

A* ALGORITHM USING MAKEBLOCK MBOT

ROBOT MOTION PLANNING

ASHIR JUDAH -

FAUZI MOHAMED -

JEREMIAH EROME - 200354951

LIST OF KEYWORDS

- - A* ALGORITHM
- - MBOT
- - PATH FINDING
- - ARDUINO C
- - HEURISTICS

INTRODUCTION

- THE MOTIVATION FOR THIS PROJECT WAS TO IMPLEMENT A ROBOT THAT WOULD REACH A GOAL
- THIS WAS CHOSEN BECAUSE WE WERE INTERESTED IN EXPLORING THE CONCEPT OF AI IN VIDEO GAMES AND ROBOTICS
- WE THOUGHT THAT AN IMPLEMENTATION OF THIS CONCEPT WOULD BE A GOOD EXPERIENCE AND BENEFICIAL IN OUR EDUCATION
- THE DESIGN OF THE PROJECT WAS TO CREATE A PATH FINDING ALGORITHM TO FIND THE GOAL ROUTE IN A VIRTUAL MAP IN ORDER FOR THE ROBOT TO MOVE TO THE GOAL

PROBLEM

- WE PLANNED TO IMPLEMENT THE MAKEBLOCK MBOT ROBOT USING A^* ALGORITHM IN ORDER TO UTILIZE THE PATH FINDING METHOD TO SEARCH FOR THE GOAL TILE FROM THE INITIAL TILE
- WE WANTED TO IMPROVE THE CONSISTENCY OF OUR SEARCH
- THE SEARCH FOR THE GOAL STATE IS DONE USING A 10 x 10 GRID AS THE MAP, AND USING HEURISTICS FROM THE A^* ALGORITHM TO FIND THE GOAL STATE
- USING THE MBOT ROBOT AND WORKING WITH ARDUINO C AND MBLOCK SOFTWARE, WE DESIGN THE PATHFINDING MAP AND ALGORITHM TO GET THE ROBOT FROM ITS INITIAL STATE TO THE GOAL STATE.

WORKING WITH MAKEBLOCK SOFTWARE & ARDUINO C

- MBLOCK 3 IS A SOFTWARE THAT IS USED TO IMPLEMENT FUNCTIONALITIES INTO HARDWARE BASED ON ARDUINO C AND SCRATCH CODE
- WE INITIALLY STARTED WITH MBLOCK 5 BUT AS WE STARTED TO USE IT, WE REALIZED THAT IT DIDN'T FIT WITH WHAT WE WERE TRYING TO USE, BECAUSE OF LACK OF MBOT REUSABLE CODE
- WE SWITCHED TO MBLOCK 3 TO CREATE PARTS OF OUR IMPLEMENTATION, PRIMARILY FOR FUNCTIONS THAT WERE FOCUSED ON USING MBOT MOVEMENTS
- IT WAS ALSO USED TO INITIALIZE THE BASE MOVEMENTS OF THE MBOT'S ROBOT WHEELS
- ARDUINO C (C & C++ FUNCTIONALITIES) IS THE PROGRAMMING PLATFORM WE USED FOR OUR IMPLEMENTATION
- PRIMARILY IT WAS EASY TO LEARN, AND IS VERY COMPATIBLE WITH ACADEMICS, SINCE A MAJOR PART OF OUR DEGREE COMPOSES OF LEARNING C++

0		1		2		3		4		5		6		7		8		9

10		11		12		13		14		15		16		17		18		19

20		21		22		23		24		25		26		27		28		29

30		31		32		33		34		35		36		37		38		39

40		41		42		43		44		45		46		47		48		49

50		51		52		53		54		55		56		57		58		59

60		61		62		63		64		65		66		67		68		69

70		71		72		73		74		75		76		77		78		79

80		81		82		83		84		85		86		87		88		89

90		91		92		93		94		95		96		97		98		99

3		0		0		0		0		0		0		0		0		0

0		0		0		0		0		0		0		0		0		0

0		0		0		0		0		0		2		0		0		0

0		0		2		0		0		0		0		0		0		0

0		0		0		0		0		1		0		0		0		0

0		0		0		0		0		0		0		0		0		0

0		0		0		0		2		0		0		0		0		0

0		2		0		0		0		0		0		2		0		0

0		0		0		0		0		0		0		0		0		0

0		0		0		0		0		0		0		0		0		0

GRID MAP & POSSIBLE MOVES

A* ALGORITHM & MANHATTAN DISTANCE

PATH FINDING

- MANHATTAN DISTANCE IS BEST USED WHEN THE ROBOT ONLY CONSIDERS 4 POSSIBLE MOVES RATHER THAN HEXAGONAL OR OCTAGONAL
- MANHATTAN DISTANCE FINDS THE AMOUNT OF TILES NEEDED HORIZONTALLY AND VERTICALLY TO GET THE MBOT FROM THE INITIAL TILE TO THE GOAL TILE
- THE G VALUE WAS CALCULATED BASED ON THE “DEPTH” OF THE “TREE” IF WE WERE TO THINK ABOUT IT LIKE A TREE. SO, ESSENTIALLY THE INITIAL TILE HAD A ZERO G VALUE AND ITS CHILDREN HAD ONE AS A G VALUE, AND EVERYTIME THE TILE HAD AVAILABLE CHILDREN, WE INCREMENTED THE G VALUE BY 1
- THE F VALUE WAS CALCULATED BY ADDING THE TILE’S INDIVIDUAL G AND H VALUES WHICH RESULTS IN OUR F VALUE, USED IN THE HEURISTICS FUNCTION

FUNCTIONS

- VOID MOVEMENT()
- VOID FORWARD()
- VOID LEFTTURN()
- VOID RIGHTTURN()
- VOID BACKWARDS()
- VOID BUILDMAP()
- VOID PRINTGRID1()
- VOID PRINTGRID2()
- VOID LOOP ()
- VOID _DELAY(FLOAT SECONDS)
- VOID DELAY(FLOAT MILLISECONDS)

0		1		2		3		4		5		6		7		8		9

10		11		12		13		14		15		16		17		18		19

20		21		22		23		24		25		26		27		28		29

30		31		32		33		34		35		36		37		38		39

40		41		42		43		44		45		46		47		48		49

50		51		52		53		54		55		56		57		58		59

60		61		62		63		64		65		66		67		68		69

70		71		72		73		74		75		76		77		78		79

80		81		82		83		84		85		86		87		88		89

90		91		92		93		94		95		96		97		98		99

FUNCTIONS

- VOID _LOOP()
- VOID SETGOAL()
- VOID POSSMOV(BYTE GRIDNOM)
- VOID ADDOPENLIST(BYTE AOL)
- VOID HEURISTICS(BYTE CURIN)
- BYTE GETNEXTFI()
- VOID ADDCLOSEDLIST()
- VOID REMOVEFOL(BYTE RFOL)
- BOOL OLE()
- BOOL ISGOAL(BYTE IG)
- BOOL ALREADYONOL(BYTE ROWAOL, BYTE COLAOL)

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

MBOT MOVEMENT METHOD & REASONING

- WE DECIDED TO USE TIME AS MOVEMENT RATHER THAN DISTANCE FOR THE FOLLOWING REASONS
- - WE COULD NOT FIND A ROOM WITH THE SAME DIMENSIONS AS OUR GRID AND THE ROOMS THAT CAME CLOSE HAD OBJECTS WITHIN THEM
- - IF WE USED THE LINE FOLLOWING SENSOR, THE MBOT WOULD CONTINUOUSLY FOLLOW THE BLACK LINE WITHOUT STOPPING, UNLESS WE WORKED WITH THE ULTRASONIC SENSOR AS WELL AS A VARIETY OF EXTRA MEASURES
- - WE ALSO COULD HAVE USED THE ULTRASONIC SENSORS BUT SINCE THE ROOM OR LOCATION WE HAD WAS NOT THE SAME DIMENSIONS WE NEEDED, THOSE CALCULATIONS BASED OFF DISTANCE WOULD BE INCORRECT
- ANYTIME THE MBOT NEEDED TO GO FORWARD, IT WOULD APPROXIMATELY MOVE FORWARD FOR A SECOND AND A HALF
- ANYTIME THE MBOT NEEDED TO TURN LEFT OR RIGHT, WE SET THE SPEED OF THE MOVEMENT TO BE FIXED AND TURNED THE MBOT APPROXIMATELY FOR A SECOND FOR BOTH TURNS
- WE IMPLEMENTED THE MBOT TO TAKE TWO RIGHT TURNS TO TURN THE MBOT 180 DEGREES TO FACE THE OTHER WAY
- AFTER EVERY MOVE IT TAKES, IT REPOSITIONS ITSELF TO FACE THE "FRONT"

EXPERIMENTS & RESULTS

- [HTTPS://YOUTU.BE/ADDFbS26GVG](https://youtu.be/ADDFbS26GVG)
- [HTTPS://YOUTU.BE/XJY-V-PrWp8](https://youtu.be/XJY-V-PrWp8)

ISSUES & DIFFICULTIES

- THERE WERE OCCASIONS WHERE THE LOWEST F VALUE PICKED WAS A TILE WAY EARLIER IN THE PATH, INSTEAD OF THE TILES NEAR THE CURRENT POSITION OF THE MBOT
- THE ARDUINO'S DYNAMIC MEMORY IS QUITE SMALL IN RELATIONS TO OTHER PLATFORMS
- THE PROGRAM TOOK UP APPROXIMATELY 91% OF THE DYNAMIC MEMORY, AND IF THE DYNAMIC MEMORY IS ALMOST FULL, THIS PROBLEM MAY LEAD TO MISCALCULATIONS, ERRORS AND/OR BUGS, ACCORDING TO ARDUINO
- THE LOCAL MEMORY HOWEVER WAS COMPLETELY FINE AND WE USED APPROXIMATELY 22% OF THE LOCAL MEMORY
- ANY DUST/GRAINS/ETC ON THE SURFACE WE RUN THE MBOT'S PROGRAM ON, THESE DEBRIS WOULD AFFECT THE MBOT'S MOVEMENTS AND SLOW IT DOWN
- ANOTHER ISSUE WE RAN INTO IS THE FACT THAT THE MBOT'S LEFT WHEEL IS WEAKER THAN ITS RIGHT WHEEL
- LASTLY, WHEN AVOIDING OBSTACLES TO REACH ITS GOAL STATE, THE MBOT WOULD MAKE APPROPRIATE MOVES BY AVOIDING THE OBSTACLE BUT WOULD BE 1 TILE OUT OF PLACE (IN ANY DIRECTION) WHEN SEARCHING FOR THE GOAL TILE
- SINCE MOST OF THE DYNAMIC MEMORY WAS OCCUPIED, IT COULD LEAD TO ERRORS AND SUCH STATED BY THE ARDUINO IDE, THERE WERE OCCASIONS WHERE WE RAN THE PROGRAM AND THE MBOT WOULD TAKE DIFFERENT MOVES THAT WAS NOT IMPLEMENTED ACCORDING TO OUR ALGORITHM

DESIRED IMPROVEMENTS & FUTURE WORK

- WE BELIEVE THAT WE COULD FIX THE ISSUES IN OUR ALGORITHM CONCERNING THE OPEN LIST AND CLOSED LIST
- WE COULD ALSO FINE TUNE THE MOVEMENTS OF THE MBOT SO THAT IT RUNS MORE SMOOTHLY, BUT AS IT IS NOW, WE BELIEVE THAT IT IS DEFINITELY GOOD ENOUGH
- ANOTHER POTENTIAL IMPROVEMENT WE COULD MAKE, IS TO USE USING ALL THE SENSORS OF THE MBOT
- IN THE FUTURE, SOME OF US WOULD LIKE TO WORK WITH MORE ROBOT MOTION PLANNING, IN ORDER TO IMPACT THE INDUSTRY AND MAKE A CHANGE TO THE AI AND COMPUTER SCIENCE WORLD
- WE COULD FINE TUNE OUR PROGRAM AND ALGORITHM TO MAKE IT OPTIMAL, AND FROM THERE EXPERIMENT WITH MORE PATH FINDING AND SEARCH ALGORITHMS, ALONG WITH INCORPORATING ROBOTICS
- IN PARTICULAR, WE WANT TO IMPLEMENT THE LINE SENSOR INTO THE PROGRAM AND USE A DRAWN 10X10 GRID TO MAKE SURE THAT THE ROBOT DOES NOT DEVIATE FROM THE GIVEN PATH
- ANOTHER IDEA IS TO IMPLEMENT THE ROBOT'S ULTRASONIC SENSOR INTO PROGRAM SO THAT THE ROBOT CAN MORE ACCURATELY CALCULATE THE DISTANCE FROM THE GOAL.

CONCLUSION

- THIS PROJECT WAS A GREAT LEARNING EXPERIENCE FOR ALL OF US
- GAVE US A LITTLE MORE INSIGHT ON THE A^* ALGORITHM AND HOW IT IS APPLIED
- GAVE US A LOT MORE INSIGHT IN ROBOTICS, THOUGH THE mBOT IS JUST THE SURFACE WHEN IT COMES TO WORKING WITH AND IMPLEMENTING ROBOTICS
- GAVE US INSIGHT ON THE LIMITATIONS OF OUR ROBOT, AND HELPED US TO CONTEMPLATE OTHER WAYS TO FURTHER IMPROVE THE A^* ALGORITHM
- WITH THE GROWTH OF ROBOTICS AND PROGRAMMING, IT WAS IMPORTANT LEARNING ABOUT THIS AREA IN COMPUTER SCIENCE AND THE EXPERIENCE HELPED WITH GIVING US SOMETHING DIFFERENT FROM WORKING ON THIS PROJECT

QUESTIONS AND/OR COMMENTS

REFERENCES