# LodeRunner

Lode Runner project with SFML.

Project creators:

Alexander Yershov 319140943

Anael Zateikin 207190216.

This is the Lode Runner game created with SFML.

The classes in the project:

1. Main classes: Controller, Board

The main classes are responsible for the game logic- They communicate between one another.

Board - Responsible for keeping the board, file stream and a vector of all the static objects of which the map is constructed by.

The class functions include a function that translates ascii character into a name of object, functions that organize all the needed actions to manipulate the static object vector and functions that extract information and export it to controller.

The class has functions that receive a moving object as a parameter, checks collision and handles accordingly. In this function, in the case of no collisions, it makes the object fall.

Controller - Responsible for running the game loop. In the game loop, the game window is displayed, events of digging and mouse pressing are handled, movement of moving objects and checks game status (won\lost).

More functions that are included in controller: creation of moving objects, handling case of new game, new level and strike.

The members of controller class are the moving objects vector, Board type member, caption, timer and start window.

2. Assisting classes - ResourcesManager, Macros

ResourcesManager - This class is a singleton responsible for loading the textures, soundbuffers and font. The decision to make this class a singleton was so it can be easily accessed from every class in the project. This class is also responsible for achieving the animation of the moving objects.

Macros - Assembles all the constant declarations and enum classes declared in the project. Also has a function that checks if an object is static or not.

3. GameObject - the root of the tree(inheritance) hierarchy.

This class' constructor sets the sprites' basic information (position, texture and scale). The class also has a draw function.

All the objects the construct the map inherit from this class, It is an abstract class.

4. Split into two types of objects- MovingObject, StaticObject

MovingObject - derived from GameObject, an abstract class. This class contains members of the object's locations, direction and state of falling.

The class has a pure abstract function 'move' and a fall function.

StaticObject - this is an abstract class derived from GameObject. This class only contains virtual functions for handling collisions.

5. Static - Floor, Bonus, Wall, Coin, Bar, Ladder.

All those classes are derived from StaticObject.

Floor - Has two virtual function of collision with moving objects. Also, the class has the dig function - removing the floor for a limited time.

Bonus - a superclass that separates into 4 mini classes, each class represents a different kind of bonus. In each derived bonus class, there is a specific handle collision with the player for that bonus.

Wall - contains functions that commit double dispatch with a moving object.

Coin - All coins must be collected to finish a level. Contains a handle collision function that erases the coin from the static-object vector stored in the board and increases the score in the game.

Ladder - Has two virtual function of collision with moving objects.

6. Moving - Player, Enemy

Both classes derive from MovingObject.

Player - Defines the behavior of the player and stored in the moving objects vector. The game by definition only has a single player.

The class contains a unique member 'lives' and the functions 'move' and all the collision handling with the game objects.

Enemy - Defines the behavior of the enemy and also stored in the moving objects vector. Each level loads a different amount of enemies and sets each enemy into one of the three objects (sub-classes of Enemy class): SmartEnemy, MediocreEnemy and DumbEnemy. Each of those enemy kinds has a different moving functionality - while the smart enemy follows the player, the mediocre goes sideways until bumping into a wall and then changes direction and the dumb one randomly switches directions every few seconds.

Files in the game: (All located in Resource folder)

Font - SundayMorning.ttf

enemy and player images.

music icon images.

static objects images.

board.txt - a text files that contains all the levels in the game, separated with a new line.

Main data bases:

* m\_movingObj - vector of all moving objects, located in Controller.
* m\_staticObj - vector of all static objects, located in Board.

Together the two vectors make up whole map.

Notes:

1. All the levels of the game are stored in a single file.

To add a file, the user must copy the new board and paste it into the file "board.txt".

The user MUST make sure that there is an empty line before the new inserted level and make sure that there is an empty line at the end of the new board (meaning the last row of the board must

end with a newline).

2. The dig action removes the floor completely and if a character walks on that floor, it will fall through the missing floor. A character cannot step on another character stuck in a pit.

3. When a character falls - it cannot move sideways until it reaches a valid surface to stop the fall.