



IMPLEMENTATION TEAM REPORT

Hex Game

Team Members:

Mohamed Atef (Leader)	1132054
Dina Usama	1132009
Sherif Hussien	1122049

Table of contents

- [Meetings](#) [Page 2](#)
- [Bug-reports](#) [Page 4](#)
- [Playing manual guide](#) [Page 5](#)

Meetings

<i>Day/Time</i>	<i>Details</i>	<i>Duration</i>
21 March 4 pm	A meeting between design and implementation team to know what are the steps that we will work on in our design Task given to implementation team was to study the design to find the language that will suit the project	2 hours
23 March 4 pm	In this meeting the implementation team decided that the used language will be C# which will ease the process of making the interface in unity We also agreed that the code will be composed of 3 main files Stack file Board file Game algorithm file	1 hour
26 March 1 pm	In this meeting we wrote the stack file which contains the stack class	1 hour

	and its functions such as push, pop, isEmpty and top	
16 April 6 pm	Skype meeting to check progress in which we almost finished Board file and we were discussing the game algorithm	1 hour
19 April 6 pm	Skype meeting between implementation team and Ahmed Adel (Leader) to check the progress done till now	1 hour
24 April 1 pm	Integration between Board file, Stack file and Game algorithm file	1 hour
27 April 4 pm	A meeting to fix the bugs after integration	2 hours
30 April 1 pm	Bugs fixed but non optimal positions are placed, another fix suggested	1 hour
2 May 4 pm	More fixes are made to board file	1 hour
4 May 4 pm	We had to add some more algorithms to the game which required adding	3 hours

more files to the project
like Bridge file

Bug Reports

Bug File	Bug	Fix
MonteCarlo.cs	Copying tree node state wasn't correct	We implemented copy tree state function in state class
MonteCarlo.cs	Copying player node state wasn't correct	We implemented copy player state function in player class
MonteCarlo.cs	Taking too much time in simulation	We made it the simulation threaded to not take that much time in each simulation step
MonteCarlo.cs	The simulation didn't make optimal moves	We used DSU algorithm and implemented it

Playing Manual Guide

There are 3 modes to choose from

1. Player vs My agent
2. Player vs Opponent agent
3. My agent vs Opponent agent

The steps to launch the game are pretty easy

1. Choose the mode you want to play in
2. Choose if you want to enable swap move or not
3. Click on the cell you want to play in
4. Wait for the opponent to make his play
5. Enjoy the game until one of the two players win the game