**DEV report**

By creating this pygame “frequency” I’ve justed a lot of code reusing an ecapsulations.

For example in frequency.py:

The addPlayers Function takes an integer as parameter and will loop 4 times to create 4 players. This number will be added in the addPlayer function. The addPlayer function will give each player a color, the cost of the units and images. The loadUnitImages Function will load the images an

units and the costs of the units. Than append this in the self.player\_list of the addPlayer function.

COLOR\_NAMES = [ "red", "blue", "yellow", "green"]

**def addPlayers**(self):  
 self.player\_list = []  
 **for** i **in** range(4):  
 self.addPlayer(i)

#the 4 players are added in this function. each player receive his own color, the price of his units and the units images in the addPlayer()  
**def addPlayer**(self, playerNumber):  
 self.player\_list.append( Player(self.game.COLOR\_LIST[playerNumber],#the color of each player menu  
 self.game.UNIT\_COSTS[playerNumber],#the cost of the units of each player  
 self.loadUnitImages(playerNumber)) )#the units images in different colors  
#in this loop, all the units images are loaded here in different colors  
**def loadUnitImages**(self, playerNumber):  
 color = COLOR\_NAMES[playerNumber]  
 images = []  
 images.append(pygame.image.load(os.path.join( color + "soldier.png")))  
 images.append(pygame.image.load(os.path.join( color + "robot.png")))  
 images.append(pygame.image.load(os.path.join( color + "tank.png")))  
 images.append(pygame.image.load(os.path.join( color + "bar.png")))  
 images.append(pygame.image.load(os.path.join( color + "boat.png")))  
 images.append(pygame.image.load(os.path.join( color + "harbor.png")))  
 **return** images

All the button in board.py are a form of inheritance.

every time when a button is created, the subclasses of the superclass(Button) are created as well.

The sub classes(actionButton(Button), boardsquarebutton(Button) and unitButton(Button) have acces to the variables and functions of the Button class By calling the init method from the class Button, to have acces to those methods within the subclasses.

Encapsulation and loose coupeling are not used in this game.

**Stefan schmeltz.**