School of Engineering and Information Sciences

MIDDLESEX UNIVERSITY

EXAMINATION PAPER

2009/2010 Winter/Spring term

MODULE TITLE Advanced Topics in Games Development

MODULE NUMBER CMT 3325

MODULE LEADER'S NAME Chris Huyck

Time allowed: 3 hours

Total number of questions: 4 questions

Instructions to candidates: Answer all questions. Each question carries 25 marks.

Materials provided: The book Programming Game AI by Example (by Buckland) is allowed, along with one A4 page marked as notes on the top. Please submit that page with the exam (it will not be marked).

Equipment permitted: none

Total number of pages: 3

No books, papers or electronic devise is to be brought into the examination room other than any specified above.

Candidates are warned that illegible scripts will not be marked

1. AI

We discussed the cannibals and missionaries problem in class. There are three missionaries and three cannibals on one side of the river and they want to get to the other. They have a boat, but the boat only carries two people, and at least one person needs to be in the boat to cross the river. If the cannibals ever outnumber the missionaries on either side of the river, they missionaries get eaten.

(a) Draw the initial state, the goal state and all possible move pairs from the initial state. Mark the states that are illegal (where the missionaries are eaten).

(6 marks)

Marking scheme:
2 points for initial and final state
2 points for first moves
2 points for second moves and legal marking

Sample answer:
The initial state is mmmcccb/ (m is a missionary, c a cannibal, b the boat and / the river).
The final state is /mmmcccb
First moves are 1. mccc/mmb I (I denotes illegal)
2. mmccc/mb I
3. mmcc/mcb
4. mmmc/ccb
5. mmmcc/cb
Second moves are (ignore 1 and 2 as illegal)
From 3 mmcccb/m I
mmmccb/c
From 4 mmmccb/c
From 5 mmmcccb/

(b) Describe the state space and its size.

(5 marks)

Marking Scheme:
2 marks for a reasonable answer.
3 marks for a reasonable description.

Sample Answer:
You can exhaustively enumerate the reachable states. There are at least 10 from the above example. If you consider that each of the 7 pieces can be in 2 places you get a 7 digit binary number which is 32 possible states. As there are redundant states with the binary calculation, there are probably around 24 states.

(c) Describe a search algorithm (preferably in pseudo-code) that can solve the problem.

(8 marks)

Marking Scheme:
2 marks reasonable answer

2 marks some algorithm (or sound description)
2 marks a working algorithm
2 marks a good algorithm (not something that can get stuck in a loop).
Sample Answer:
A reasonable answer is breadth first search.
Make the initial state the current state
Make the queue of states to check and the list of visited states empty
Until final state found
Expand all legal states from the current state
For each of those
If they are not already visited, and are legal put them in the queue to check
Add the current state to list
Remove the front of the queue and make it the current state

(d) Describe the numbers of states in a draughts (checkers) game.

(5 marks)

Marking Scheme:
2 marks reasonable answer
3 marks description.
Sample Answer:
There are 32 usable positions on a checker board. As a first pass, each can have one of five states: empty, black, black king, red, or red king. So, there are 32^5 possible states. That of course can be reduced as you're limited to 12 black and 12 red checkers; at least 8 must be empty.

2. Physics
(a) A football is sitting still in the grass. Will it be there in an hour? Paraphrase (or quote) Newton's law that gives you the answer.

(8 marks)

Marking scheme
2 marks for it still being there
6 marks for the paraphrase
Sample answer:
Unless something comes along and moves it, it should still be there. This is due to Newton's law that states things stay at rest (or continue moving as they are) unless some force is applied to them.

(b) If that football weighs 200 grams and is dropped from 15 meters, how fast is it going when it hits the ground? (Please show work.)

(9 marks)

Marking Scheme
3 marks for acceleration
2 marks for velocity
2 marks for time slicing or an analytical solution
2 marks for a roughly correction solution
Sample answer:
The gravitational constant for earth is 9.81 m/s^2. Time slicing it by .2 and rounding to 10m/s^2

Time	Velocity	Distance
.2	**2m/s**	**.4 m**
.4	**4 m/s**	**.4 + .8m = 1.2**
.6	**6 m/s**	**1.2 +1.2m = 2.4**
.8	**8 m/s**	**2.4m + 1.6m = 4m**
1.0	**10 m/s**	**4m + 2m = 6m**
1.2	**12 m/s**	**6m + 2.4m = 8.4m**
1.4	**14 m/s**	**8.4 + 2.8m = 9.2m**
1.6	**16 m/s**	**9.2 + 3.2m = 12.4 m**
1.8	**18 m/s**	**12.4 + 3.6m = 16m**

So, it is travelling at around 17m/s when it hits the ground.

(c) Part b assumed the movement of the ball could be described quite simply. What other forces might be considered for football kicking in the development of a football game.

(8 marks)

Marking scheme
2 marks translational motion.
2 marks rotational motion.
2 marks friction.
1 mark for gravity
1 mark for the leg or other reasonable points

Sample answer:
Gravity is an important consideration for ball movement, but kicking is also important. Kicking involves a foot hitting the ball. There are a range of mechanics behind the leg motion, but those would probably be ignored in a game. Kicking will involve translational motion (the ball moves) but also involve rotational motion. Rotational motion allows the ball to spin and drift through space. A third type of force that should be considered is friction. The ball should travel differently through the air than on the ground.

3. Software and AI

(a) For the first coursework, Huyck provided an API for you to use while developing an agent for Space Wars. Describe that API and devise another. Compare the two APIs

(10 marks)

Marking Scheme:
2 marks getShip1 (from provided)
1 mark cover functions (from provided)
2 marks largely correct provided solution (they can get the first three marks by providing any two APIs)
2 marks second different API
3 marks reasonable comparison

Sample Answer:
The API allowed you to get a ship variable to refer to the agent's opponent's ship. This was getShip1 or getShip2 (you needed two because you weren't sure which you'd be). These came from the program. Once you got the ship you could query it's location, velocity, acceleration or its bullets with cover functions (like ship.getVelocity()).
Another option would have been to merely ask the program for those values.
The first option makes it explicit whose values you are retrieving, but it does require more support (the extra getShip functions), and it requires the program to support them. The second option requires the program to support more things, does not allow multiple agents to work easily, and doesn't isolate code in the ship class. All in all, the first is better as it supports more data hiding, but they're not very different.

(b) Space wars involved agents, asteroids, ships, bullets and a sun. Describe a good class structure for this game, and why it is good (you can use the one from the game or your own structure).

(8 marks)

Marking Scheme:
2 marks class structure
2 marks inheritance
1 mark sound structure
3 marks for why it is good

Sample Answer:
Object
MoveableObject isa object
StationaryObject isa object
Sun isa stationaryObject
Ship isa moveableObject
Bullet isa moveableObject
Asteroid isa moveableObject
Agent isa Ship
User isa Ship
Screen (which does collision detection)
Keyboard
Sound

It's a good structure because it reuses properties and methods. The objects in the game have similarities, and several layers of inheritance allow specialization and reuse. The screen binds the whole game together, containing the objects. Some helper classes provide interaction. The agent and user subclasses of ship enable specialization for software agents not provided in the original game.

(c) How does this structure compare to a semantic net?

(7 marks)

Marking Scheme:
1 mark for classes are nodes.
2 marks for inheritance is the same.
2 marks for instanceOf
1 mark for variables are arcs
1 mark for values are also nodes
(Other answers may get points, but this is a robust answer.)
Sample Answer:
Classes and data values (not described above) are nodes in the semantic nets. The variables are specialized arcs. Inheritance is the isa arc, and instantiation is the instanceOf arc. Class subcategorisation is a hallmark of semantic nets, and has been reused in Object oriented analysis and design.

4. Theory, software and AI
(a) Draw a finite state automata for a chatbot (agent) in a role playing game.

(8 marks)

Marking Scheme:
2 marks state
2 marks transitions
2 marks text output and input
2 marks reasonable conversation
Sample Answer:
S0 start SE end. I'll use SX "text" to say State X produces the text text.

S0 "Hello, what can I get you?"
--- hello → S1
--- I'll have a (coke, juice, milk) -→S2
-→I'll have an (other drink) --→ S3
-→?Y → S4
S1 "what can I get you?"
-→ coke, juick, milk-> S2
→other drink-→ S3
→?X ->S4
S2 "here's your (coke, juice, milk)"
->empty ->SE
S3 "sorry, we don't have that"
->empty->S1
S4 "sorry, I didn't catch that"

->empty->S1
SE “bye now”

(b) One problem with chatbots and other simple agents is that they do not change over time. Aside from machine learning (part c) what could be done to make the agent more flexible so that it does not always say the same sort of things?

(8 marks)

Marking Scheme:
2 randomness
2 larger state space
2 memory
2 variables
Sample Answer:
The simplest chatbot is deterministic. The next step is to allow input from the user to change this process, as done in part a. After that there is a range of ways to change things. One simple mechanism is just to make a larger FSA; this makes it less likely that a user will visit the same states in the same order. Another simple way to change things is to add some randomness; this can make it even less likely that things will be repeated, but it can add some oddities. Another thing can be to add some wildcards as in part a; the user can put in a,b or c, a class item, or anything. That particular item can then be used in the next step. Finally, a richer set of memory can be used. This can be finite, or an infinite stack. This can keep a context. Beyond that, the scope is almost limitless with full fledged parsing, and discourse management capable of being implemented; this of course can take a lot of development.

(c) Consider adding machine learning to the agent. Describe an appropriate simple machine learning mechanism and a more complex one. Describe the tradeoffs between these mechanisms.

(9 marks)

Marking Scheme:
3 marks simple.
3 marks reasonable complex
3 marks excellent complex
Sample Answer: (Both sample answers are merely samples. Students will typically produce radically different answers.)
A simple machine learning algorithm is to merely cache away a user’s request. In part a, the user asks for a drink. Cache that away, and the next time the user comes in you could say, can I get you a milk (or whatever they asked for last time).
A more complex learning algorithm depends on the task the agent is carrying out. In the bar tending example above, the agent might want to maximise the number of drinks he sells. He could modify his banter to sell more drinks, develop new drinks, or change the price. This kind of exploration could be enabled by a rule based system; machine learning could be used to weight the rules. Feedback from the environment (success or not) could change the weighting of rules. Of course there

might be problems with the exploration exploitation dilemma, but that would make a more interesting agent.