CS-224 Object Oriented Programming and Design Methodologies

Assignment 04

Fall 2021

1 Guidelines

You need to submit this assignment on Nov 7 at 2359. Some important guidelines about the assignment are as following:

- You can do this assignment in a group of two students, and if you want you can also do it alone.
- You will submit your assignment to LMS (only one member of the group will submit).
- Clearly mention the group composition in submitted file name e.g. AhmadHassan_ah01345_BatoolAiman_ba03451.zip.
- You need to follow the best programming practices
- Submit assignment on time; late submissions will not be accepted.
- Some assignments will require you to submit multiple files. Always Zip and send them.
- It is better to submit incomplete assignment than none at all.
- It is better to submit the work that you have done yourself than what you have plagiarized.
- It is strongly advised that you start working on the assignment the day you get it. Assignments WILL take time.
- DO NOT send your assignment to your instructor, if you do, your assignment will get ZERO for not following clear instructions.
- You can be called in for Viva for any assignment that you submit

2 BattleField

A sample code is given in BattleField folder, if you run it you can see a bullet is created everytime you click on the screen, it is moving slightly towards right side. This example creates and display an object using SDL library. It uses `Bullet` class to create and draw an object. This class is inherited from `Unit` class that is fully implemented and takes care of all the drawing aspects, hence you don't need to change anything in it.

You are required to:

- Create a class `TankTurret`, this class will be displaying a tank turret, and can be implemented similar to `Bullet`.
- Create a class `TankBody`, this class will be displaying a tank body, and can be implemented similar to `Bullet`.
- Create a class `Tank`, this class makes a composition of TankTurret and TankBody objects, i.e. in each Tank there is a tankBody and a tankTurret object.
- When you click on the screen, a Tank object is created dynamically instead of a Bullet. Remember that the tank object should be created by using new operator, hence the list of tanks will also be storing pointers of tanks.
- When you press `F key`, it fires bullets from all the tanks displayed on the screen. It means, at pressing the F key, you have to create one bullet object corresponding to each Tank being currently displayed. The bullets will also be created dynamically, and would be stored in a list. Here the list type would be pointers to Bullet.
- Tank should make an animation to move its turret slightly towards left, when a bullet is fired, then it returns back to its position.
- Modify the Bullet class, so that it displays a vanishing animation when it reaches to right most corner of the screen. Look at the different bullet animations in the assets file.
- Remember to remove the objects from memory when they are not in use any more.

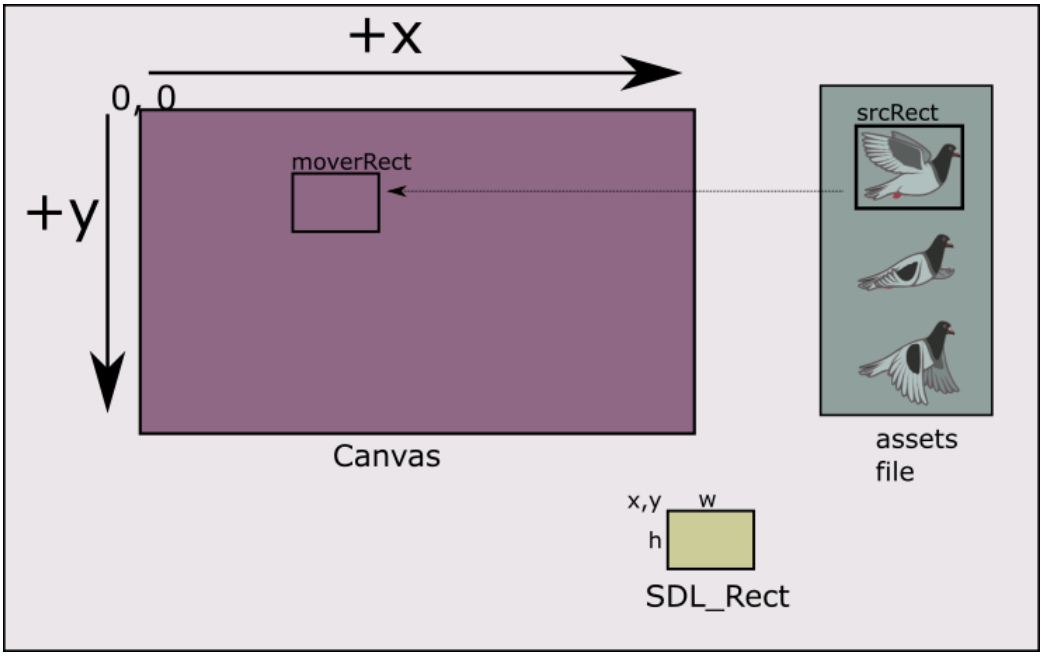

Figure 1: SDL Drawing Basics

2.1 SDL Drawing Basics

The basic drawing function in SDL is very simple, you need two SDL_Rect variables to draw a portion of image from assets file to the canvas. `SDL_Rect` is a simple structure containing `{x, y, w, h}` attributes. `(x, y)` is the top-left corner, and `w, h` are width and height of rectangle. You define a `srcRect` for desired object in assets file, and define a `moverRect` for this image to be drawn on desired location on canvas. Refer to Figure 1 for all this process. Finally you call
`SDL_RenderCopy(gRenderer, assets, &pigeonSrc, &pigeonMover);`
that displays this image to the canvas, voila!!!. Refer to `assets.png` file for all the required image assets.

You can draw as many objects in the `HUMania.cpp` $\Rightarrow$ `drawObjects()`, as you want. Since this function is called infinitely, you can change the `x, y` attributes of `moverRect` to move the objects on screen, and you can change the `srcRect` values to get a flying animation.

3 `std::list` Tutorial

Following is a basic example to work with vector. Complete reference for C++ vector is given here `https://en.cppreference.com/w/cpp/container/vector`

```
#include<iostream>
#include<list>

using namespace std;

class Distance{
   int feet, inches;
   public:
   Distance(int ft, int inch): feet(ft), inches(inch){}
   void show(){
      cout<<feet<<"'"<<inches<<"\""<<endl;
   }
};

int main(){
   list<Distance*> dst; // It's a vector that can store Distance
       type objects
   dst.push_back(new Distance(3, 4)); // create an object, and push
       it in vector
   dst.push_back(new Distance(5, 2));
   dst.push_back(new Distance(2, 7));
   dst.push_back(new Distance(7, 8));
   dst.push_back(new Distance(13, 1));

   for(int i=0;i<dst.size();i++)
      dst[i]->show(); // call show method of dst[i] object

   for(int i=0;i<dst.size();i++)
      delete dst[i]; // Let's delete all the objects

   // all the objects are deleted, but dst still holds the pointers
   dst.clear(); // It clears up all the pointers stored in dst
}

/////////////// Output: ////////////////////
3'4"
5'2"
```

```
2'7"
7'8"
13'1"
```

---

4 Some important points:

- Sample code is there for your benefit. If you are going to use it, understand how it works.
- You do not need to follow the code given exactly. You can make changes where you see fit provided that it makes sense.
- Make the class declarations in hpp files, and provide function implementations in cpp files. Don't use hpp files for implementation purposes.
- As a general rule, class's deta is private, and functions are public. Don't use getter/setter functions to manipulate data, rather think in object oriented directions and provide all the functionality in the class.
- Complete reference for C++ vector is given here `https://en.cppreference.com/w/cpp/container/vector`
- You need to define separate `*.hpp` and `*.cpp` files for all the classes.
- Exact x,y,w,h values for images in assets file can be found by `http://www.spritecow.com/`.
- A tutorial for file I/O is given `http://www.cplusplus.com/doc/tutorial/files/`.
- You should take `www.cplusplus.com` and `www.cppreference.com` as primary web source to search about C++
- You have to follow best OOP practices as discussed in lectures.

5 Rubric

Comments	The code was properly commented	1
Coding	The code followed best practices guideline	2
OOP Concepts	The code followed best OOP practices	3
Functionality	All the functionality is implemented as described above	4
Total		10

Table 1: Grading Rubric