MADMOO ARCADE APPLICATION STRUCTURE

In terms of Panda JS the game will be divided in three scenes which are initiated in the main.js file.
- The first scene will be loaded right after the assets have finished loading and will contain some background and buttons to start the game and go to the final scene.
- The second scene will contain the game itself. There we will draw all the game elements like the dog and the ducks and execute the mechanics of the game.
- The third scene will show the highest score and some background image.

Main game components

- The "**Round**" class – puts a game round together. This class holds a reference to all other object that are used in the game. It is responsible for creating the "Dog" object at the beginning of the round and managing the relationships between the "Dog" and "Duck" objects through the round. It also keeps track of the number of ducks created, the current score and so on.
A "Round" object is created during the initiation of the game scene.

- The "**Dog**" class – holds the information and mechanics for the dog. It holds its "view container" (an object that is used by the rendering engine to draw the dog on the screen) and other relevant properties like size, position, texture object …. . It also contains a frame-by-frame description of the texture used for the different animations. This class has a method for every action performed by the dog (like walk, jump, catch a duck ….) and some helper methods if needed.

- The "**Duck**" class - holds the information and mechanics for the ducks. A "Duck" object is created when the dog jumps in the grass or when an other duck is killed. On its creation it receives a path(a list of predefined coordinates) which it will pass on the screen. The path is randomly chosen from a list with predefined paths. Otherwise the "Duck" class is similar with the "Dog" - it keeps track of its view object, animations description and holds a method for every action a duck can perform.

Both the "Dog" and the "Duck" class have an instance of a "EventEmitter" class that is used for notifications between them.

- The "**EventEmitter**" class is used for communication between the components of the game.
Every object that has instance of this class can emit custom events and other objects can subscribe to them. For example the "Dog" object can emit an event when it jumps into the grass and the "Round" object can subscribe to that event so it knows when to start creating the ducks.

The "**SoundManager**" class is used to manage the sounds in the game. The audio assets must be loaded in the main.js file like the graphics. The "SoundManager" should play some background music and different sounds triggered by events in game (shots, flapping of wings and so on).