



## **CU6051ES - Artificial Intelligence**

### **Course work II**

**Group Members:**

#	Student ID	Name
1.		Chamith Chathuka Wickramarathna
2.		Sriyantha de Silva

## Declaration

**Module: CU6051ES**

**Deadline: 30/05/2017**

**Module Leader: Mrs Mahesha Thejani**

## PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(signature: ) \_\_\_\_\_ Date: 18/07/17

(signature: ) \_\_\_\_\_ Date: 18/07/17

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples...

- (i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.
- (ii) Falsifying data in experimental results.
- (iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
- (iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.
- (v) Collusion to present joint work as the work solely of one individual.
- (vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.
- (vii) Other conduct calculated to secure an advantage on assessment.
- (viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " $e = mc^2$  (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

## Contents

Declaration .....	2
Introduction .....	4
Problem.....	5
Common Applications of the Shortest path problem.....	6
Pseudo Code implementation of Dijkstra Algorithm .....	8
Implementation of the Dijkstra's shortest path algorithm.....	9
Applications of Dijkstra's algorithm.....	11
Roll Ball Follower Game .....	12
Roll Ball Follower Game .....	13
Introduction.....	13
Scripting language Used .....	13
Tools used for the development.....	13
NavMeshAgent.....	16
Start .....	17
Update.....	17
Lessons Learned.....	18
References .....	19

## **Introduction**

Artificial intelligence in simple definition the theory and the development of computer or related systems to perform tasks normally that are required human intelligence or more than the normal human intelligence, applications of artificial intelligence through the systems are visual perception, intelligent decision-making like in expert systems, speech recognition using natural language processing and the translation of human language to another language. Poole and Mackworth (2011, p. 03) state “Artificial intelligence, or AI, is the field that studies the synthesis and analysis of computational agents that act intelligently.”

This documentation describes the Shortest path problem and as for solution for the Shortest path problem authors suggests Dijkstra’s Shortest path algorithm. Sample Implementation of algorithm done using the C# language and the implementation of the algorithm using simple game developed using Unity 4.7 game engine.

## Problem

In graph theory finding a problem of finding the shortest path from a one node to all other nodes in the graph, node is known as the vertex in graph theory. This problem is known as Single-Source Shortest Path Problem. Find a shortest path from a source vertex  $v$  to all other vertices in the graph. Path length is sum of the weights of edges on the path. The vertex placed where path begins known as Source Vertex and the Vertex placed at the ends is the destination is known as the destination vertex.

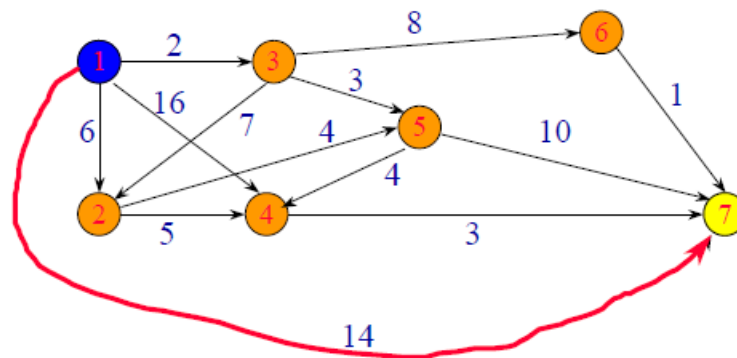


Figure 1 Path length calculation

Path from 1 to 7. Path length is 14

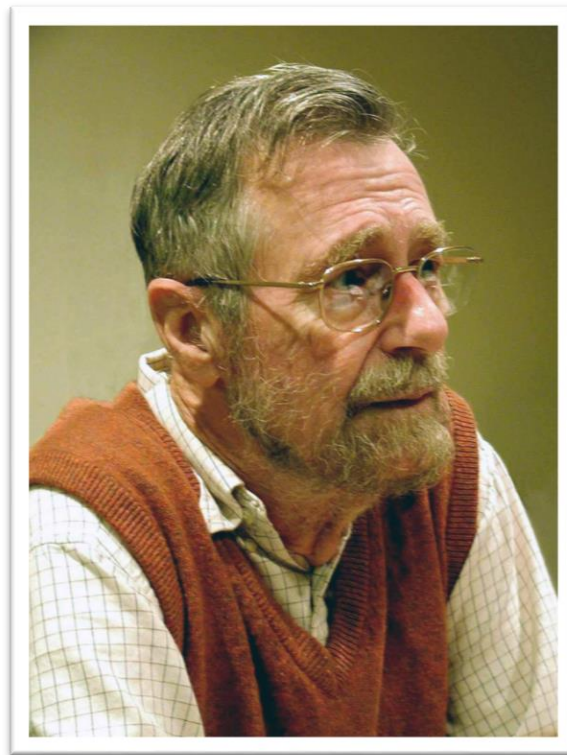
## Common Applications of the Shortest path problem

Most common application of the shortest path problem is when finding the closest way to find the from a starting point and destination of a map. Improving the data packet routing in internet needs to be done in the shortest path from server to the client machine.

found. There are various algorithms are available to solve the shortest path problem. Each has own pros and cons. Dijkstra's algorithm is the most popular algorithm for solving the Shortest path problem. A\* Search algorithm is also another powerful single pair shortest path finding algorithm due to the use of heuristics. Floyd-Warshall algorithm solves all types of shortest paths (Cherkassky, Goldberg and Radzik, 1996).

Since the selected problem only addresses the single source shorts path problem, authors have decided to use the Dijkstra's shortest path algorithm to solve the problem.

Edger Wybe Dijkstra was a Dutch computer scientist who was born in 11th May 1930 and he has received the A. M. Turing award widely considered the most prestigious award in computer science. He was died in 6th August 2002



*Figure 2 Edger W. Dijkstra*

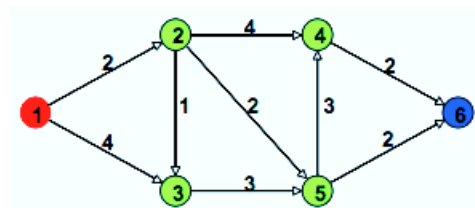
Both directed and undirected graphs can be solved using the Dijkstra's algorithm.

To solve the problem using the Dijkstra's algorithm must satisfy the below conditions.

- Graphs must be connected.
- All edges must only have positive numbers.

Original algorithm outputs value of shortest path

not the path itself with slight modification, we can obtain the path



*Figure 3 Shortest Path Algorithm*

## Pseudo Code implementation of Dijkstra Algorithm

```
dist[s] ← 0
for all v ∈ V - {s}
    do dist[v] ← ∞
S ← ∅
Q ← V
while Q ≠ ∅
    do u ← mindistance(Q, dist)
        S ← S ∪ {u}
        for all v ∈ neighbors[u]
            do if dist[v] > dist[u] + w(u, v)
                then d[v] ← d[u] + w(u, v)
return dist
```

(distance to source vertex is zero)

(set all other distances to infinity)

(S, the set of visited vertices is initially empty) (Q, the queue initially contains all vertices)

(while the queue is not empty)

(if new shortest path found)

(set new value of shortest path)

(if desired, add traceback code)



## Implementation of the Dijkstra's shortest path algorithm

Below is the implementation from C# language

```
using System;
using System.Collections.Generic;

namespace Dijkstras
{
    class Graph
    {
        Dictionary<char, Dictionary<char, int>> vertices = new Dictionary<char, Dictionary<char, int>>();

        public void add_vertex(char name, Dictionary<char, int> edges)
        {
            vertices[name] = edges;
        }

        public List<char> shortest_path(char start, char finish)
        {
            var previous = new Dictionary<char, char>();
            var distances = new Dictionary<char, int>();
            var nodes = new List<char>();

            List<char> path = null;

            foreach (var vertex in vertices)
            {
                if (vertex.Key == start)
                {
                    distances[vertex.Key] = 0;
                }
                else
                {
                    distances[vertex.Key] = int.MaxValue;
                }

                nodes.Add(vertex.Key);
            }

            while (nodes.Count != 0)
            {
                nodes.Sort((x, y) => distances[x] - distances[y]);

                var smallest = nodes[0];
                nodes.Remove(smallest);

                if (smallest == finish)
                {
                    path = new List<char>();
                    while (previous.ContainsKey(smallest))

```

```

        {
            path.Add(smallest);
            smallest = previous[smallest];
        }

        break;
    }

    if (distances[smallest] == int.MaxValue)
    {
        break;
    }

    foreach (var neighbor in vertices[smallest])
    {
        var alt = distances[smallest] + neighbor.Value;
        if (alt < distances[neighbor.Key])
        {
            distances[neighbor.Key] = alt;
            previous[neighbor.Key] = smallest;
        }
    }
}

return path;
}
}

class MainClass
{
    public static void Main(string[] args)
    {
        Graph g = new Graph();
        g.add_vertex('A', new Dictionary<char, int>() {{'B', 7}, {'C', 8}});
        g.add_vertex('B', new Dictionary<char, int>() {{'A', 7}, {'F', 2}});
        g.add_vertex('C', new Dictionary<char, int>() {{'A', 8}, {'F', 6}, {'G', 4}});
        g.add_vertex('D', new Dictionary<char, int>() {{'F', 8}});
        g.add_vertex('E', new Dictionary<char, int>() {{'H', 1}});
        g.add_vertex('F', new Dictionary<char, int>() {{'B', 2}, {'C', 6}, {'D', 8}, {'G', 9}, {'H', 3}});
        g.add_vertex('G', new Dictionary<char, int>() {{'C', 4}, {'F', 9}});
        g.add_vertex('H', new Dictionary<char, int>() {{'E', 1}, {'F', 3}});

        g.shortest_path('A', 'H').ForEach( x => Console.WriteLine(x) );
    }
}
}

```

Sample data added to check the output of the algorithm implementation.

## Applications of Dijkstra's algorithm

- In Algorithmic trading in financial sector deals with trading assets and goods to obtain arbitrage opportunity. This can be done with any assets between any markets.
- convert 1000 USD to 950 EUR and then 950 EUR to 1020 CAD which you convert back to 1007 USD :) Just by converting from currency to currency you can make money.
- Find the shortest path between two locations in location and navigation related applications. Ex – google Maps Navigation
- Uber cost estimation between shortest paths
- Solving the word ladder puzzle.
- Given a network of computers, find the shortest path from machine A to machine B
- In video games, these algorithms are frequently used to find the shortest path between two points on a map. "Pathfinding," as it is called in this context, can be used by AI to plot routes, or by the game engine to assist users in plotting routes.
- In military drones which used cannot be used with radio signal controlling.

## **Roll Ball Follower Game**

### **Tools Used for the development of the Game**

Developer was at the beginning completely newbie for the graphical game development. Developer did a survey on current game engines and their suitability for the project. There were several choices for the developer to develop the Game to present the implementation of the Dijkstra's Shortest path algorithm. As a result, developer found the below game engines for each language.

#### **Unity 3D**

Unity game engine was the most affordable option for the Game developer according to the survey done. Also, it helps developer to script the game using C#, JavaScript for Unity (Unity Script) and Boo language. There was great community support for the Unity, so the developer had plenty of tutorials to learn the Unity language. Also, the unity assets store was helpful for the developer even though the developer did not use assets store for this project, it is a considerable help for developers to download preloaded character and environment models.

#### **Unreal Engine 4**

This is the most popular engine for large scale game development, Developers were provided with free unreal engine access with the student membership facilities provided by the London Metropolitan university with the GitHub student pack. Unreal was free for to use. Coding needs to be done using C++ language, which is still unfamiliar with the developer. According to the survey results Unreal engine has the largest community support and the vast diversity of platform. support. Because of these reasons many Award-winning games were developed with Unreal Engine 4.

#### **Cry Engine 3**

This also an industry level game engine used for game development with community support. Easiest implementation of Ai also provided by Cry Engine.

#### **JMonkey Engine**

JMonkey scripting part everything can be done using Java Language, which is most familiar language of the developer but the lack of assets and community support for the JMonkey engine developer did not use the JMonkey engine. This is an open source project which is hosted on GitHub.

#### **Sparrow**

Sparrow was game engine for iOS, and it has considerable community support and learning tutorials, developer did not choose it due to the lack of platform support in Sparrow Engine.

Rage engine, Project Anarchy, Game maker studio are some other game engines found by user in the survey.

## Roll Ball Follower Game

### Introduction

Game was developed to represent the implementation of shortest path algorithm by Dijkstra. This is a simple game developed using Unity Game engine 4.7, There are spears in the ground with implemented physics objects like gravity which is provided with Unity. One sphere must be controlled by the Game player and that sphere has speed of 10 points over the AI implemented sphere which has only 5 speed points. When user moving the player object with Key board Ai Sphere automatically calculate the shortest path for the Player sphere and it automatically rolls to the position of the player.

### Scripting language Used

Developer used C# language to script the game behaviours and the implementation of the Ai facility in the Game. For the camera controlling classes and the Player controlling classes also implemented using the C# language.

### Tools used for the development

Unity Editor 4.7 used for the application development of 3D development. It provided enough assets to implement the main idea of shortest path algorithm. Developer selected web player as the platform the Game, but it can be changed at any stage.

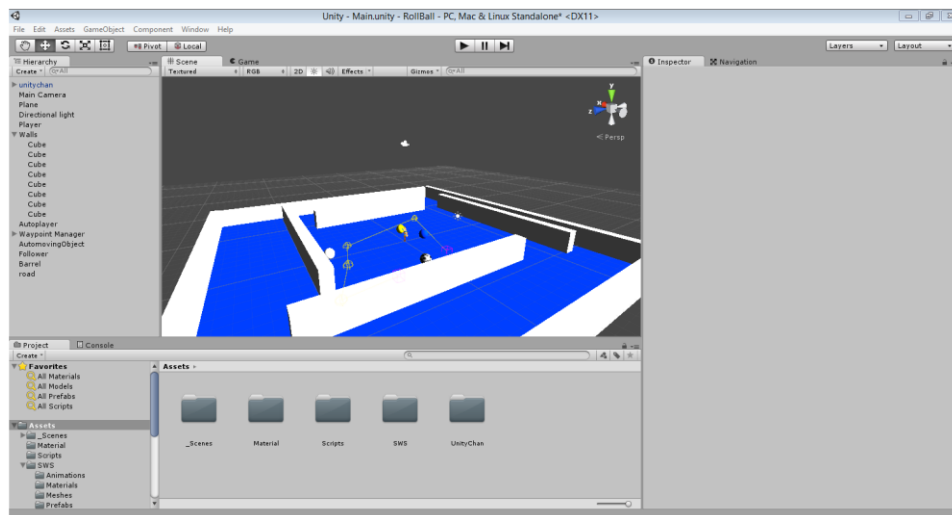


Figure 4 Unity 4.7 Editor

For the development of the task user has placed one ambient light and one Camera which is controlled by MyCameraController class which is scripted using C# to follow the player object in the Game View. MyCameracontroller class implemented to follow the player object to have a clear view. There was Plane object placed as the ground in Blue colour and material to represent the ground where the game play happens. Surrounded by 4 walls which are implemented using cubes in unity.

Player object and AI objects were implemented using Sphere objects. To identify the player and object Player object was applied with yellow colour material. Ground colour implemented with blue colour material.

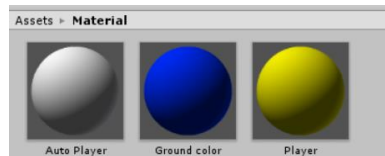


Figure 5 Material

Only one scene was used for entire program. Scene named as Main scene.

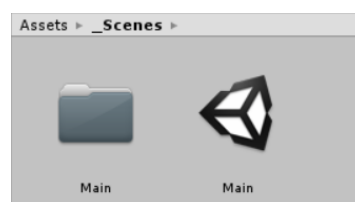
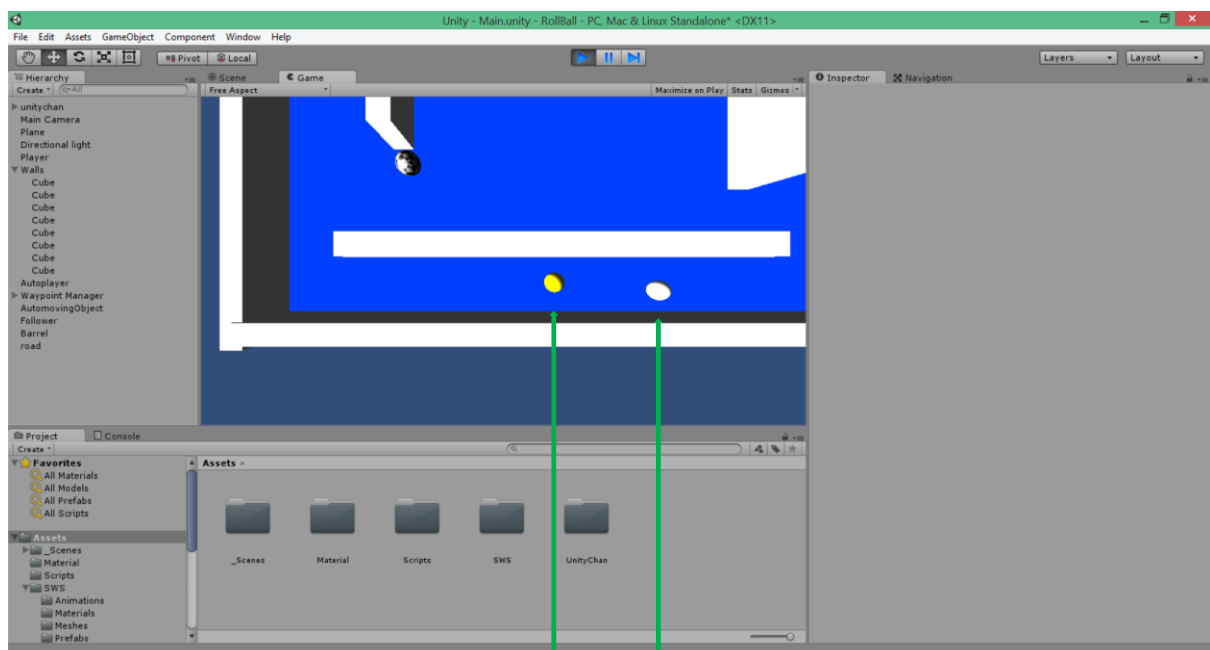


Figure 6 Scene

During the game play mode, Yellow Sphere is used as the player controlled object, using the arrow keys. The following white ball object is the object implemented with the shortest path finding algorithm.



Player Object

AI implemented Object

For the implementation of the Ai object done easily with Unity integration within 2 lines of code.

NavMesh Agent in Unity. NavMeshAgent components helped developer to easily represent the main idea of shortest path algorithm (UnityDocs, 2017). Nav mesh agent help to create characters avoiding each other while reaching to the goal. Agent reason the about the game world using the NavMesh agent. Shape of cylinder is used to detected and respond to collisions between other agents and obstacles.

## NavMeshAgent

Is true implementation of Ai. Agent is defined by an upright cylinder. Whose size is specified by the radius and Height properties.

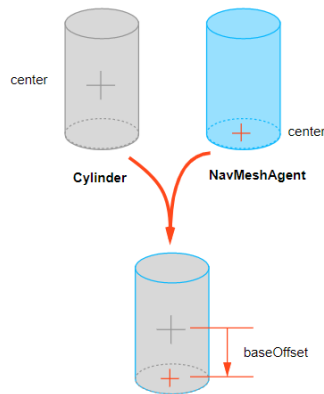


Figure 7 Nav Mesh Agent

Complete idea of the Dijkstra's shortest path algorithm was able to implement using Unity Game Engine with the following few lines of code which is used to implement the shortest path algorithm in the Follower class in the game.

```
using UnityEngine;
using System.Collections;

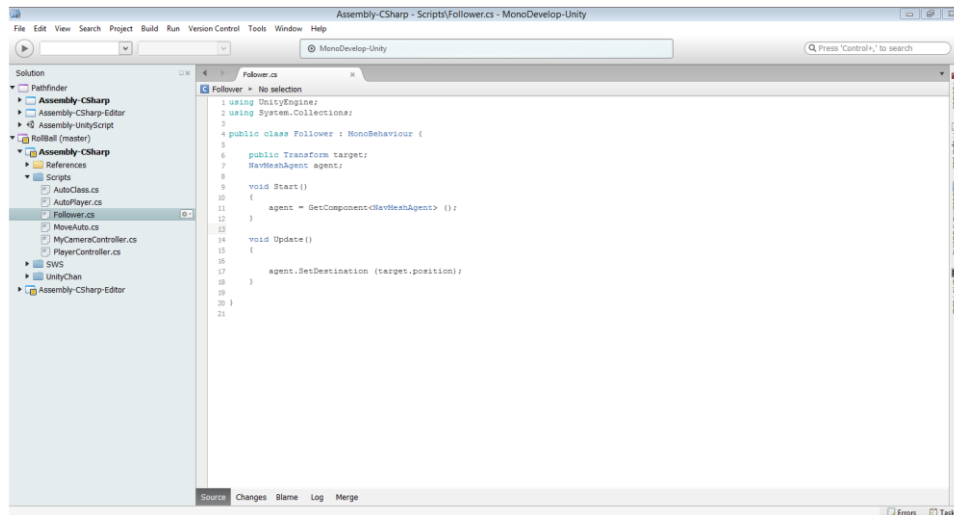
public class Follower : MonoBehaviour {

    public Transform target;
    NavMeshAgent agent;

    void Start()
    {
        agent = GetComponent<NavMeshAgent> ();
    }

    void Update()
    {
        agent.SetDestination (target.position);
    }
}
```





*Figure 8 Script Development with Mono Code*

Follower class is extended by Mono behaviour class, super class of the Follower Class is MonoBehaviour class. It is the base class for every unity script derives.

### Start

Start method called on the frame when a script is enabled just before the update method is called for the first time. Start method is overridden in the Follower class to initialise the NavMeshComponentAgent and assign that agent to reference of NavMesh.

### Update

Update method is also a method overridden in the class Follower. Update method called every frame loaded to the system. This method is invoked each time when new frame loaded.

## Lessons Learned

At the beginning developer was a completely newbie to the graphical game development but with the help of unity online documentation and video tutorials developer was able to understand the concepts of Unity.

Unity is very good tool for game development on multiple platforms, because of the availability to script the game in multiple languages like C#, Java script for Unity and Boo language, unity game development made easier for everyone.

Since developer was comfortable with C# unity helped to develop the game as most of the other major Game engine scripts should be programmed using C++ programming language.

Developer also learned how to use the unity assets store which has prebuilt components to use some are completely free to use and some are available to purchase.

Developer has learnt there Ai techniques are already included with the Game Engines which are very easy to use methods referring to the documentation.

## References

- Cherkassky, B., Goldberg, A. and Radzik, T. (1996). Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73(2), pp.129-174.
- Cormen, T., Leiserson, C., Rivest, R. and Stein, C. (2014). *Introduction to algorithms*. Cambridge, Massachusetts: The MIT Press.
- Poole, D. and Mackworth, A. (2011). *Artificial intelligence*. New York: Cambridge University Press.