School of Engineering and Information Sciences

MIDDLESEX UNIVERSITY

EXAMINATION PAPER

2009/2010 Summer 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 are permitted to be brought into the examination room other than any specified above.

Candidates are warned that illegible scripts will not be marked

1. AI

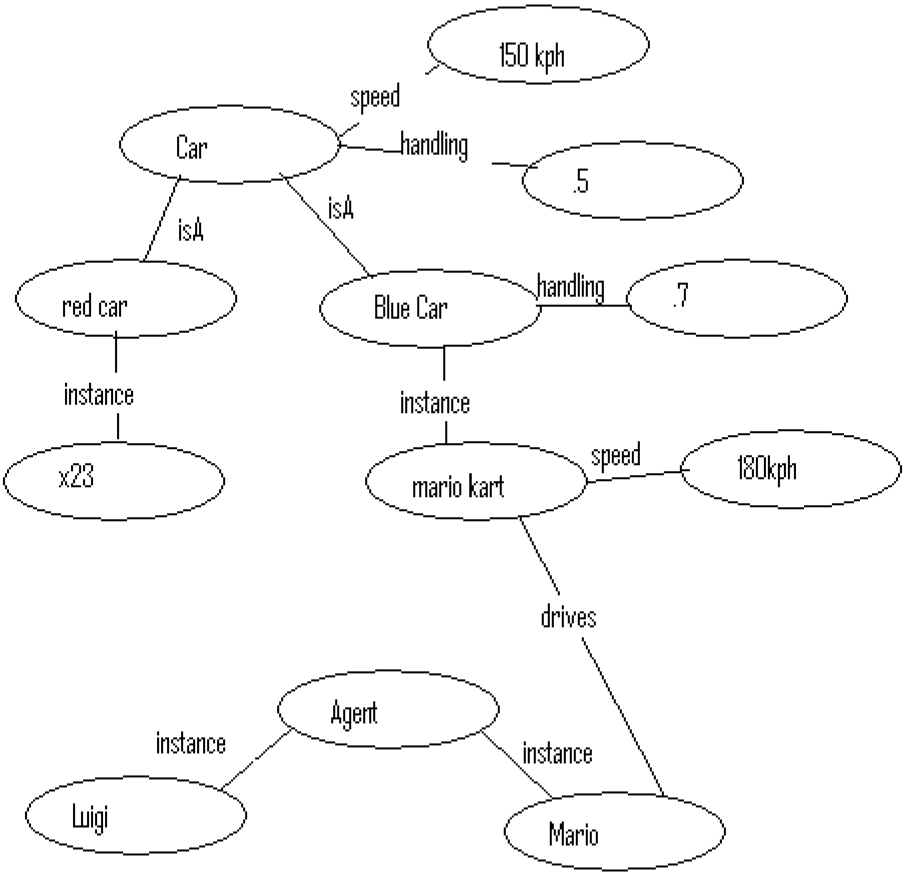
(a) Draw a semantic net that might be used for an automobile racing game. Provide at least 12 nodes and the 3 most important types of relation.

(11 marks)

Marking scheme:
3 points for 12 nodes
2 points for isa
2 points for instance or part-of
2 points for other type of arc
2 points for a good solution.

Sample answer:

The user bit might be a bit dodgy.



(b) In the automobile racing game provided as an XNA starter kit, the system provides you with the ability to race against an earlier time you raced. Your opponent is a ghost and you can not run into it. What simple machine learning mechanism is used for this?

(6 marks)

Marking scheme:
3 points for a reasonable answer
3 points for caching

Sample answer:

The simplest form of machine learning is to just cache away behaviour. So, in this case it looks like the starter kit merely records the user's behaviour. This can require a lot of space, but works. However, it requires things to be the same when played back. In this case, the ghost behaviour means that it behaves the same on the replay; it wouldn't work if the user could bump the ghost.

(c) Many AI techniques have been developed to play board games such as chess. Describe an AI technique that is widely used for chess; provide details and example boards.

(8 marks)

Marking scheme:
2 points for evaluation.
2 points for look ahead
2 points for a good answer. (Not necessarily minimax.)
2 points for examples.

Sample answer:

The standard technique uses an evaluation function for the quality of the board. The function involves number of pieces (for and against), control of portions of the board, and threat on pieces (such as the king). From the current position, moves are calculated ahead with a movement tree. Some form of minimax would be used to note that the opponent wants to minimize while the system wants to maximize. In very few boards can you see the end of the game, so this tree is incomplete.

An elided example has black (the player) in caps and white in lowercase

```
0                          20                 -infinity
………………..               ………………..           ……………..
……KQ……… --→              ……K……… ------→      ……  r………
……rq……….                ……rQ……….            ……Q……….
………………..               ………………..           ………………..
|
|
V
0                          -10
…….K……….                ……rK……….
…….Q……… --→              …….Q………
……rq………..               …….q……….
………………..               ………………..
```

Values are on the boards with black maximizing and white minimizing. Lookahead shows the top moves is short sighted.

2. Physics

(a) If two people are sitting (with their feet off the ground) in chairs that roll, and one pushes the other, what will happen? Paraphrase (or quote) Newton's law that gives you the answer.

(8 marks)

Marking scheme:
3 points for they'll move away from each other.
2 points for a reasonable paraphrase
3 points for a solid paraphrase (or exact quote).

Sample answer:

Both chairs will move away from each other and the initial point. This is based on Newton's third law, "To every action there is always an equal and opposed reaction."

(b) The people and the chairs are in interstellar space. After the push, using well positioned thrusters, the two people in the chairs come to rest 10 meters apart. One person and chair weighs 100 kilograms, and the other 200 kilograms. How much gravitational force does the larger exert on the smaller? (Please show work.)

(9 marks)

Marking scheme:
4 points equation
2 points G
3 points correct

Sample answer:

$F=Gm_1m_2/r^2$

G = 6.656e-11

F = 6.656e-11*100,000*200,000/10*10 =
6.656e-11*100,000*2000 =
6.656e-11*2e+8 = 13.312e-3 = .013312

(c) Part b assumed the movement of the chairs could be described quite simply. If the chairs are sitting in a room on earth, what other forces might be considered for a video game involving chair pushing? What might be involved in an interesting game involving chairs?

(8 marks)

Marking scheme:
2 points gravity between the objects is largely irrelevant.
2 earth's gravity
2 friction
2 other points.

Sample answer:

The two obvious forces are the gravity of the earth and friction of the wheels against the surface. The gravity between the two objects is more or less irrelevant and can be ignored in a game. Wind is probably more of a factor. I think a game that involved pushing chairs around would require some force to actually move the chairs (like a fire extinguisher or a person pushing the chair). An interesting game might involve wheel-chairs.

3. Theory and Software

(a) Provide an API (the public portion of the class structure) for a Finite State Automata (FSA).

(9 marks)

Marking scheme:
2 points an API
2 points constructing an FSA
2 points using an FSA
2 points a complete API

Sample answer:
AddState(S1)
ConnectStates(S1,S2,transit)
Run()
ApplyInput(t);
GetState();

(b) What are the pros and cons of providing a small API and a large API?

(8 marks)

Marking scheme:
2 points smaller is better for data hiding
2 points smaller is better for maintanance
2 points for larger is better for the user (at least initially).
2 points other reasonable points.

Sample answer:

One of the benefits of a small API is that the underlying code can be modified without having to support a large API. As an API grows larger, more of the underlying behaviour can be exposed. On the other hand, a larger API makes it easier for a user of the API to do what they want. It's also easier to document a smaller API, and easier to understand it.

(c) Here is some text from http://www.cplusplus.com/doc/tutorial/templates/ explaining C++ templates: Function templates are special functions that can operate with *generic types*. This allows us to create a function template whose functionality can be adapted to more than one type or class without repeating the entire code for each type. What would be the pros and cons of providing a C++ template vs. specific FSA classes?

(8 marks)

Marking scheme:
3 points reduces duplication
3 is harder to understand
2 other reasonable points.

Sample answer:

One of the benefits of a template is that it reduces the need to re-implement. Whenever an FSA using any underlying type of state and transitions is needed, the template can be used. However, templates can be hard to understand; they require a level of abstract thinking that can be difficult to follow; and debugging proceeds through an extra layer (that may not be present in the source code). In the case of really core data

structures (e.g. lists) a template may be useful. In the case of rarely used structures, like FSAs, it might be best to duplicate code.

4. Search

The Towers of Hanoi is a classic AI problem. There are 3 towers and several differently sized discs. Discs can only be placed on the ground, or on larger discs. The problem is to move the discs (all initially on the first peg) to the second peg.

(a) Draw the initial state, the goal state and three moves of discs.

(6 marks)

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

Sample answer:
Initial state ABCD 1 2 3 (A-D are discs with A the smallest; 1 2 3 are pegs. Discs are to the left of the pegs they're on.)
Goal state 1 ABCD 2 3
BCD1 A 2 3
CD 1 A2 B3
CD 1 2 AB3

(b) Assuming there are 4 discs, describe the state space and its size. In this case it can be done quite precisely.

(7 marks)

Marking scheme:
2 points for reasonable answer.
2 points for exact answer.
3 points for description.

Sample answer:
Any state can be reproduced on 3 different peg configurations, so figure out the number for 1 and then multiply by 3. For example, the Initial and goal states above are 2 of the 3 possible 5,0,0 states.
The base states in order toward solution, with spurious paths to the right are:
ABCD123
BCD1A2
CD 1 A2 B3 -> ACD12B3
CD 1 2 AB3
D1 C2 AB3 -> D1AC2B3
AD1 C2 B3
AD1BC23 -> D1BC2A3
D1ABC23
1ABC2D3 (This is isomorphic with the prior one, and no new ones are possible.)
That's 11x3 or 33 states.

(c) Write (pseudo) code to solve this problem.

(7 marks)

Marking scheme:
2 points for pseudo code
2 points for a close solution
3 point for an exact one.

Sample answer:

Simple pseudo code just follows the arc of states above and back with a big cascaded if.

(d) Write (pseudo) code to solve the problem for any number of discs.

(5 marks)

Marking scheme:
2 points for a reasonable shot
3 point for a correct solution. (Iterative solutions do exist.)

Note a correct answer here gives full points for part c.

Sample answer:

```
Move (ABCD..N from1 to3 other2)  //move the list from 1 to 3
Function move (movelist from to other) {
Head = movelist – last// ABCD..N -> ABCD..N-1
If (length (head) > 1) {
  Move(movelist, from, other, to)
  PrimitiveMove (last, from, to)
  Move(movelist,other, to,from)
}
Else (primitiveMove(last,from,to))
}
//It’s a recursive solution.  Tower of Hanoi is often used to teach recursion.
```