Assessment task 2

Outcomes covered 2 and 3

Assessment task instructions

This is an open-book project covering Outcomes 2 and 3. The project is broken down into two stages. Stage 1 is the program implementation and Stage 2 is the testing of the completed program.

You are required to create design documentation based on the client brief. All the Evidence Requirements which you must achieve are detailed after the client brief.

This project will be carried out under supervised and unsupervised conditions, ie you may work on this in your own time. The assessor will check the authenticity of any work you have done unsupervised. This may involve methods such as interviews, demonstrations, checking files, etc and may be carried out at random and pre-arranged times.

The assessor will specify the various deadline periods for the project. It is up to you to determine your own deadlines within these. You may decide to work on multiple tasks at the same time but you should try to fully complete and achieve one stage before completing the next. Applying this method of working is good preparation for the HND Graded Unit.

You should read all the evidence requirements for each stage and clarify any points with the assessor before you commence the project.

You are required to read the following brief and then complete the stages detailed below.

Client brief

Retro Games Ltd have commissioned you to design and develop a simplified version of Sokoban. Sokoban is a popular game by which a player pushes **crates** around a **map** to get them all in the right location. Sokoban is a single player game. The game is played on a 2 dimensional grid, but the rooms are not usually of regular shape. The edges of the room are indicated by a wall, and the player and boxes cannot get through the wall. There is a warehouse keeper, who the player must control in order to move the crates from their starting positions onto the diamonds. The **diamonds** are the end points for the crates. You can only push a crate when you are to one side of it and its opposite side is clear, which makes the task somewhat tricky for more complicated maps.

There are **walls** all around the map, and also in the middle in various configurations. Crates cannot be pushed through walls. Once a crate is up against a wall you can only push it along the wall, as you need to get behind a crate in order to push it. Once a crate is in a corner it is impossible to move it again. The warehouse keeper is unable to climb over crates, and is only strong enough to move one crate at a time. Crates can only be pushed, not pulled.

The game will require at least five levels. Each level should be harder to solve than the previous one, either by having more crates or obstacles, or tighter corridors, or a more complex starting arrangement of crates. The program should record how many moves a player takes to solve a level, and output this information visually.

Stage 1 — Analysis

You are required to analyse the system requirements from the client brief. This could be carried out using various techniques, such as Natural Language Analysis, interviewing the client, research into similar existing systems, or any other suitable techniques. The outcome of this stage will include:

- analysis documentation (NLA and/or CRC Cards)
- use case diagram
- four or more use case scenarios which include pre and post conditions, trigger event and the best case scenario flow of events. Alternative or exceptional behavior must be included in at least one use case scenario

Stage 2 — Static model

You are required to produce a static model of the system. The output of this stage is a class diagram that includes:

- visibility of attributes and operations (public, private or protected)
- specification of appropriate association, aggregation and inheritance relationships between classes

Stage 3 — Dynamic model

You are required to produce a dynamic model of the system. The outputs of this stage are:

- sequence diagram showing the flow of messages between three or more objects, for one use case
- construction of one other dynamic model diagram from a choice of — activity, collaboration, statechart, component. The diagram must be appropriate for the scenario