## Instructions

This activity is designed to allow you to demonstrate your understanding of the Intended Learning Outcomes (ILOs) for this unit. Questions are of an open form, and there may not be a single correct answer. It is up to you to demonstrate and support what you know in your responses.

- Select the best **one** or **two** questions that you feel will allow you to demonstrate your knowledge.
- Do not answer all questions. (There's no room anyway!)
- A maximum of the first **three** answers will be marked. Extra answers will be ignored.
- You can cross out answers you do not want marked.

Clearly write your name and student ID on every answer page.
Clearly label the question you are answering.
Clearly label each figure and include captions. Figures should be on the blank side of answer pages.
Answer in any order you wish.
You must submit both this document and all your answer pages at the end of the test.

You have 1 hour to answer the questions.

## Assessment

Date : 23/04/2018 Name: PETER ARGENT ID: 7649991

Markers will provide feedback in the matrix below regarding the depth of understanding you have demonstrated in each of your answers. (Simple numeric marks are not used.)

| ILO's | Q 6 | Q 5 | Q 2 |
|---|---|---|---|
| 1. Discuss & implement software development techniques to support the creation of AI behaviour in games | | Shallow | Shallow |
| 2. Understand and utilise a variety of graph and path planning techniques | | | |
| 3. Create realistic movement for agents using steering force models | Shallow | | |
| 4. Create agents that are capable of planning actions in order to achieve goals | | Shallow | Shallow |
| 5. Combine AI techniques to create more advanced game AI. | | | |

**Overall Comments**
You have some good ideas and knowledge, but it might have been worth answering fewer questions so you could go into more depth.

**Assessment Key**

| | |
|---|---|
| N/R or W | *Not Relevant (does not answer the question) or Wrong (incorrect) details* |
| Shallow | *Simple (relevant) details but not very deep (terms, concepts, process)* |
| Good | *Medium level of details (good descriptions, lists, combined ideas)* |
| Deep | *Strong relational knowledge (compare, analyse, contrast, relate)* |
| Very Deep | *Reflection, extended knowledge, generalisation (extrapolation), theorisation (ie. "Wow!")* |

## ILOs

For reference, all the ILOs for the unit are repeated below:

1. Discuss & implement software development techniques to support the creation of AI behaviour in games
2. Understand and utilise a variety of graph and path planning techniques
3. Create realistic movement for agents using steering force models
4. Create agents that are capable of planning actions in order to achieve goals
5. Combine AI techniques to create more advanced game AI.

## Open Answer Questions

Select and answer from the following set of open answer questions.

1. Finite state machine (FSM) models are one of the oldest and perhaps most used AI techniques for games. **Create** an example FSM design, using **figures** to support your answer, and **discuss** the advantages and limitations of using FSMs for game AI.

2. Games can be *balanced* or *unbalance*, actions can be balanced or unbalanced, and players can be *biased*. Using **specific explanations** to support your answer, **discuss,** especially in relation to player experience and AI bot design. Use **figures** to support your answer.

3. *Goal-oriented behaviour* (GOB) includes a wide-range of techniques that agents can use to select actions and achieve goals. *Simple goal insistence* (SGI) techniques for GOB action selection are limited and prone to making "unintelligent" decisions. **Describe** and **explain**, using a specific example, the limitation of SGI for goal-oriented behaviour (GOB).

4. **Explain** and **discuss** the problems of "side effects" and "time-delay" for an agent using goal-oriented behaviour (GOB). Include a **description** of how a model of "discontentment" can be used to address these problems.

5. The terms "*strategy*" and "*tactic*" can be defined as a means of clarifying concepts in relation to games and AI. **Explain** and **Discuss**. Use examples to support your answer.

6. **Describe** in general vector terms, and using **figures** to support your answers, the low-level steering behaviours of *seek*, *flee*, *arrive*, *pursuit* and *evade*.
(Tip: Don't go into compound behaviours such as path following or hiding.)

7. Low-level steering behaviour can be combined with tactical information to create higher-level behaviours such as *interposition*, *offset pursuit*, *path-following* and *hiding*. **Discuss** and **explain**. Use **figures** to support your answer.

8. **Describe** in detail the three core group steering behaviours used to create "flocking" behaviour, including an **explanation** of how the adjustment or weighting of each influences the overall behaviour. Use **figures** to support your answer.

*Tips: Remember that this activity is an open answer style opportunity for you to demonstrate your knowledge of the intended learning outcomes, and so referring to them should help you in your answers and the points you select. Refer also to the instructions. A quick plan before for each answer is a good idea.*

Name: PETER ARGENT ID: 7649991 Page 1

Question 6: Describe in Vector Terms low-level steering behaviors.

Seek: Upon Each Tick, calculates a vector of force or acceleration towards a specified target (see figure 6.1) ✓

Flee: Upon Each tick where the agent is within range of the target, apply a force/acceleration vector to the agent in the opposite direction of the target (see figure 6.2) ✓

Pursuit: like seek, except the Target has it's own velocity so we should be running ~~from~~ towards ~~where~~ the ~~target~~'s future location (see figure 6.3) - how do we get the future location?

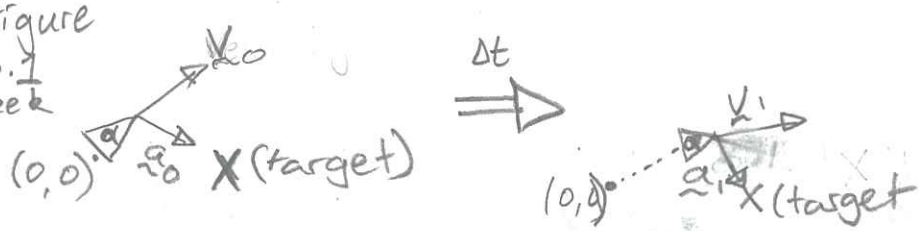
Evade: similar to Flee, Evade has an agent accelerate away from a particular location, in this case that location is away from where the persuer is likely to be. (see figure 6.4

Arrive: Uses seek behavior to calculate the direction of the vector of acceleration but also limits the magnitude of that vector as it approches the target position. - How?

Seek can be used as the base for multiple movement behaviours.

Use this side for Figures. Clearly Label and Caption each Figure. Refer to Figures in your answers.

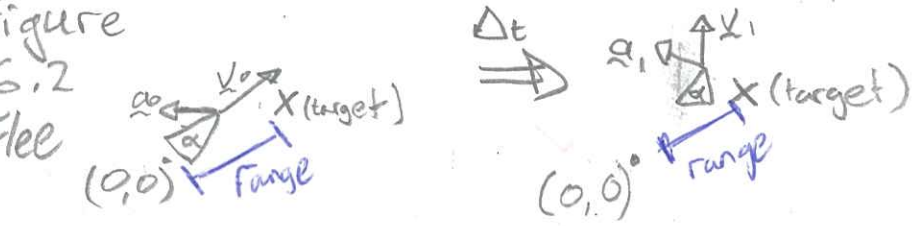
Figure 6.1 Seek




$$\underset{\sim}{V}_1 = \underset{\sim}{V}_o + \underset{\sim}{a}_o$$

$$\underset{\sim}{V}_n = \underset{\sim}{V}_{n-1} + \underset{\sim}{a}_{n-1}$$

Figure 6.2 Flee




$$\underset{\sim}{V}_1 = \underset{\sim}{V}_o - \underset{\sim}{a}_o$$

$$\underset{\sim}{V}_n = \underset{\sim}{V}_{n-1} - \underset{\sim}{a}_{n-1}$$

when within range

Figure 6.3 Pursuit

$\alpha$ is chasing $b$


Figure 6.4 Evade

$\alpha$ is chasing $b$

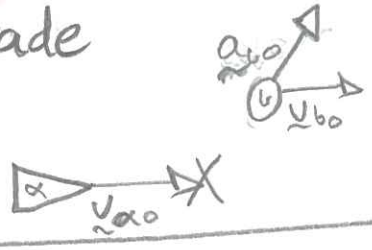
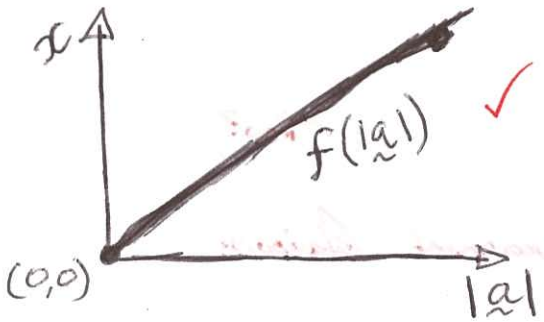



Figure 6.5 Arrive




x is distance to target

$|\underset{\sim}{a}|$ is the magnitude of acceleration.

$f(|\underset{\sim}{a}|)$ is the graph of how ~~xxxx~~ the magnitude of acceleration relates to ~~&~~ x

Question 5: Explain & Discuss "Strategy" vs "Tactics" in Game AI

~~In Game AI strategy can be thought of as a goal to achieve a goal whereas tactics could be the means that plan is implemented.~~

~~Using an RPG as an example Your goal is to get past an NPC Strategy w~~

In Game AI strategy is the goal an agent is trying to Achieve and tactics are the means by which to achieve that goal.

For example in an RTS you may have different strategys that result in a win;
- Elimination, - Wonder - Influence. — these might still be too specific to be strategies.

The tactics for each of these strategys could be

(Wonder) • Build lots of workers to speed construction and resource gathering, then when resources are depleted all workers build wonder.

(Elimination) • Build the cheapest battle units as fast as possible then send them all at the enemy while building more of the same unit to send at the enemy again whilst they are in disarray.

(Elimination) • Research the Tech tree as fast as possible the crush the enemy forces with the most advanced units.

Use this side for Figures. Clearly Label and Caption each Figure. Refer to Figures in your answers.

Figure 1.1 FSM Simple

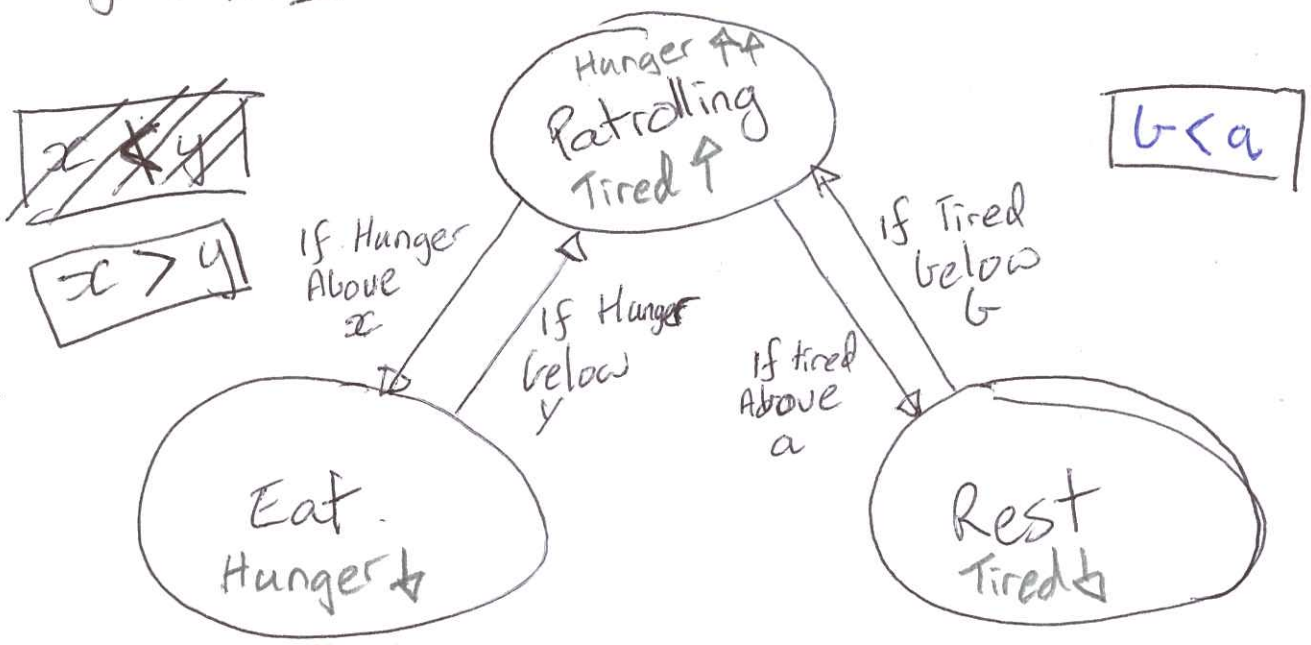
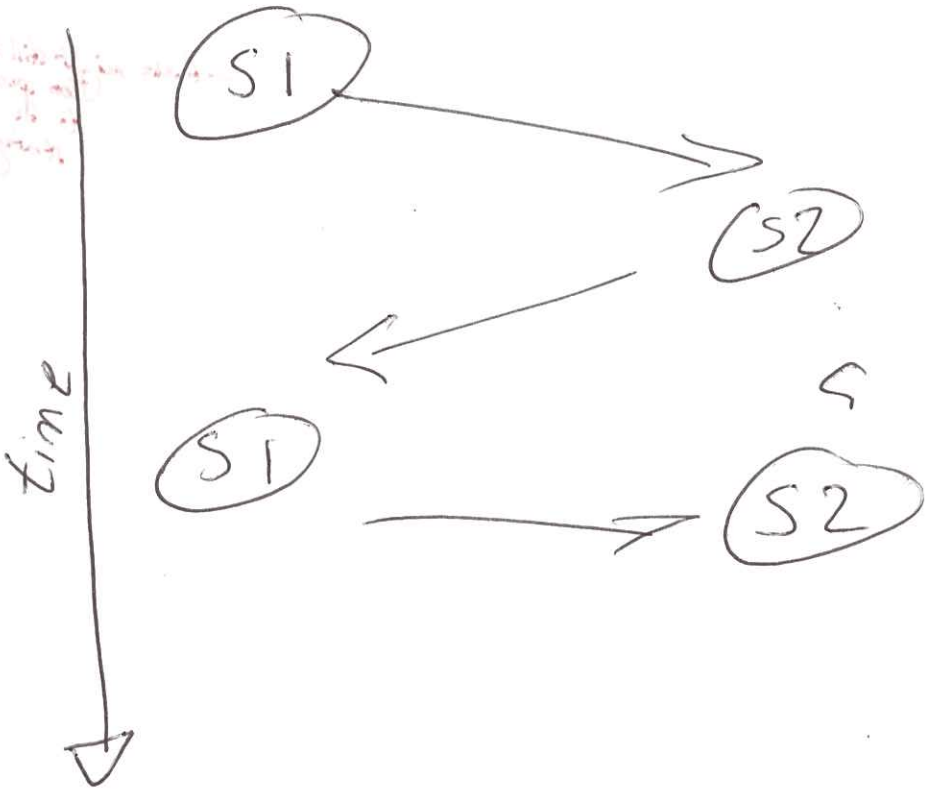



Figure 1.2



# Question 1: Finite State Machine

In Figure 1.1 we have a simple NPC whose default state is patrolling town. They have two variables that decrease at different rates when Patrolling, Hunger and Tired. They can stop patrolling to lower either of these by either sleeping/resting or Eating.

The Advantage of using a state machine is it allows an agent to keep track of a few needs and satiate them when nessecary. ✓ Unfortunatly when an Agent has a significant amount of needs that all have different requirements, it may cause an agent to be 'stupid' due to bouncing constantly between states trying to satiate too many needs at once (see figure 1.2) - not sure...if implemented well this doesn't have to be a problem

Use this side for Figures. Clearly Label and Caption each Figure. Refer to Figures in your answers.

Use this side for Figures. Clearly Label and Caption each Figure. Refer to Figures in your answers.