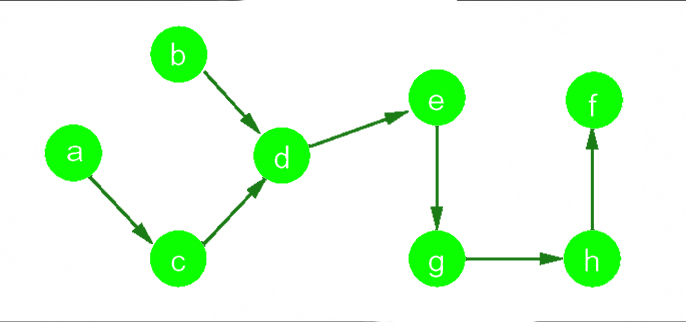


След това избира следващата точка с условията че: е съседна на „c“; пътят до нея е най-малък; не е част от N. Съответната точка е „a“ и тя бива добавена към N.



Избирането на следващата точка е на същия принцип както при избирането на „a“, но избора на точка не е между съседните на някоя, между съседните на всички точки които участват в N.

След като алгоритъма обходи клетките и пътят е завършен, крайният резултат ще изглежда така:



Ние приложихме този алгоритъм в генерирането на лабиринт като първо създадохме една мрежа от кубчета, които ще наричаме клетки. След това към всяка клетка прикрепихме по едно случайно число. Генерирането на пътя, по който играчът ще минава по-късно, започва като добавя първата клетка към списъка, който следи кои клетки са част от начертания път. И продължава, като добавя клетки в зависимост от тяхното число. След като алгоритъма на Прим си свърши работата, повдигаме клетките, които не са част от начертания път. Правейки това вече имаме стени и път по който играчът да преминава.

1. **Реализация**

За реализацията сме използвали главно две програми:

* **Unity 3D** e game engine който е създаден за да бъде достъпен за начинаещи и независими разработчици на игри. Интерфейсът и способностите му са лесни за научаване и същевременно имат достатъчно напреднали функционалности. Ние го използваме точно по тези причини и защото е безплатен.

MonoDevelop е вградената среда за програмиране в Unity, предлагаща разширени библиотеки специално за разработка на игри. Дава възможност за писане на езиците - C#, JavaScript и Boo.

* **Blender** представлява приложение, което позволява създаването на широк обхват от 2D и 3D съдържание. Опциите, които програмата предлага са: 3D моделиране, UV разгъване, добавяне на текстури и материали, добавяне на кости и раздвижване, използване на физика, провеждане на различни видове симулации и анимиране. Предимствата на Blender пред останалите програми за графична обработка, като Maya и Autodesk, е че е безплатен, с отворен код и има удобно групиран потребителски интерфейс.
  + - 1. **Контроли и упътване**

Контроли:

* „W“ – напред
* „S“ – назад
* „A“ – наляво
* „D“ – надясно
* Мишка – оглеждане
* „E“ – хващане/пускане на обектю
* „Esc“ – пауза

Упътване:

* За да преминете лабиринта трябва първо да съберете определен брой предмети, изписан в горния десен ъгъл на екрана. Те са разпръснати из нивото. След това ще се появи червен лазер, който ще ви опъти до края.
* Когато бъдете изпратени на задачата трябва да наредите четирите форми в подходящите за тях дупки. След което цилиндърът по средата ще се спусне на нивото на земята и ще можете да преминете обратно към лабиринта.

1. **Заключение**

Резултатът е една развлекателна с наблягане на нестандартното мислене игра подходяща за всички възрасти. В бъдеще мислим да развием точно това като добавим още логически задачи и главоблъсканици. Също така да измислим уникални разновидности на лабиринта.