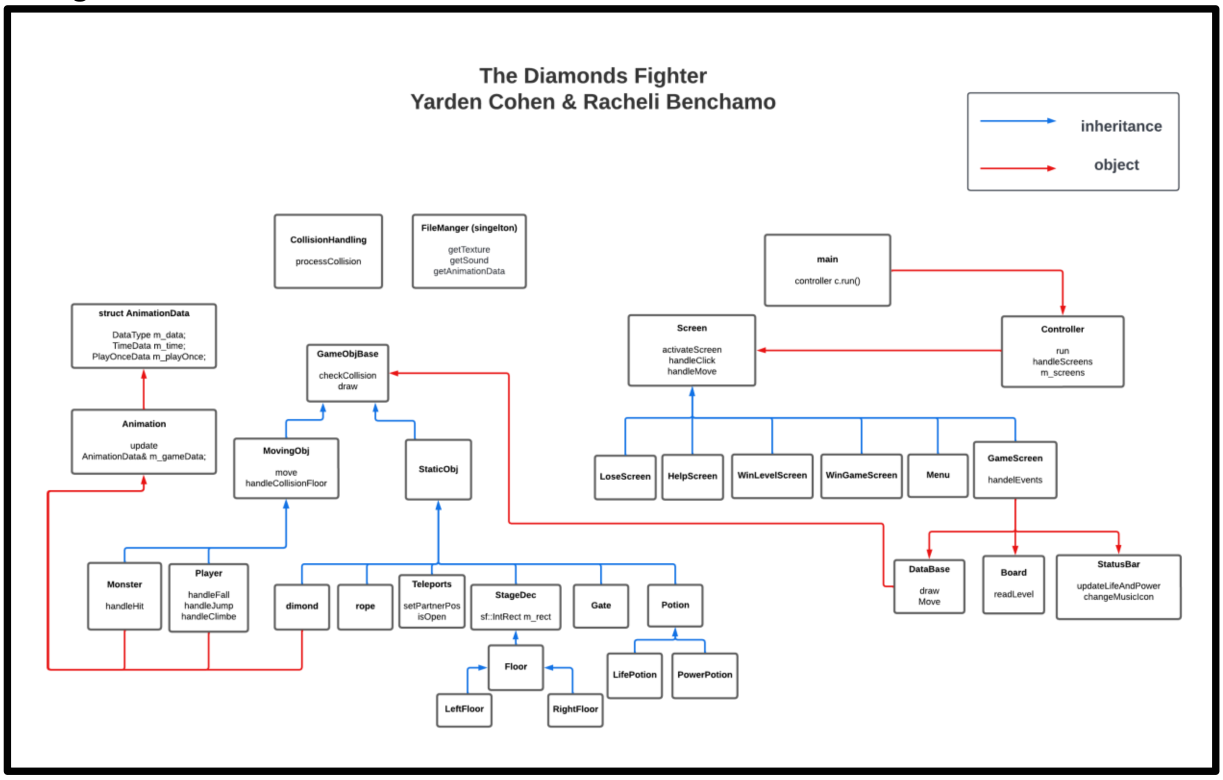
README
The Diamonds Fighter- oop2

Racheli Benchamo, ID: 209179738
Yarden Cohen, ID: 315177832

Design:


The Diamonds Fighter
Yarden Cohen & Racheli Benchamo
inheritance
object
CollisionHandling processCollision
FileManger (singelton) getTexture getSound getAnimationData
main controller c.run()
struct AnimationData DataType m_data; TimeData m_time; PlayOnceData m_playOnce;
GameObjBase checkCollision draw
Screen activateScreen handleClick handleMove
Controller run handleScreens m_screens
Animation update AnimationData& m_gameData;
MovingObj move handleCollisionFloor
StaticObj
LoseScreen
HelpScreen
WinLevelScreen
WinGameScreen
Menu
GameScreen handelEvents
DataBase draw Move
Board readLevel
StatusBar updateLifeAndPower changeMusicIcon
Monster handleHit
Player handleFall handleJump handleClimbe
dimond
rope
Teleports setPartnerPos isOpen
StageDec sf::IntRect m_rect
Gate
Potion
Floor
LeftFloor
RightFloor
LifePotion
PowerPotion

General Description
The Diamonds Fighter is a 2D platform game with an emphasis on exploration and collecting diamonds to open the gate to the next stage.
The player controls male or female character.
The character, must jump, climb, fight the monsters and use various other abilities to navigate the game's world and finish the different stages.

Input format:
The program accepts as input in the Resources folder a file containing all the stages when a space line separates a stage from another, and where in the first row of each stage appears the size of the board and the type of that stage (forest, desert, or snow).
(Meaning- rows, cols, type).
Also each level have to contain one gate and even number of teleports.

Game rules:
Collect all the diamonds and enter the gate to the next stage while fighting the monsters.
Each collision with a monster reduces the amount of life of the player depending on the .type
Every fight (by pressing X) against a monster reduces the amount of power a player has.
If the player runs out of life, he dies and must repeat the stage anew.

The player can replenish the stock of life and power by pressing the Z button when standing next to a potion that fell from a monster, he/she killed.
When the player is standing next to teleport, he can click the UP-button end enter it and re-emerge where the teleport partner is located.

Description of the classes:

Controller-
The main class whose role is to manage the game.
The department manages the game screens and makes sure they appear in the right order at the right time.

FileManager-
Manages the backgrounds of the game, the texture of the objects in the game, the font and sounds of the game.
The class is implemented using a design pattern called singleton, so that it is possible to access images and sounds with a pointer to the same aids required from anywhere in the program.

Screen-
The base class of the various **screens** in the game. It has functions for loading basic buttons for screens and for a basic title.

GameScreen-
manage the game screen.
The class manages the stages of the game as well as the events in the game.
Holds m_dataBase which holds the relevant objects at each stage and takes care of their movement and collisions.

MenuScreen-
Manages the menu window at the beginning of the game
You can select an explanation window, start game, and exit game. You must select a player before starting a new game.

HelpScreen-
You can read game information or return to the menu screen.

WinLevelScreen-
you can move to the next stage or return to the main menu.

WinGameScreen-
you can exit the game or return to the main menu.

LoseLevelScreen -
you can repeat the stage or return to the main menu.

Board-
read the stage and the number of diamonds required to collect in it from the Levels.txt input file and sends the level size and diamonds to the DataBase class.
Holds a pointer to the text file, and a vector holds the size of the current stage.

StatusBar-
Manages the game status bar:
How much life and power were left for the player, how many diamonds he collected from the required amount and Current stage number. There are also four more buttons: Music on / off button. Sound on / off button. Stop / Continue button.
Button for starting the current stage from the beginning.

DataBase-
Manages everything related to the vectors of the objects.
It initializes its objects, manages the displacement, collisions and displays them on the game board.
Connects the teleport cells by guerrilla to each cell to which cell it will be sent (other than itself) when a character stands on it.
The class holds unique_ptr vectors of player, monsters, rope, teleport, and static objects.

GameObjBase –
An abstract base class. The role of the class is to possess and maintain the qualities which the classes inheriting from it use.
Such as draw, getSprite, and the function of a collision test between two objects of this type. Holds sf:: Sprite.
Classes inherited from GameObjBase: MovingObj, StaticObj

MovingObj-
An abstract intermediate class whose rule is to hold and maintain the properties of the moving objects in the board.
Such as the move function and the animation of the player and the monsters.
Holds sf: Vector2f which retains the previous position of a time figure should be returned in case it moves to a place it cannot move to.
Classes inherited from **movingObj**: Player, Monster.

Player-
The class that manages the main player in the game. Manages the jump,fall and climb as well as the cases where he fights, collects a diamonds or is hit by a monster.

Monster-
The class that represents the monsters. Manages the random movement of monsters on the board as well as the situations in which they are hit by the player or die.

StaticObj-
An intermediate class that manages the properties of the static objects in the game board.

Classes inheriting from **StaticObj** : Gate, Diamond, StageDec, Potion, Teleport, Rope.

Gate-
The class that represents the gates in the game.

Diamond-
The class that represents the diamonds in the game.

Rope-
The class that represents the ropes in the game.

Teleport-
The class that represents the teleport cells.
Holds sf: Vector2f which saves the position of the snap cell of its partner.

Potion-
The class that represents the potions in the game. There are life potion and power potion and they have classes that represents them.

StageDec-
The class that represents the decoration in the game as trees rocks and floor. There different classes for the floor, the left and right edge floor that represents them.

Stage file format:
The file name will be Level.txt
The first row will show the size of the stage and the type of the stage.
F-for forest, S-for snow and D for desert.
The map of the stage will then appear.
An empty line will separate between stages!

The symbol of the object in the stages file are:
Player ='P'
Empty = ' '
Diamond = 'D'
Rope = '|'
Teleport= 'X',
Middle floor = '='
Left edge floor = '('
Right edge floor = ')'
Big tree = 'T'
Small tree = 't'
Rock = 'R'
Small monster = '1'
Middle monster = '2'
Big monster = '3'
Gate ='#'

Data structure:
The DataBase class holds unique ptr vectors for the figures, static objects, monsters, rope, and teleport cells.

Other:
No bugs were aware of.

Link to the game video: