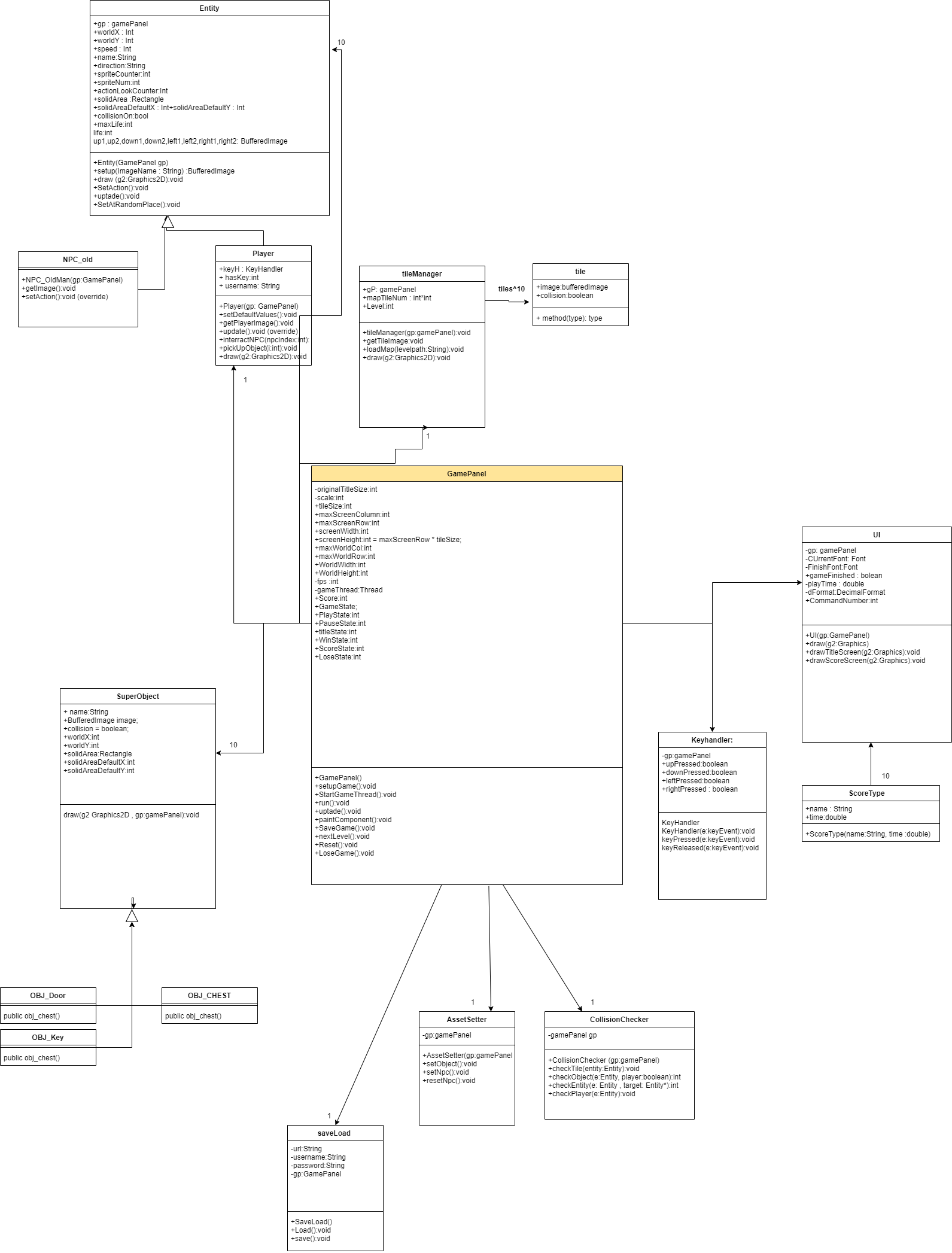
Nijat Sadigov huydfs  
task3

Labyrinth Create the Labyrinth game, where objective of the player is to escape from this labyrinth. The player starts at the bottom left corner of the labyrinth. He has to get to the top right corner of the labyrinth as fast he can, avoiding a meeting with the evil dragon. The player can move only in four directions: left, right, up or down. There are several escape paths in all labyrinths. The dragon starts off from a randomly chosen position, and moves randomly in the labyrinth so that it choose a direction and goes in that direction until it reaches a wall. Then it chooses randomly a different direction. If the dragon gets to a neighboring field of the player, then the player dies. Because it is dark in the labyrinth, the player can see only the neighboring fields at a distance of 3 units. Record the number of how many labyrinths did the player solve, and if he loses his life, then save this number together with his name into the database. Create a menu item, which displays a highscore table of the players for the 10 best scores. Also, create a menu item which restarts the game. Take care that the player and the dragon cannot start off on walls.



UML diragram

***GamePanel:***

***SetupGame()***

Setting default values of objects and npcs.

StartGameTHread()

Start game thread;

***RUN()***

Include update and repaint method till game is continuing , repeat is 60 times per second;

***Update()***

Update values of game objects

***PaintCOmponent()***

Draw elements of game

***SaveGame***

Save game level

***NExtLevel***

Download next level to screen

***Reset***

Reset game

***LoseGame()***

If player loses

***AssetSetter***

***setObject()***

Set default values of objects by its type.

***setNpc()***

Set default values of NPC.

***ResetNpc()***

reset default values of NPC.

***COLLISIONCHECKER***

***CheckTIle***

Check if Entity touch tile

**checkObject**

check if player interact with anth object

***checkENtity***

check if player touch npc

***checkPlayer***

checks if something touch with enemy

***SaveLoad***

**Save()**

Save game

**Load()**

**Load game**

***UI***

**Draw**

Depends on GameState it draws game screen, lose and win cases screen , in addition titlescreen, screen by calling functions

**drawtitleScreen**

**Draw title screen**

**draw ScoreSCreen**

Draw Score Screen

***TIleManager***

***GetTileImage***

get image of tile

***loadMap***

Load map depends on tileType

***draw***

draw tiles

***Entity***

***Draw***

Draw entity

***Setup***

set default values

***setAction***

Set action for npcs

***Uptade***

Uptade Entity values

***setAtRandomPlace***

Find random empty place to npc