**Oop2 – project – Dangerous Dave**

Eliav Aviad: 032543415

Elhanan Aviad: 312292105

**Game description:**

The mission of the game is to guide Dave through the levels, collecting trophies in the hideout of his enemy, while collecting as many points as possible.

**Design:**

The basic class is GameObj, which bequeaths to all static and movable objects.   
The Controller is the game manager, start the game and managing all the Continuity/progress of the game. We load the levels from generic files names (making it easy to add levels) and insert them to the Board members. The Board, held in the Controller, holds the game objects and responsible for most of the game actions, such as movement,  
adding new objects, collisions etc.

**List of files created by us:**

Animation – execute animation for all objects.  
Board – class to store and manage board of the game, holds all the entities of the game.  
Bullet - mobile objects, the player and enemies shoot it.  
Button – button that can be pressed.  
CircularEnemy - enemy with Circular movement.  
Collision - Author: Nick (original version), ahnonay (SFML2 compatibility), https://github.com/SFML/SFML/wiki/Source:-Simple-Collision-Detection-for-SFML-2

CollisionHandling - MultyMethod collision class.  
ConstEnemy – enemy with constant movement.  
Controller – class that run and manage the game.  
Dead - Created when enemy\player dies.  
Diamond – points to the player.  
Door – the finish point of the level.  
Enemy – bequeaths to Const/Circular enemy  
Fire – plain fire.  
Flame - flaming fire.  
GameObj – all entities come from here, holds animation & sprite members.  
Gun – allows the player to shoot.  
Jet – let the player the ability of flying.  
JetBar – represent the duration of the jet.  
macros (no cpp) – holds all the game consts.  
Menu - Class of the game menu.

Moveable - all mobile objects come from here.  
Player – the player of the game and his movements.  
Sounds - sound singleton resource.  
Static – all immobile objects come from here.  
Stats – to display the game stats.  
Stone – still stone object.  
Textures - textures singleton resource.  
Trophy – player take to be able to finish the level.  
Wall - still wall object.

**Main data structures:**

Vectors of unique\_ptr – to store al entities.  
Maps – for the Textures & Sounds.  
Pair – for the buttons.

**Notable algorithms:**

Basic physics-based Movement algorithm for the moveable.  
Const enemy change direction (left/right) when hits wall/stone.  
Circular enemy change direction (all directions) after fixed distance.  
when moveable object dies, a dead object is created to animate explosion.  
Usage of a pixel perfect test for collisions that requires higher precision.  
the game collision method is MultyMethod collision.

**Known bugs:**

Unknown.

**Other comments:**

Unknown.