**Pseudocode**

**Decomposition**

*Purpose*

To create a classic arcade game Pacman where you control Pacman around a maze, while trying to eat kibbles and avoiding ghouls.

*Description*

Gameplay – Pacman will have three lives at the start of the game and will lose a life if it is eaten by the ghouls. If Pacman loses all its lives then the game ends with the player loosing. For the player to win, Pacman must eat all the kibble on the screen by moving over them.

Pacman – Pacman character is controlled using the arrow keys and will move one grid space per key press. Pacman cannot move through solid walls of the maze and will lose a life when collide with the ghoul characters. Pacman will start off with three lives and the game will end when all three lives are lost. Pacman will also eat the kibble that exist throughout the maze causing the kibble to disappear and granting the player 1 point. There is only one Pacman per game.

Ghouls – The four ghoul characters will start in all four corners of the maze and will move in a random direction at the start. The ghouls will start to chase after Pacman at a 50% chance when their vertical or horizontal position are align. If the ghouls collides with Pacman, Pacman will die and the player will lose one life. The ghoul character cannot move through walls and when the character collide with a wall, its direction will be change randomly. There are four ghoul characters in the game.

Kibbles – The kibbles will disappear when Pacman collides with it and will grant the player 1 point that will tally up throughout the game. The kibbles will occupy all the spaces that is not a wall in the maze at the start of the game.

Maze – Using the inherit data grid view properties, a maze is a 21 by 21 grid which are made up of walls, kibble and blank spaces. The maze is the space that all the characters will move around in.

**Extra features**

Menu – Player can use the menu to start a new game, pause the game and exit the game. New game cannot be start while the game is playing, if the player wish to start a new game, the player must pause the game first. The menu also shows the score and how many lives Pacman has left.

Lives – Pacman’s lives are display using three picture boxes with each picture box representing a life. The picture will become invisible with each Pacman’s death leaving the current amount of lives left on the screen.

Sounds – Adding sounds such as intro music, Pacman eating kibble and Pacman death to make the game more aesthetically pleasing and further engage and immerse the player.

Ghouls chasing Pacman - The ghouls will start to chase after Pacman at a 50% chance when their vertical or horizontal position are align. This ensure that the ghouls can trap Pacman instead of moving around the maze aimlessly.

**Form Design**

**Abstraction**

Character class – This class is the base class for all the characters in the game.

Pacman class – This class will control the behavior specific to the character Pacman. This class will also inherits from the character class.

Ghoul class - This class will control the behavior specific to the character ghoul. This class will also inherits from the character class.

Maze Class – This will inherit the data grid view component which will allow the maze to be form into a 21 x 21 grid by using images with data grid view properties.

Controller class - This will control the gameplay and game functions of the program

Form class - This will contain all the event handlers for the components in the form. The event will be how the user interacts with the program and the handlers will be the action that occurs from the event.

**Encapsulation**

inherits

Character

Bitmap characterImage

Maze maze

Point position

Point nextPosition

Direction direction

Character(Bitmap characterImage, Maze maze)

SetNextPosition()

Move() abstract

Draw() abstract

CheckForWall()

GetGridCellForPosition(Point point)

MoveOutsideMaze()

Ghoul

Random rand;

bool foundPacman;

Ghoul(Bitmap characterImage, Point position, Random rand, Maze maze)

SetDirection()

ChasePacman(Point pacmanPosition)

CheckForPacman(Point pacmanPosition)

Move() override

Draw() override

Pacman

int score

int life

SoundPlayer eatKibble

Pacman(Bitmap characterImage, Maze maze)

Move() override

Draw() override

EatKibble()

Drawn on

Form

Maze maze

Controller controller

Random rand

bool pause

Bitmap k

Bitmap w

Bitmap b

SoundPlayer intro

SoundPlayer win

PrivateFontCollection pfc

Form()

timer1\_Tick()

Form1\_KeyDown()

button1\_Click()

button2\_Click()

button3\_Click()

Maze

string map;

int nKibbles;

Bitmap wall;

Bitmap kibble;

Bitmap blank;

Maze(Bitmap k, Bitmap w, Bitmap b)

Draw()

Controller

Maze maze;

Pacman pacman

List<Ghoul> ghouls

Label scoreLabel

bool gameEnd

bool gameWin

SoundPlayer pacmanDeath

Bitmap pacmanMouthOpen

Bitmap pacmanMouthClose

Controller(Random rand, Maze maze, Label scoreLabel)

StartNewGame()

StartNewLife()

GamePlay()

MovePacman(Direction direction)

PacmanAnimation()

SetLives(PictureBox live1, PictureBox live2, PictureBox live3)

DrawGhouls()

MoveGhouls()

CheckForPacmanGhouls()

ChasePacman()

CheckNumberOfKibble()

Has a

Has a

**Iterative Refinement**

*Character*

SetNextPosition() - setting the next position according to the direction set from the arrow keys press

CheckForWall() - checking to see if the next position is a wall

Move() - abstract method that is never use but serves as a structure

Draw() - abstract method that is never use but serves as a structure

GetGridCellForPosition(Point point) - setting the point as a grid reference

MoveOutsideMaze() - setting the next position to the furthest cell on the opposite side when any character reaches the edge of the maze

*Pacman*

Move() - moving Pacman by one maze grid

EatKibble() - making the kibble disappear when Pacman moves over it

Draw() - drawing the Pacman onto the maze

*Ghoul*

SetDirection() - setting the direction that the ghoul will move in

ChasePacman() - changing the direction of the ghoul to chase after Pacman if their vertical or horizontal position are the same

Move() - moving the ghoul by one maze cell

CheckForPacman() - checking to see if ghoul position is the same as Pacman position

Draw() - displaying the ghoul image onto the maze

*Controller*

StartNewGame() - setting up for a new game

StartNewLife() - setting for a new round once Pacman has died

GamePlay() - the event that occurs every timer tick, this is when the game is running

MovePacman(Direction direction) - moving and setting Pacman's direction according to the arrow key press

PacmanAnimation() - changing the Pacman's image between mouth open and close to create an animation

SetLives() - setting the visibility of the three pictureboxes according to how many lives Pacman has left

DrawGhouls() - drawing all four ghouls onto the maze

MoveGhouls() - moving all four ghouls around the maze

CheckForPacmanGhouls() - getting all four ghouls to check if its position is the same as Pacman's position

ChasePacman() - getting all four ghouls to chase after Pacman

CheckNumberOfKibble() - checking to see if all the kibbles are gone, if all the kibbles are gone the player wins

*Maze*

Draw() - drawing the maze onto the form

Form

timer1\_Tick() - timer tick event

Form1\_KeyDown() - key press event

button1\_Click() - new game button click event

button2\_Click() - pause and start button click event

button3\_Click() - exit button click event