**Production System For 15 Puzzle**

**End term report**

*By*

*ARAVIND KOTTAKOTA ROLL NO:A13*

*PADAGALA SANJANA ROLL NO:B 51*

*N.V.L SUPRIYA ROLL NO:B 44*

*K.Joseph Paul RollNo:B62*

Section:- K18JE



School of Computer Science Engineering

Lovely Professional University, Jalandhar

April, 2020

Student Declaration

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

*ARAVIND KOTTAKOTA ROLL NO:A13*

*PADAGALA SANJANA ROLL NO:B 51*

*N.V.L SUPRIYA ROLL NO:B 44*

*K.Joseph Paul*

Place: Punjab.

Date: 5 April,2020.

**TABLE OF CONTENTS**

TITLE PAGE NO.

1. **Background and objectives of the project assigned………………………...4**

**1.1 Description………………………………………………………………...4**

**1.2Background…………………………………………………………….....5**

**1.3 Motivation & Outcomes………………………………………………….5**

**1.4 Goals&Objectives……………………………………………………...6**

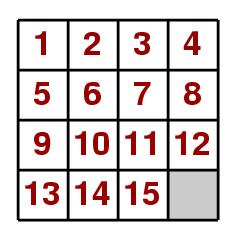
1. **Description of Project………………………………………………………...6**
   1. **Modules …………………………………………………………………..6**
   2. **Flow of project …………………………………………………………...7**
2. **Description of Work Division…………………………………………………8**
3. **Implementation of scheduled work of Project………………………………8**
4. **Technologies and Framework to be used…………………………………….9**
5. **SWOT Analysis………………………………………………………………...14**
   1. **Strengths and weaknesses…………………………………………………14**
   2. **Opportunities and threats ………………………………………………...14**
6. **Github commits/repository……………………………………………………14**
7. **Background and objectives of the project assigned**
   1. **Description:**

**Production System:**

A production system (or production rule system) is a computer program typically used to provide some form of artificial intelligence, which consists primarily of a set of rules about behaviour but it also includes the mechanism necessary to follow those rules as the system responds to states of the world.

**15-Puzzle Problem :**

The 15 puzzle problem is invented by sam loyd in 1878. In this problem there are 15 tiles, which are numbered from 0 – 15. The objective of this problem is to transform the arrangement of tiles from initial arrangement to a goal arrangement. The initial and goal arrangement is shown by following figure.

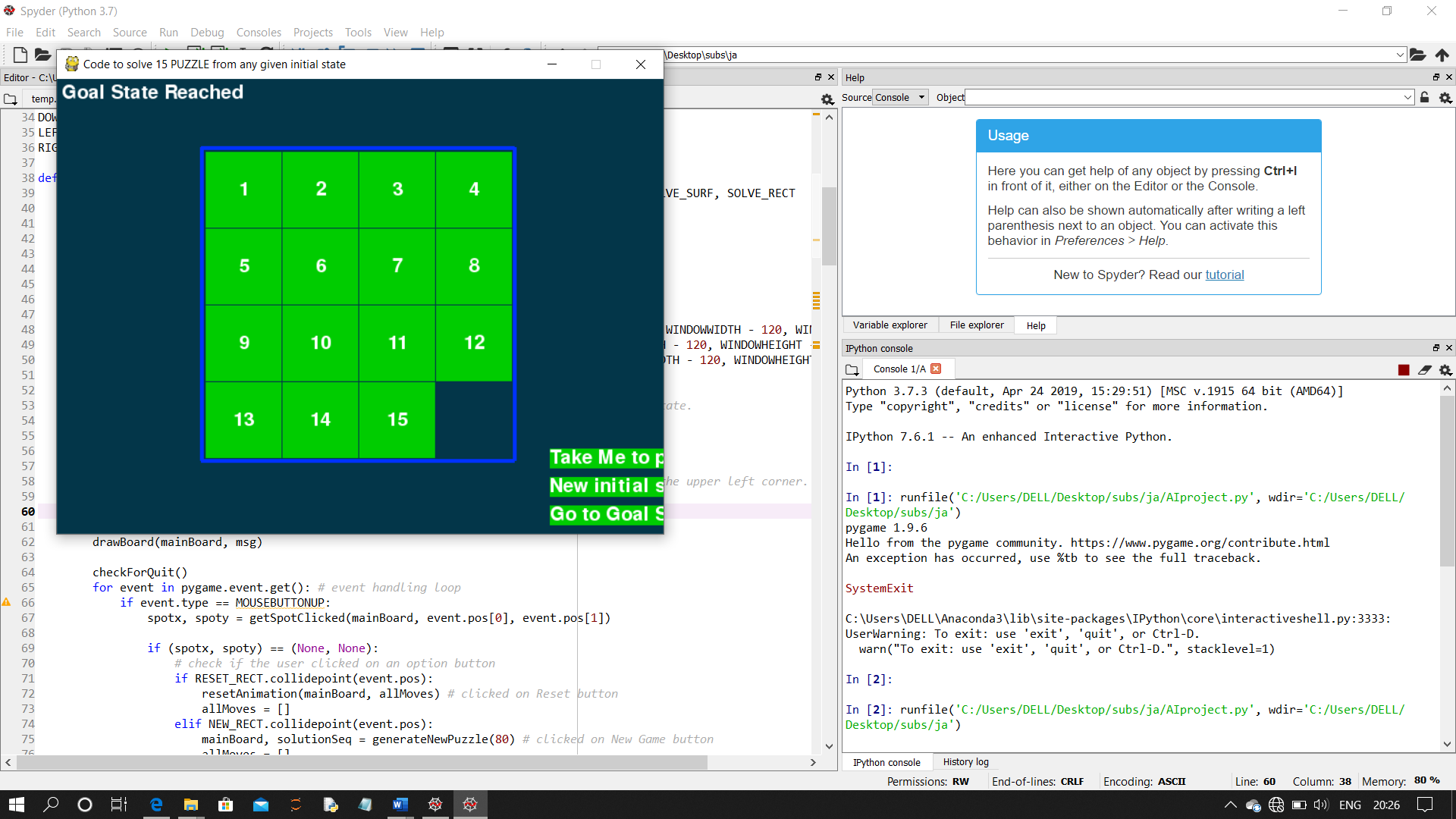
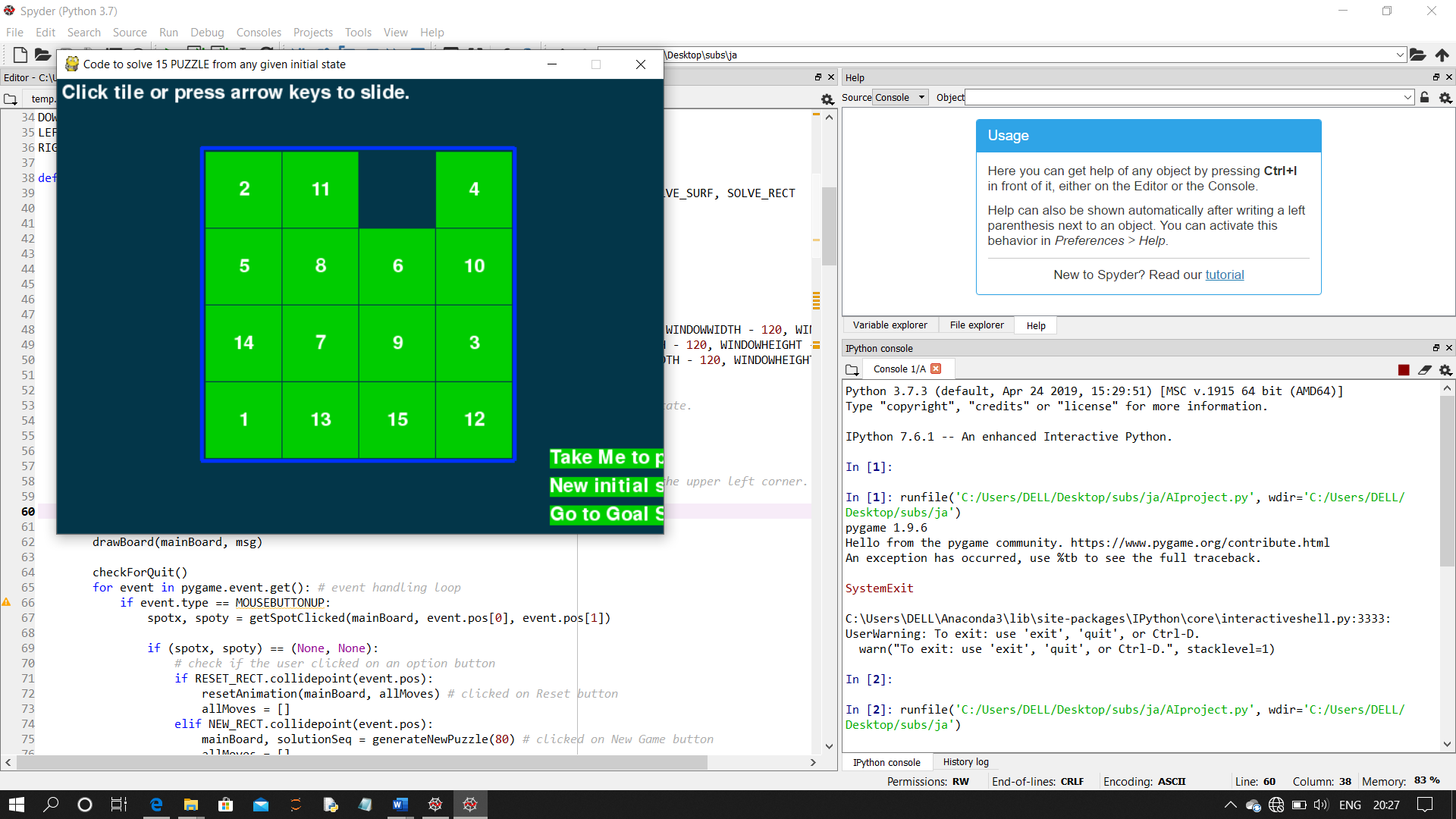


* 1. **Background:**

The basic background behind this project was to make us understand the usage of the production system where different algorithms are used and all the different flow of the data and the information is done. The idea of the project was to make us familiar with the python language the usage of the python languages in the Artificial Intelligence the various libraries which are present in python and the different functions which help us to run the code. The idea of doing this group project was to built a team work between all the students in the group due to which all the students can work in a team and the project can be done accordingly.

* 1. **Motivation& Outcomes:**

the basic motivation behind the project was to know different things and also was to create a python code on my own which can reach the goal state from any given initial state. By the motivation given by the faculty I and my group students were able to complete the work and the below given picture are the initial state and the goal state generated by my code.



**Initial State Goal State**

* 1. **Goals & Objectives:**

The goals which were achieved at the end of the project was that all my team members were able to understand the usage of different python libraries and the functioning of the production system. We also understood that the production system works on:

1. The recursive path algorithm

2. A production system working loop till the solution matches the memory pattern

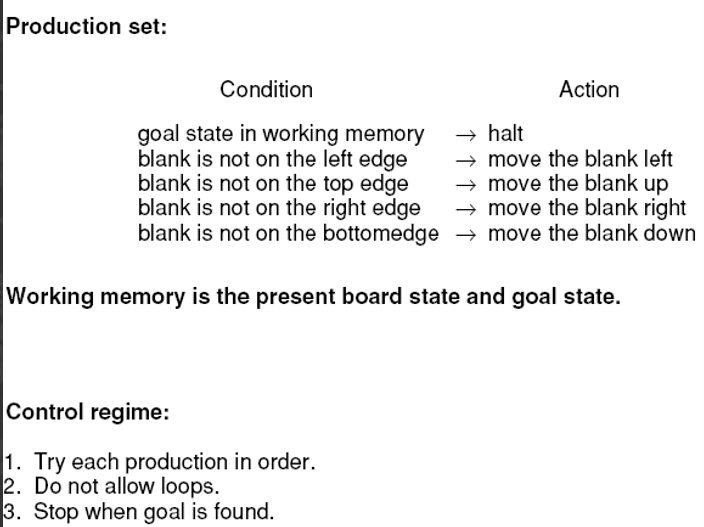
3. Trace of simple production system

4. Data Driven Search

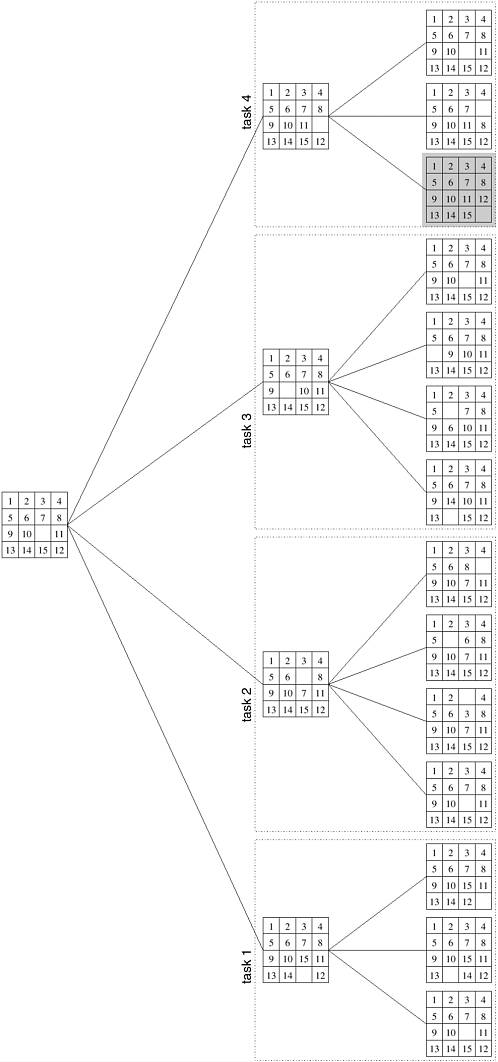
5. Goal Driven Search

1. **Description of Project:**
   1. **Modules:**

The different modules used in this project are:



Branch and Bound method:



1. **Description of work division:**

The work of this project was divided equally among all the group members the basic idea was that all the members of the group should understand the working of the production system and the process by which the 15 puzzle is solved. So, the division was done accordingly.

Kottakota Aravind………………………….. Working code

Sanjana Padagala…………………………………….Searching various algorithms

NVL.Supriya………………………………the graphical representation of the algorithm

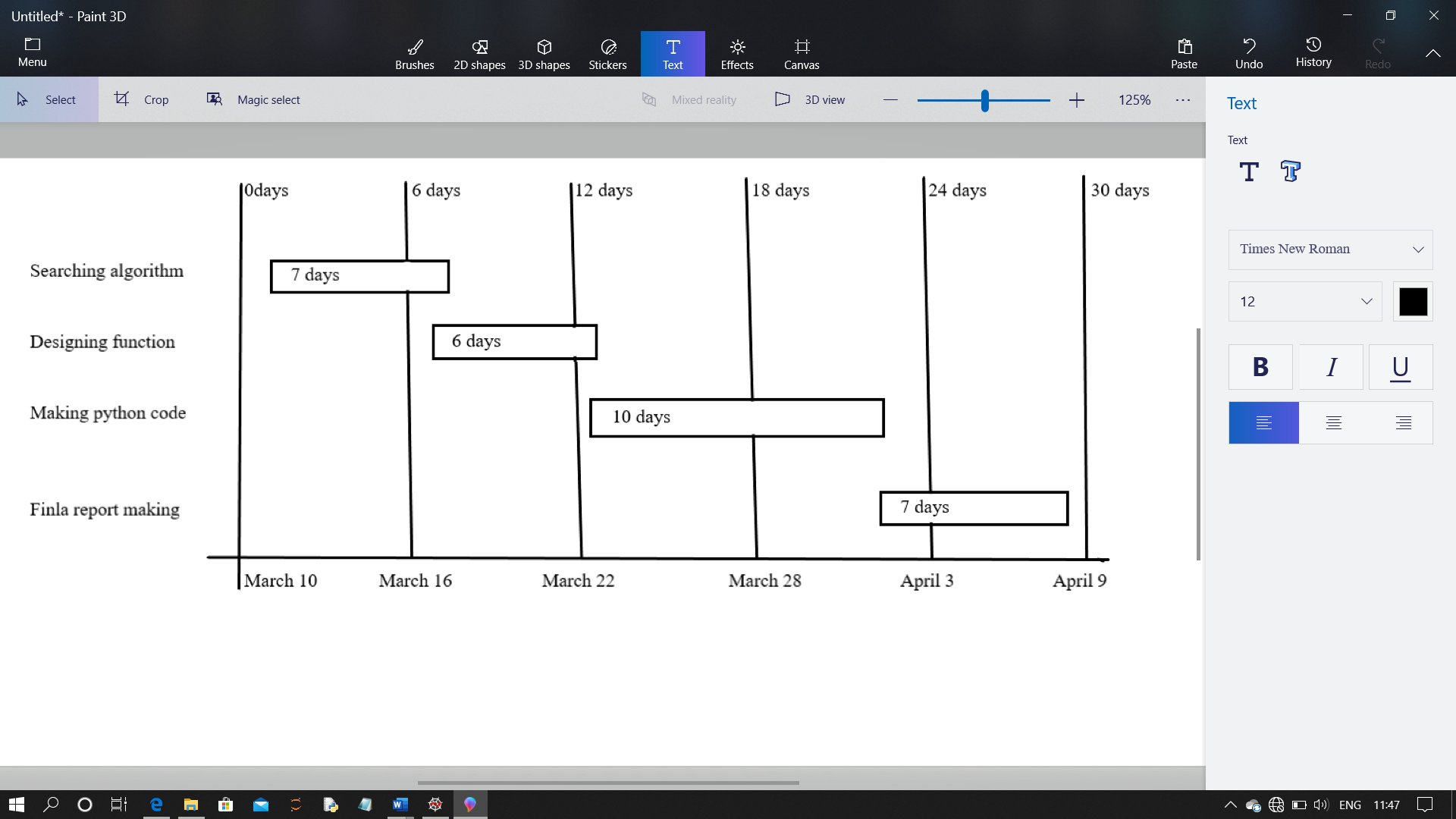
K.Joseph paul .........................working code and for algorithm.

By this division of the work everyone understood the working of the code and the production system. The final report was made by all the members of the group as it was the only thing left for the completion of the project.

1. **Implementation of scheduled work of Project:**

The implementation of the project was done by as show in below gantt chart

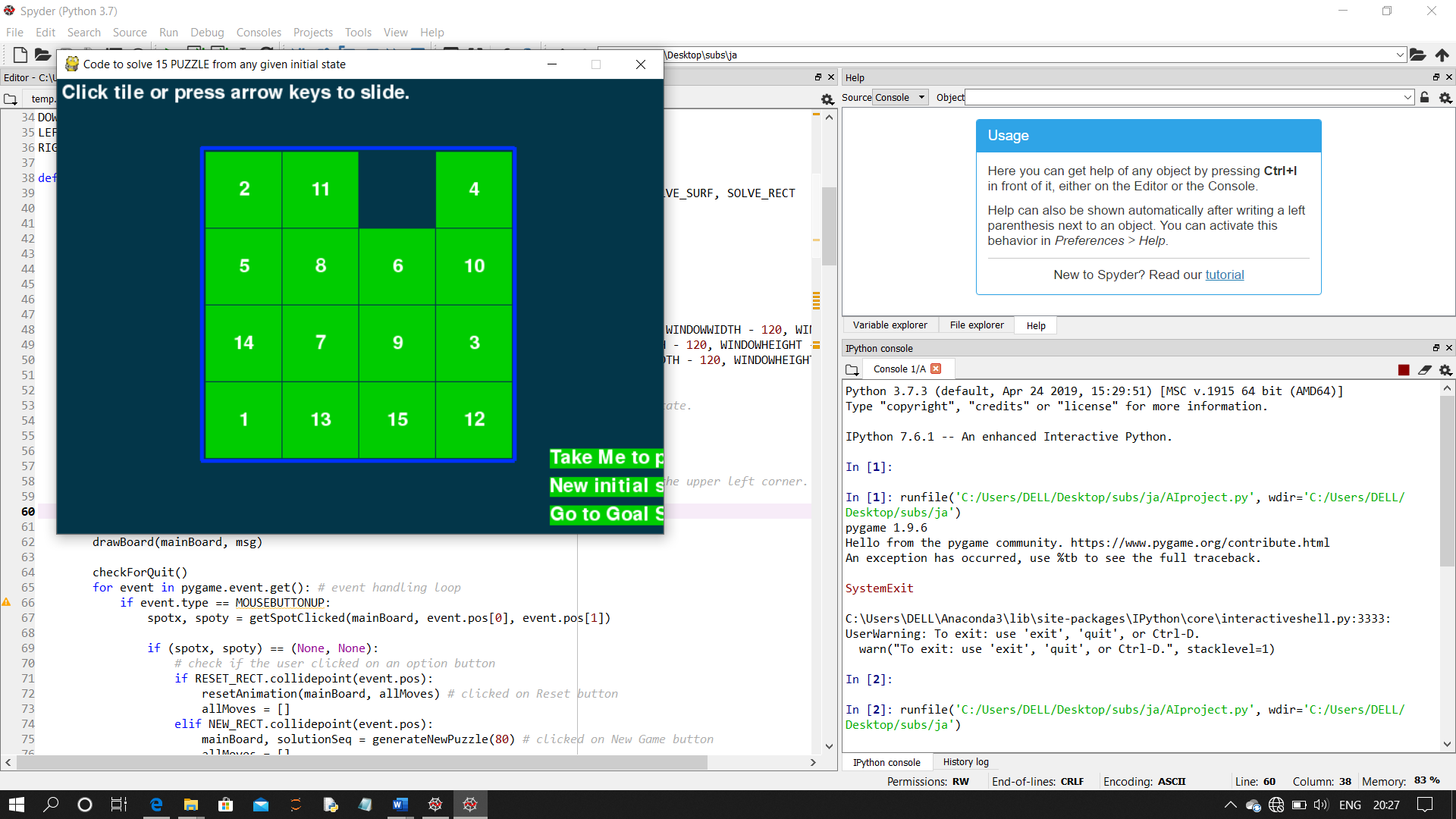
|  |  |  |
| --- | --- | --- |
| **S no** | **Name of the work** | **No of days** |
| 1 | Searching algorithm | 7 |
| 2 | Designing functions | 6 |
| 3 | Making python code | 10 |
| 4 | Final report | 7 |



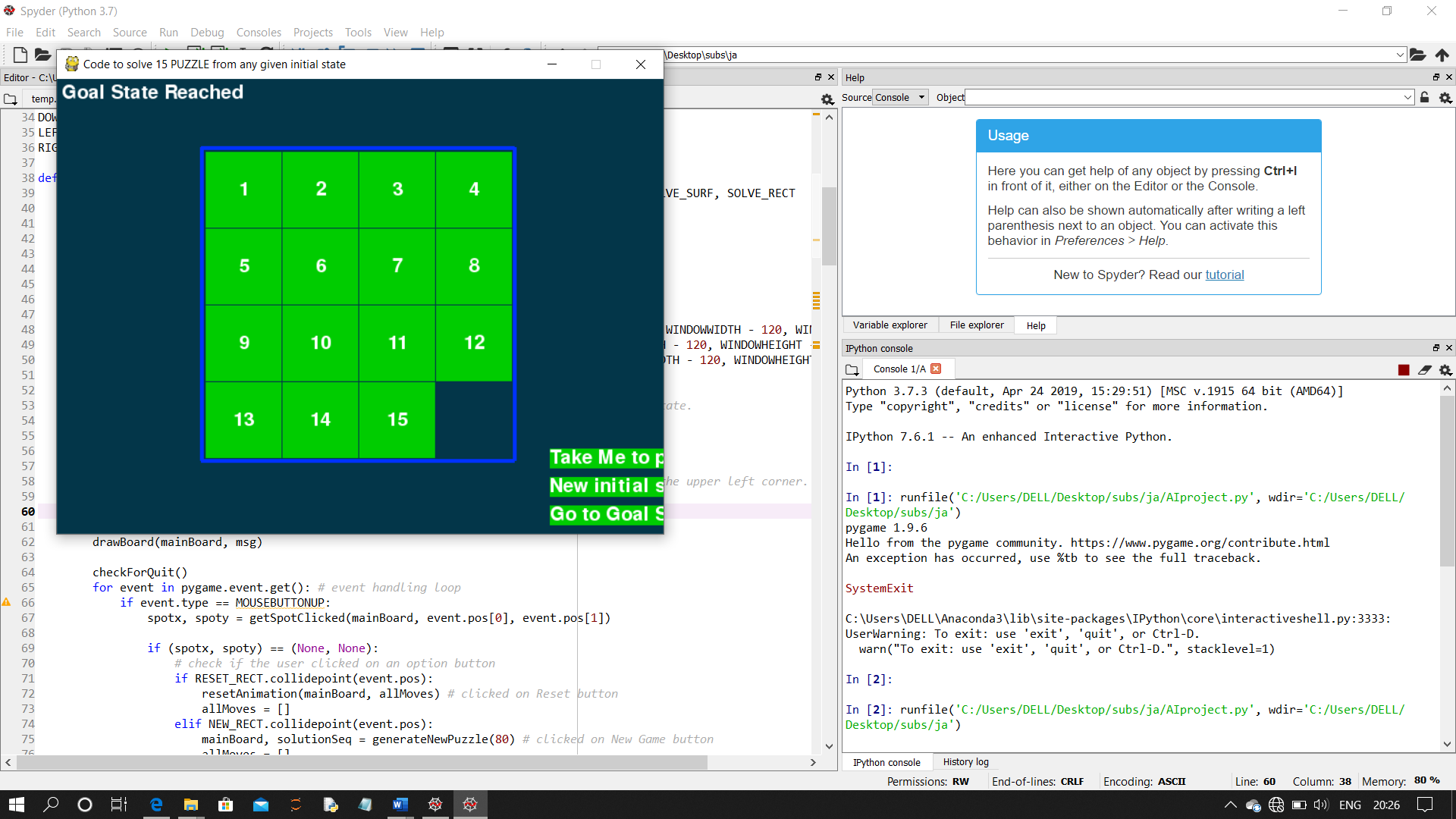
1. **Technologies and Framework to be used:**

* In the code created there is an important gaming library used which is pygame.
* The below given main function connects all the function together
* The below given is the generate new puzzle function from my python code which helps us to create a new initial state.
* The below given is the slide animation function from my python code which helps us to create the animation moves of the slides when we move to goal state.
* The below given is the initial state created by the generate new puzzle function.
* from random import randint

* class Puzzle:
* def \_init\_(self):
* self.items = {}
* self.position = None
* def main\_frame(self):
* d = self.items
* print('+-----+-----+-----+-----+')
* print('|%s|%s|%s|%s|' % (d[1], d[2], d[3], d[4]))
* print('+-----+-----+-----+-----+')
* print('|%s|%s|%s|%s|' % (d[5], d[6], d[7], d[8]))
* print('+-----+-----+-----+-----+')
* print('|%s|%s|%s|%s|' % (d[9], d[10], d[11], d[12]))
* print('+-----+-----+-----+-----+')
* print('|%s|%s|%s|%s|' % (d[13], d[14], d[15], d[16]))
* print('+-----+-----+-----+-----+')
* def format(self, ch):
* ch = ch.strip()
* if len(ch) == 1:
* return ' ' + ch + ' '
* elif len(ch) == 2:
* return ' ' + ch + ' '
* elif len(ch) == 0:
* return ' '
* def change(self, to):
* fro = self.position
* for a, b in self.items.items():
* if b == self.format(str(to)):
* to = a
* break
* self.items[fro], self.items[to] = self.items[to], self.items[fro]
* self.position = to
* def build\_board(self, difficulty):
* for i in range(1, 17):
* self.items[i] = self.format(str(i))
* tmp = 0
* for a, b in self.items.items():
* if b == ' 16 ':
* self.items[a] = ' '
* tmp = a
* break
* self.position = tmp
* if difficulty == 0:
* diff = 10
* elif difficulty == 1:
* diff = 50
* else:
* diff = 100
* for \_ in range(diff):
* lst = self.valid\_moves()
* lst1 = []
* for j in lst:
* lst1.append(int(j.strip()))
* self.change(lst1[randint(0, len(lst1)-1)])
* def valid\_moves(self):
* pos = self.position
* if pos in [6, 7, 10, 11]:
* return self.items[pos - 4], self.items[pos - 1],\
* self.items[pos + 1], self.items[pos + 4]
* elif pos in [5, 9]:
* return self.items[pos - 4], self.items[pos + 4],\
* self.items[pos + 1]
* elif pos in [8, 12]:
* return self.items[pos - 4], self.items[pos + 4],\
* self.items[pos - 1]
* elif pos in [2, 3]:
* return self.items[pos - 1], self.items[pos + 1], self.items[pos + 4]
* elif pos in [14, 15]:
* return self.items[pos - 1], self.items[pos + 1],\
* self.items[pos - 4]
* elif pos == 1:
* return self.items[pos + 1], self.items[pos + 4]
* elif pos == 4:
* return self.items[pos - 1], self.items[pos + 4]
* elif pos == 13:
* return self.items[pos + 1], self.items[pos - 4]
* elif pos == 16:
* return self.items[pos - 1], self.items[pos - 4]
* def game\_over(self):
* flag = False
* for a, b in self.items.items():
* if b == ' ':
* pass
* else:
* if a == int(b.strip()):
* flag = True
* else:
* flag = False
* return flag
* g = Puzzle()
* g.build\_board(int(input('Enter the difficulty : 0 1 2\n2 '
* '=> highest 0=> lowest\n')))
* g.main\_frame()
* print('Enter 0 to exit')
* while True:
* print('Hello user:\nTo change the position just enter the no. near it')
* lst = g.valid\_moves()
* lst1 = []
* for i in lst:
* lst1.append(int(i.strip()))
* print(i.strip(), '\t', end='')
* print()
* x = int(input())
* if x == 0:
* break
* elif x not in lst1:
* print('Wrong move')
* else:
* g.change(x)
* g.main\_frame()
* if g.game\_over():
* print('You WON')
* break



* The below give is the GUI output of the goal state from my python code.



1. **SWOT analysis:**
   1. **Strength & Weakness**

The project is capable of solving all the questions form any initial state to goal state. It can also help you understand all the functioning of production system.

* 1. **Opportunities and threats:**

This project can not solve the questions if the functions do not perform well.

1. **Github commits/repository**:

**BONAFIDE CERTIFICATE**

Certified that this project report “**Production System for 15 puzzle**” is the bonafide work of “Aravind kottakota ,sanjana padagala, NVL.Supriya , K.joseph**”** who carried out the project work under my supervision.