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: 9 / 5 / 2019 Name: Sam Huffer ID: 10163317 7

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	Q	Q
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.			
Overall Comments			

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. **Describe, discuss** and **compare** finite state machines (FSM) and rule-based systems (RBS), noting how they can be used for game AI. (Tip: Aim to identify the key differences between the two, and consider practical implementation issues. Tables and list are good.)

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. **Describe** and **explain**, using at least one specific example, how a game can be represented as a *state graph*, and how this can be used with search by game AI code. Use **figures** to support your answer.

4. *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).

5. **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.

6. 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.

7. **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.)

8. 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.

9. **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.

* Question 2: Balanced/unbalanced games & actions, & biased players

In games, if the actions available to a player would all grant equal advantage or disadvantage if chosen, then they are balanced (e.g. Fig. 1). If the available actions afford unequal advantage or disadvantage, they are unbalanced (e.g. Fig 2). Irrespective of how balanced the actions in a game are, if the same actions are available to all players, the game is balanced (Fig. 3). If different options are available to each player, and the options available don't confer equal advantage or disadvantage, the game is not balanced (Fig. 4).

A biased player will tend to favour particular actions or tactics in a game over others. (E.g. in Battleships, human players, when they score a hit, tend to guess in the surrounding area so they can sink the hit ship. Human players, knowing this, avoid clustering ships in one corner of the board so that other ships aren't hit while the opponent tries to sink another.) An unbiased player will use all actions or tactics equally or without preference (e.g. at random).

In terms of player experience, whether a bot is made biased or unbiased (or how it is biased) will depend on how difficult the bot is intended to be. (For example, a bot that makes random, sometimes poor, chess moves vs one that always moves the strongest piece they can vs one that makes the best move their algorithm can discern will vary in difficulty and be appropriate for different players with various skill levels. Another consideration will be if the bot is supposed to have a particular character or personality. For example, a happy berserker could be made biased towards (and even be better when) using short-range weapons or melee attacks in a FPS, where as a character preferring to be removed from danger might be biased towards long-range weapons.

* Question 6: Strategy and tactics.

A strategy is a broad, overarching goal that one wishes to achieve in a game or competitive scenario (e.g. win the game, don't die, take out as many players as possible), and can often be transplanted across different fictional and game genres (e.g.

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

Fig 1: outcomes of Rock Paper Scissors outcome against opponent's choice.

Player uses \ vs	R	P	S
R	/	X	✓
P	✓	/	X
S	X	✓	/

Fig 2: outcomes of Rock Paper Shotgun

you choose \ vs	Outcome against R	P	S
R	\	X	X
P	✓	\	X
S	✓	✓	\

Fig 3: a balanced game of Chess

R	kn	B	k	Q	B	kn	R
P	P	P	P	P	P	P	P

vs

P	P	P	P	P	P	P	P
R	kn	B	k	Q	B	kn	R

Fig 4: an unbalanced game of chess; perhaps one player has a handicap

R			k				R
P	P	P	P	P	P	P	P

vs

P	P	P	P	P	P	P	P
R	kn	B	k	Q	B	kn	R

"don't die" ~~are~~ is applicable in FPSs, RPGs, ~~and~~ RTSs, etc., and in fantasy, science fiction, noir, etc.). A tactic is a concrete step taken in ~~achieving~~ furthering one's strategy (e.g. capture that rook with a Bishop, collect a health pack to restore HP, fireball everything in the room), and will be context specific (e.g. one cannot cast a fireball at everything in a room in a non-fantasy game, or capture a rook with a Bishop in games that aren't akin to or derrived from chess) and applicable only in games of certain fictional or gamic genres that ~~do~~ have actions and materials required by that tactic.

Games AI can be made to use different tactics and take specific actions according to the current strategy they are employing, or the state they are in. For ~~example~~, in Halo, Elites will employ the tactic of "pacing" while in the state of "wait for player", "shoot" ~~and~~ ~~dodge~~ when in the state of "kill player", and "take cover" when in the state of "low health" or "shields are down". The state or ~~strategy~~, and therefore the associated tactics, will change based on the current ~~surrounding environment~~ game state (fig. 5). ~~In contrast, the Grunts will~~

The Grunts will also adapt tactics based on game strategy/state state, but will do so differently (fig. 6) to give them a different personality, as discussed above.

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

Fig-5: State graph for Halo's Elites

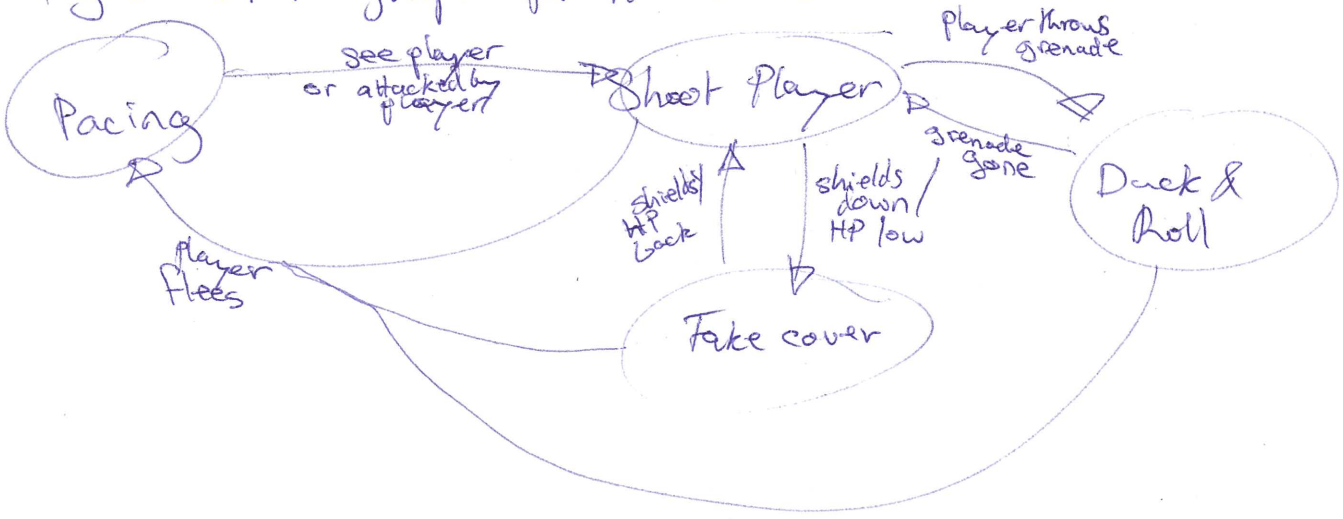
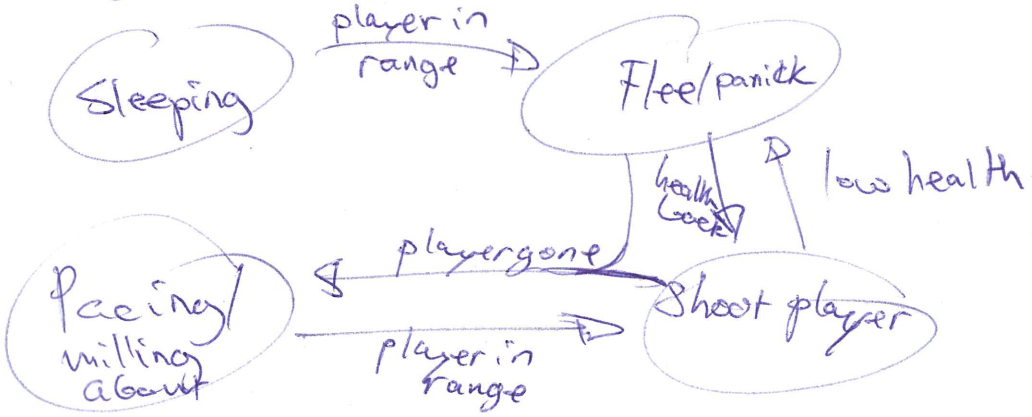


Fig.6: State graph for Halo's Grunts



Name: _____ ID: _____ Page ____

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