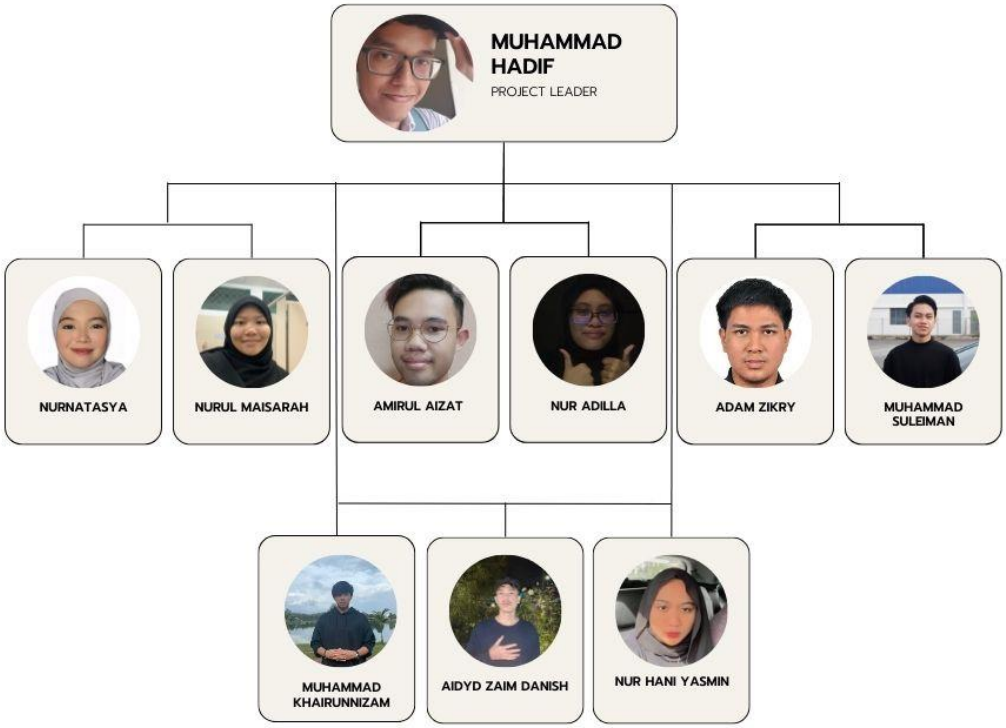
PROJECT PROPOSAL OBJECT ORIENTED PROGRAMMING

OBJECT ORIENTED PROGRAMMING
(TFB1033/TEB1043/TDB2153)

Lecturer name: DR. NORDIN ZAKARIA

NAME	COURSE	STUDENT ID
MUHAMMAD HADIF AZHAD	INFORMATION TECHNOLOGY	22007551
AMIRUL AIZAT	INFORMATION TECHNOLOGY	21001078
NURNATASYA BINTI MOHD SOFIAN	INFORMATION TECHNOLOGY	21001355
NURUL MAISARAH	INFORMATION TECHNOLOGY	21001383
NUR ADILLA ABU SAMAH	INFORMATION TECHNOLOGY	21000706
NUR HANI YASMIN YASMIN BINTI ANIM	INFORMATION TECHNOLOGY	21001207
ADAM ZIKRY LIZAM	INFORMATION TECHNOLOGY	21000893
AIDYD ZAIM DANISH AHMAD ZAMDI	INFORMATION TECHNOLOGY	21001559
MUHAMMAD KHAIRUNNIZAM BIN AMIR	INFORMATION TECHNOLOGY	21001806
MUHAMMAD SULEIMAN MOHAMAD RAZIP	INFORMATION TECHNOLOGY	21001379

ORGANIZATION CHART



The problem-solving approach is the process of finding a solution to a problem. There are many different approaches that can be used, and each has its own advantages and disadvantages. The most common approaches are breadth first, depth first, iterative deepening depth-first, and greedy best-fit.

Breadth first: Breadth first is an approach where the problem solver looks at all of the possible solutions and then chooses the best one. This can be very time consuming, but it is guaranteed to find the best solution.

Depth first: Depth first is an approach where the problem solver looks at one possible solution and then tries to find the best way to solve it. This can be faster than breadth first, but it is not guaranteed to find the best solution.

Iterative deepening depth-first: Iterative deepening depth-first is an approach where the problem solver looks at one possible solution and then tries to find the best way to solve it. If the solution is not found, the problem solver then looks at another possible solution and repeats the process. This can be faster than breadth first, but it is not guaranteed to find the best solution.

Greedy best-fit: Greedy best-fit is an approach where the problem solver looks at all of the possible solutions and then chooses the best one. This can be very time consuming, but it is not guaranteed to find the best solution.

Search: Search is an approach where the problem solver looks at all of the possible solutions and then chooses the best one. This can be very time consuming, but it is guaranteed to find the best solution.

BEST APPLICATION SCENARIO

Breadth first is best applicable when the problem solver wants to find the best solution and is willing to spend the time to explore all options.
Depth first is best applicable when the problem solver wants to find a good solution quickly and is willing to accept the risk that the solution may not be the best.
Iterative deepening depth-first is best applicable when the problem solver wants to find a good solution quickly and is willing to accept the risk that the solution may not be the best

Greedy best-fit is best applicable when the problem solver wants to find a good solution quickly and is willing to accept the risk that the solution may not be the best.
Search is best applicable when the problem solver wants to find the best solution and is willing to spend the time to explore all options.

Differences

The main difference between the approaches is the amount of time they take to find a solution, and the guarantee of finding the best solution.

Breadth first is guaranteed to find the best solution, but it is very time consuming.
Depth first is not guaranteed to find the best solution, but it is faster.
Iterative deepening depth-first is not guaranteed to find the best solution, but it is faster.
Greedy best-fit is not guaranteed to find the best solution, but it is faster.
Search is guaranteed to find the best solution, but it is very time consuming.

Similarities

All of the approaches explore the search space to find a solution.

After many days of discussion, research and brainstorming, us as a group have decided to develop a Pacman-styled android video game, complete with a playable character, maze and enemy AI. However, we've decided that to make this more interesting and unique, we should include a twist within our product, and that is to make it a sort of Horror game. How we will do this is still rather unclear to the team, though with further planning we should be able to deliver a sufficient result.

The goal of the game is to collect all the points in the maze and avoid the ghosts. The Pacman is animated in two ways: his position in the maze and his body. We animate his body with four images, depending on the direction. The animation is used to create the illusion of Pacman opening and closing his mouth. The maze consists of 15x15 squares. The structure of the maze is based on a simple array of integers. Pacman has three lives. We also count the score.

We decide whether a game is complete, which designates that it belongs to the category of "hardest" games. He follows a really simple strategy. We begin by presenting a series of arguments that demonstrate how each video game with a particular set of gameplay characteristics belongs to a particular complexity class. The games are then categorised by the characteristics of how they may be played.

For instance, in one kind of game, the player travels over a terrain, stopping at various sites. We refers to this as "location traversal," and an example would be a game where the objective is to acquire every item scattered throughout a landscape. In several games that include location exploration, each place can only be visited once. Downhill races might be a part of games with "single use paths."

Afterward, we utilise graph theory to demonstrate that any game with location traversal and single-use pathways is NP-hard, which is the same level of complexity as the travelling salesman issue. Pac-Man actually fits within this category (the proof involves distributing power pills around the maze in a way that enforces single use paths). We demonstrate how games may be divided into additional complexity levels. For instance, if each door is controlled by two pressure plates, games that use pressure pads to open and close doors are difficult on the PSP. This group includes Doom.