**OOP2 – project – Hungry shark**

Max Dunkel: 321559775

Philip Berlin: 316339761

**Game description:**

Using arrow keys to move a shark (player's character) and eating sea creatures to get bigger and eat smaller sharks.  
To win the game the players needs to survive enough time till all the other sharks hit mine or eating by the player.  
Sharks will swim towards off you when there are in a certain distance away from your shark.  
If not eating, hitting a mine or getting hit by another sharks the player's health reduces,   
when health gets to 0 the player loose.

**Design:**

Base class GameObj, which gives to all static and movable objects.   
The Controller is the game manager, start the game and managing all the Continuity/progress of the game. The game load the level from generic files name 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, collisions etc.  
class enemyShark inherits from class moveable and class enemy inherits from class enemy shark.   
  
**List of class:**

Board – class to store and manage board of the game, holds all the entities of the game.  
Controller – class that run and manage the game.  
GameObject – all entities come from here & sprite members.  
macros (no cpp) – holds all the game consts.  
Menu - Class of the game menu, loads button: start, help, exit.  
Movable - all mobile objects come from here.  
player – character of the game and his movements.  
Static – all immobile objects come from here.  
Gamestats – to display the game stats.  
Textures - textures singleton resource.  
Text – setting in game text.  
stone – acts as wall object.  
enemyShark – enemy that chases the players character (using dynamic cast).  
enemy – sets movement for enemy shark.  
Sounds - sound singleton resource.  
LifeBar – health status of the player become fuller when shark eat and reduced when hit mine or another shark.  
Animation – animation base class.  
Collision - outsource for sub class Author: Nick (original version), ahnonay (SFML2 compatibility)  
CollisionHandling – main class for collision between objects  
Crab, jelly, Lionfish, Nemo, Orchid, PerotFish, PufFish - each of them is sets the movement of sea creatures.  
Mine – static object that explodes when detecting collision with any shark.  
explode – explosion animation after hitting a mine sits under static objects.  
Spawn – set of animation on players starting point.  
Splash – set of animation of waves as static object.

**Main data structures:**

Vectors of unique\_ptr – 1) hold static object 2) hold fishes 3) hold sharks.  
Vector of chars - to load the level once, thus able to restart level from vector.

**Notable algorithms:**

Basic physics-based Movement algorithm for the moveable.  
Fishes and other sea creatures change direction (up/down/left/right).  
Smart enemies: if the enemy shark at certain distance from the player he start moving towards him else moving randomly and eating fish.  
The game collision method is PixelPerfectTest.

**Known bugs:**

Unknown.

**Other comments:**

We tried to create layer background combined of 4 images it worked to some degree.

[Link to YouTube](https://youtu.be/gB_JwIh4Xdw)