**SuperPacmen project**

//-----------------------------------------------

2.Our names and ID numbers :

Afek Sulimani , 211417035

Dvir Karni , 209494053

//----------------------------------------------

3. explanation of the exercise:

In this exercise we created a Super Pacmen game.

The game starts with an opening screen, The screen offers the user to start a game, read information about the game and exit, selection is made by clicking with the mouse on the wanted object.  
Clicking on the information button will display all the game objects and their abilities.

Clicking the start button will load the first step file, and the game will begin.

There are few different levels, some with a countdown clock, if the user has not finished the level and the time has passed, he will have try again the level.

Level up in the game when the player has eaten all the cookies.

The game will end if the player dies 3 times and lost, or when the player finished all the levels of the game and won!.

The levels in the game are managed by a file called Playlist.txt which contains a number (how many levels in the game) and the names of all the levels by lines.

In our game we created 4 stages. To add a level, use the level editor we created in exercise 4, add the time of the level to it in the first line or 0 if there is no time. and add it to the Cmakelist line.

The game is clock driven, every move is made in real-time, there are no queue in this game. This means that some games will experience pacman who will be able to go faster , as well as demons

In the game there are 4 types of gifts which will be detailed on the opening screen.

At any time in the game, except transition screens, the player's status will be displayed and will present: The number of the level, what is the player score, how many lives he has left, how much time has passed since the start of the game, and (if the level is limited in time) how much time is left for the stage will also be displayed.

In addition there is animation of the SuperPacman and dash mode, that show the cooldown skills status.

In the game there are doors that will block the normal Pacman from passing through them, in order to be able to pass through them, he can take a key that will open one door without knowing which one. When the player is in super mode he will be able to break the doors without taking a key.

Pacman has a dash skill that allows him to dash for a few seconds. It can be activated by using the ALT button. In addition, in super mode will the player be able to attack the demons and kill them. Using the Ctrl (LEFT) button.

Colliding with alive demon in attack mode will kill the demons .if attack mode is NOT activated the pacmen will die, even in super pacmen mode.

Any normal demon can take one gift (no matter what kind) ,And so they will become smart demon.

In smart mode they try to hunt the player and will becomes faster than normal,

They will be able to move throw doors.

//----------------------------------------------------

4. A list of our files: more about each file in the degsin

------------------------------------------------------------------------------------------

Singleton class:

* Datamanger.h Define the Datamanger class, with functions , members and enum
* Datamanger.cpp Implements the functions and constructors of the class.
* MusicManager.h Define the Datamanger class, with functions, will hold the game soundBuffers as a public member.
* MusicManager.cpp Implements the functions and constructors of the class.
* TimeManager.h Define the TimeManager class, with functions , members, this game has many time related function we decided that its best to keep it in a class.

-------------------------------------------------------------------------------------------

* AddLivePresentState.h/.cpp define and Implements the function, members and constructors of the class
* AddTimePresentState.h/.cpp define and Implements the function, members and constructors of the class
* Backround.h/.cpp define and Implements the function, members and constructors of the class
* BackroundItems.h/.cpp define and Implements the function, members and constructors of the class
* Board.h/.cpp define and Implements the function, members and constructors of the class
* Controller.h/.cpp define and Implements the function, members and constructors of the class
* Cookie.h/.cpp define and Implements the function, members and constructors of the class
* DataManger.h/.cpp define and Implements the function, members and constructors of the class
* DeadDemonState.h/.cpp define and Implements the function, members and constructors of the class
* Demon.h/.cpp define and Implements the function, members and constructors of the class
* DemonState.h/.cpp define and Implements the function, members and constructors of the class
* Door.h/.cpp define and Implements the function, members and constructors of the class
* DynamicObj.h/.cpp define and Implements the function, members and constructors of the class
* GameObj.h/.cpp define and Implements the function, members and constructors of the class
* Key.h/.cpp define and Implements the function, members and constructors of the class
* KillDemonsPresentState.h/.cpp define and Implements the function, members and constructors of the class
* Level.h/.cpp define and Implements the function, members and constructors of the class
* Location.h/.cpp define and Implements the function, members and constructors of the class
* menu.h/.cpp define and Implements the function, members and constructors of the class
* musicManager.h/.cpp define and Implements the function, members and constructors of the class
* NormalDemonState.h/.cpp define and Implements the function, members and constructors of the class
* NormalPacmenState.h/.cpp define and Implements the function, members and constructors of the class
* Pacmen.h/.cpp define and Implements the function, members and constructors of the class
* PacmenState.h/.cpp define and Implements the function, members and constructors of the class
* Present.h/.cpp define and Implements the function, members and constructors of the class
* resentState.h/.cpp define and Implements the function, members and constructors of the class
* SmartDemonState.h/.cpp define and Implements the function, members and constructors of the class
* StaticObj.h/.cpp define and Implements the function, members and constructors of the class
* SuperPacmenPresentState.h/.cpp define and Implements the function, members and constructors of the class
* SuperPacmenState.h/.cpp define and Implements the function, members and constructors of the class
* TimeManager.h/.cpp define and Implements the function, members and constructors of the class
* Wall.h/.cpp define and Implements the function, members and constructors of the class

//----------------------------------------------------------------------------

5. Structs:

Location Structures: used in almost every member in the current program, has a few unique function that helped us to compare and assign new locations with the board. By converting from Location that works with strings and sf::Vectorf from sfml.

GameTexture.h will hold the textures of the game, will also have a function that load them. The struct will be hold by the Singleton class DataManger

//----------------------------------------------------------------------------

6. Algorithm:

The smart demons algorithm. The demons will to hunt the pacmen by calculating the closest rout to him, the closest rout will be based on sqrt(x^2 +y^2), if the demons were blocked then they will save the bloked direction and previous direction and will go to the closest one that are not one of those 2. The pacmen may get into a loop where he can’t move so we made another function that if the pacmen is in this loop he will go to a random good diraction

//--------------------------------------------------------------------------

7. design:

* Controller: will control the game, will hold the game clock and will reset him after every 1.0 seconds, will also change levels, call the menu and close the game if asked.
* Menu: will be responsible on the menu, the controller class have object of this class.
* Board: will be responsible on the board itself, will hold a vector of unique ptr to base class GameObj, those will be used to handle collision with the dynamic obj.

The class will also hold unique ptr to a demons objects and an object to the player (pacmen) class. The class will be responsible to delete objects, handle them and reset the board if needed.

* GameObj, will be the base abstract class of the every object in the board
* (key, door, present, cookie, pacmen, and wall), the class will hold as a private member the sprite and will export functions that will handle this sprite (such as move, draw..) .

The GameObj class declaring all the pure virtual function of the HandleCollision between the GameObj objects.

* DynamicObj will be another base abstract class of the dynamic object in this game, this class will Inherit from the GameObj class and will also export functions of animations and other moving attributes.
* Pacmen: will Inherit from the DynamicObj, is responsible on any attribute of the pacmen. The pacmen will hold an unique ptr to the pacmen state base class which will use polymorphism to act according the current pacmen state
* PacmenState: will be abstract base class of the pacmen state, will define function that will inform about the state, and will act in different ways when colliding with demon and a door

There are 2 pacmen states: super pacmen and normal pacmen.

* Demon: will Inherit from the DynamicObj, is responsible on any attribute of the demon. The demon will hold an unique ptr to the demon state base class which will use polymorphism to act according the current demon state
* PacmenDemon: will be abstract base class of the demon state, will define function that will inform about the state, and will act in different ways when colliding with pacmen and a door, will also move different (most of the state is for the different ways between the demons)

There are 3 demon states: smart demon ,normal pacmen and dead demon(temporary).

* staticObj: will inherit from the base class GameObj, most of this class purpose is to Implements the collision between the static object collision on the board (they cannot collide)
* Present: this will inherit from the staticObj, will hold a state for 4 different presents.
* DataManger will manage most of the game-related information and data, will mostly receive information about the game and will export the information with public functions, the data manger will hold 2 private members of another managers, which are musicManger, TimeManger. Will also hold the gameTexture struct.
* The program is using a double dispatch so the game will be able to act differently between every collision, the double dispatch is between the GameObj objects, and the collision function is called by the board that will call each of the moving objects HandleCollison function with any other object on the board, that will be possible by using polymorphism of the gameObj objects.
* There are another 2 classes that we want elaborate about, and that are Level: that reading a new level and saving the level information in a string, and backrounf that handle the backround of the board

//-------------------------------------------------------------------------

8. Known bugs

//-------------------------------------------------------------------------

9. notice:

- We have private members that are not private in the musicManger and in the GameTexture, we want! this members to be public, the objects in this game need to access them (the members are Textures and SoundBuffer only!)

- All of the following is to improve the gameplay experience:

- The board boundaries are existing and the pacmen cannot go throw them.

- Super pacmen CAN! Be killed by demons if he is not attacking them

- Demons CAN! Take a present, but only in the normal mode of the demons

- Pacmen will stay a super-pacmen after leveling up (the pacmen is holding is current skill state)

-To make the game feel smoother we used isKeyPressed and we had a function that return a smaller intRect of each sprite.

- smart demon cannot collide with each other.

- It really worth to win the game! Don’t give up.