Advanced Game- and Simulator Programming(STE6245)

Jiaxin Lin(140740)

February 26, 2016

Abstract

This is the report about the Simulation of collision.Using GMlib, class Sphere and class Psurf from UiT-Campus Narvik.The report main to description of GMlib,programming style,the important part of collision detection system and demo of project.

1 Introduction

This is a description of GMlib Library,the main class use in the project and the C++ programming style.

1.1 GMlib Library

GMlib is free software for use with geometric modelling with parametric in C++.In the project, Visualization of the balls and walls are using GMlib.The libary was make and update by UIT-Campus Narvik.

1.2 Main Class

In this project I creat class Testsphere,Controller inherit from PSphere, class MYPlanenine inherit form PSurf,class MP inherit from PPlane.The PSphere and PPlane both inherit from Psurf.

1.2.1 PSphere Class

This class to creat balls.In this class have some functions.

Table 1: Class PSphere main functions

Functions	**Means**
PSphere	This is the construction function to create the ball. And have overload the constructor.
getRadius	This function to get the sphere radius.
setRadius	This function to set the sphere radius.
eval	This function to set matrix.Using formula to createthe shape like the ball in 3D. (the function parameter member inherit from PSurf)

1.2.2 PPlane Class

This class to creat walls.The main function follow in the Table 2.

Table 2: Class PPlane main functions

Functions	**Means**
PPlane	This is the construction function to create the plane. Need one point and two vectors.
getNormal	This function to get the normal of the surfaces.
getU	This function to get the Vector U(vector u use to create wall).
getV	This function to get the Vector V(vector v use to create wall).
setU	This function to set the Vector U(vector u use to create wall).
setV	This function to set the Vector V(vector v use to create wall).
setP	This function to set the point of create the wall.

1.2.3 PSurf Class

This is the basic class to create the surface.This class inherit from Parametrics. The class PSurf have lots of functions which use to creat the surface you want in 3D. Just look the inheritance diagram follow the figure.

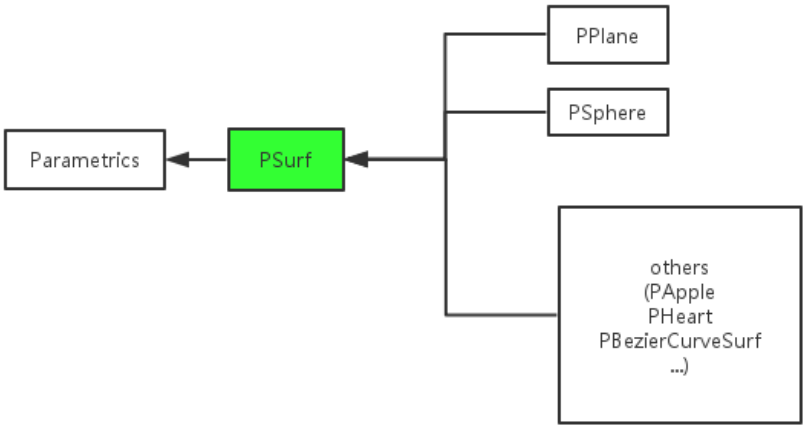

Figure 1: Inheritance diagram for Psurf

1.3 Fundamental Types

Table 3: Main Types

Fundamental Types	**Means**
GMlib::Vector	In the project this is the important type of variable. The vector have length and direction.It is a template class.
GMlib::Point	This type to create a point.It is a template like the vector.
GMlib::Array	This type to create a array.It is a container to store the number of balls,walls and collision.

1.4 Programming Style

In this project follow some programming style.

- Using Allman style.if, for, while, namespace braces, write in a new line.
- Naming class using uppercase.
- Private member variables using underline.
- Naming the type bool add is(such as isEmpty).
- Using tab to indent.

2 QT Framework

Qt is used mainly for developing application software with graphical user interfaces. Qt supports many compilers, including the GCC C++ compiler and the Visual Studio suite. Qt 5 was officially released on 19 December 2012. This new version marked a major change in the platform, with hardware-accelerated graphics, QML and JavaScript playing a major role. The traditional C++-only QWidgets continued to be supported, but did not benefit from the performance improvements available through the new architecture.[1] Qt 5 brings significant improvements to the speed and ease of developing user interfaces.[2]In this project using Qt Version 5.4.

3 Theory

This part introduce some theory whic use in this project.First give some concept and then description of class.

3.1 Concept-Compute Step

This concepet use to calculate the ball next step when the ball move.

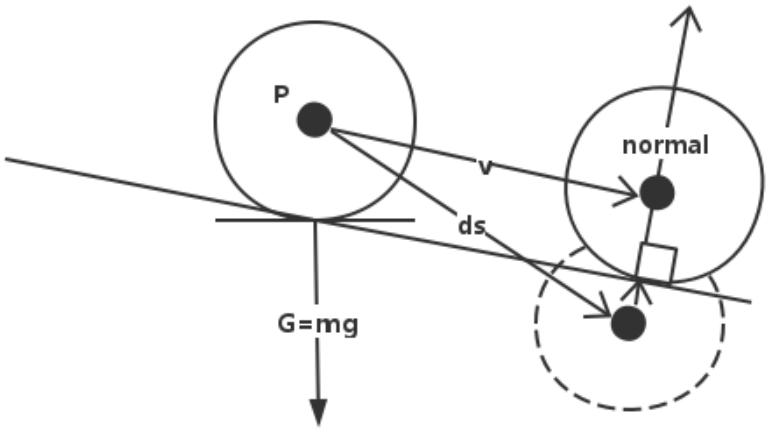

Figure 2: The ball move

According the project

$$\_ds = dt * \_velocity + (0.5 * dt * dt) * g \tag{1}$$

$$\_p = P * ds \tag{2}$$

$$\_n = m[0][0] + r * n \tag{3}$$

$$\_ds = n - P \tag{4}$$

3.2 Concept-Ball Wall Collision

This is the main part of collision.First need to know where is the wall. Creat wall need parameter follow the table.

Table 4: My caption

Variable	**Means**
Vector1	one direction of create the wall
Vector2	other direction of create the wall
Point	Decide the position to create the wall

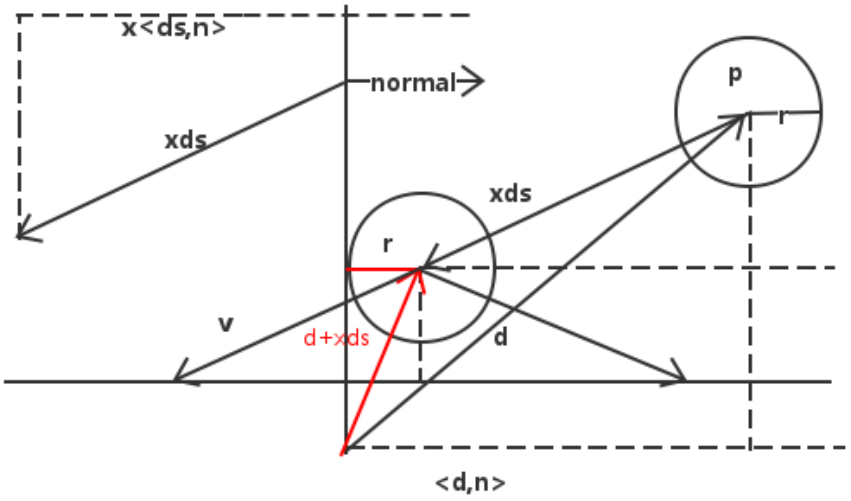

Figure 3: Ball Wall Collision

From this figure can see clearly how to detect the ball and wall collision and caculate the related data.

$$< d + xds, n >= r \qquad from \quad the \quad red \quad line \quad . \tag{5}$$

Solve this formular to caculate x.

$$< d, n > + x < ds, n >= r \tag{6}$$

$$x = \frac{r - < d, n >}{< ds, n >} \tag{7}$$

Check if the value of ds*n less than 0 and the variable x between 0 and 1 the collision is detected.

$$V = V - 2 * (n * V) * n \tag{8}$$

Set new velocity after collision.

3.3 Concept-Ball Ball Collision

This part of ball with ball collision.The ball constructor have some parameter follow the table.

Table 5: ball constructor parameter

Parameter	**Means**
mass	The mass of the ball.
velocity	The velocity of the ball
radius	From the GMlib::PSphere,the radius of the ball
plane	The ball move on the plane.Decide what kind of the plane.

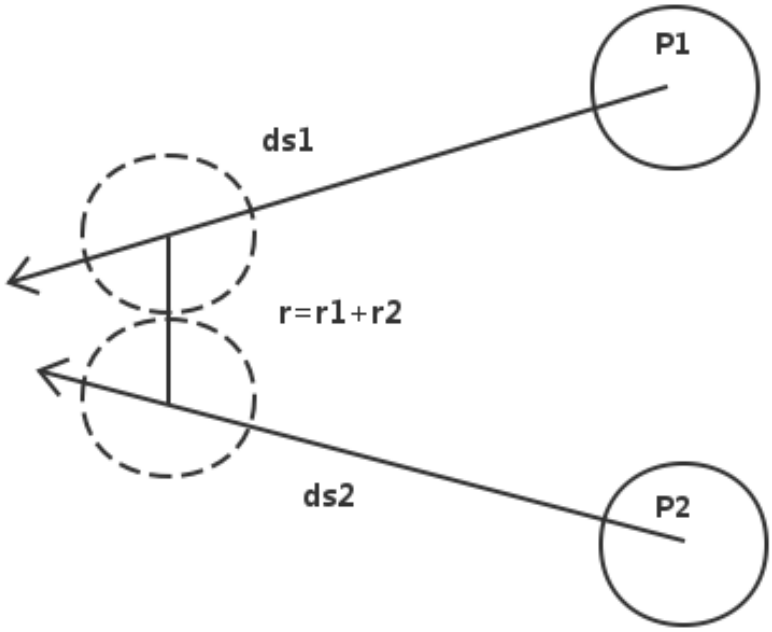

Figure 4: Ball Ball Collision

From the figure can make an equation.

$$P_1 + xds1 - (p_2 + xds2) = p_1 - p_2 + x(ds1 - ds2) = dp + xds \tag{9}$$

$$< dp + xds, dp + xds >= r^2 \tag{10}$$

Simplification

$$< ds, ds > x^2 + 2 < ds, dp > x + < dp, dp > -r^2 = 0 \tag{11}$$

To caculate x,x should between 0 and 1.

$a =< ds, ds >$
$b = 2 < ds, dp > x$
$c =< dp, dp > -r^2$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{12}$$

The new velocity after collision follow the below figure.

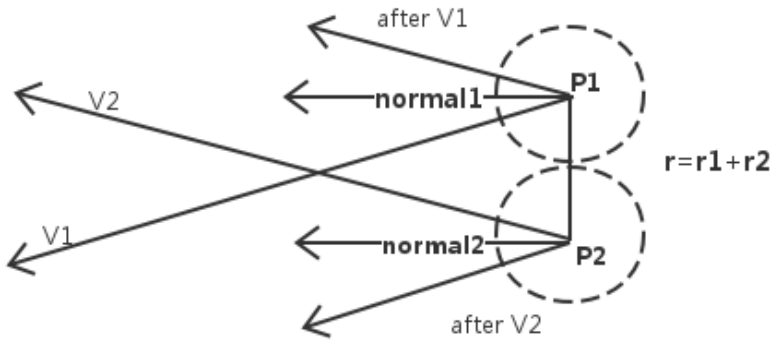

Figure 5: set new velocity

From the figure can get some formula.

$$d = p2 - p1 \tag{13}$$

The ball 1

$$v1 = (V1 * d) * d \qquad v1n = V1 - v1 \tag{14}$$

The ball 2

$$v2 = (V2 * d) * d \qquad v2n = V2 - v2 \tag{15}$$

The new velocity with mass

$$V1after = \frac{m1 - m2}{m1 + m2} * v1 + \frac{2 * m1}{m1 + m2} * v2 \tag{16}$$

$$V2after = \frac{m2 - m1}{m1 + m2} * v1 + \frac{2 * m2}{m1 + m2} * v2 \tag{17}$$

The finnal velocity of the ball

$$V1new = V1after + v1n \tag{18}$$

$$V2new = V2after + v2n \tag{19}$$

4 Structure of the resulating application

4.1 Structure of this project

The project base on teacher's framework. In this project add five parts,the main part is the Controller. The planenine means using nine point to create floor.

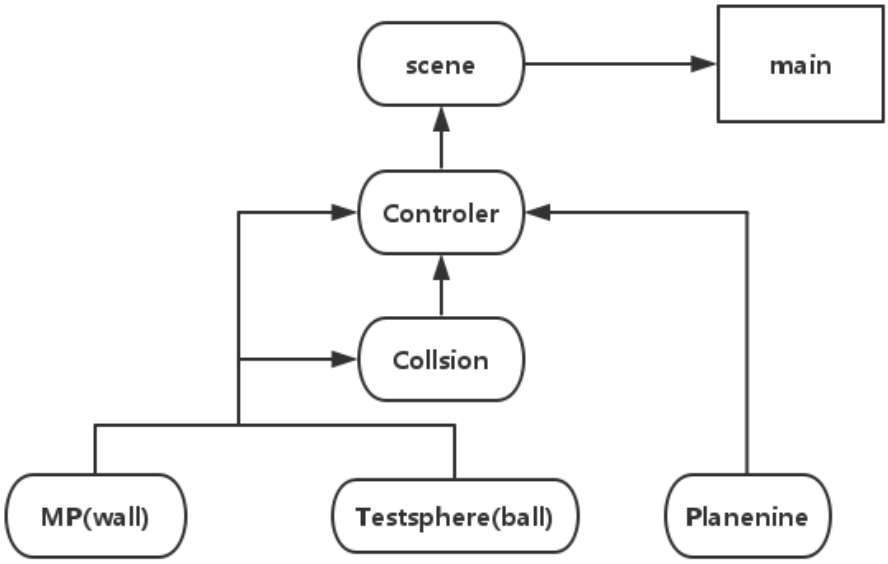

Figure 6: The structure of the project

4.2 Description of each class

Class list.Here are the classes.

Table 6: Class List

Classes	**Brief descriptions**
Testsphere	Inherit the PSphere.To create a ball.
MP	Inherit the PPlane.In this project to create walls.
MYPlanenine	Inherit the PSurf.In this project to create the floor.
Collison	This class to define collision.Ball with ball and Ball with wall
Controller	This class to find when have collision.The main part of this project.

4.2.1 Class Testsphere

This class to define the ball.The properties of the ball included

Table 7: Class Testsphere

Properties	**Descriptions**
GMlib::Vector _velocity	The velocity of the ball.
int _mass	The mass of the ball.
GMlib::Vector _ds	The distance of the ball move.
MYPlanenine* _plane	The floor of the ball move.Make connect with the floor.
double _x	The time to store.

Table 8: Functions of the Testsphere

Function	**Means**
TestSphere	Constructor,To set initial value of the ball.
setVelocity	Set new velocity of the ball.
getVelocity	Get velocity of the ball.
setMass	Set new mass of the ball.
getMass	Get mass of the ball.
setX	Set run time of the ball.
getX	Get the store of time.
moveUp	Control the ball to move up.
moveDown	Control the ball to move down.
moveLeft	Control the ball to move left.
moveRight	Control the ball to move right.
computeStep	Compute the ball move to next step.
getSurfnormal	To get the normal of the surface.

4.2.2 Class MP

This class inherit teacher's class PPlane.This class using to create walls. Using one point and two vector.

4.2.3 Class MYPlanenine

This class inherit teacher's class PSurf.This class using to create floors. Using nine points set a matrix.

4.2.4 Class Collision

Table 9: Class Collision

Properties	**Descriptions**
Testsphere *_sphere[2]	The array of Testsphere class.
MP *_wall;	Define the wall.
bool _isSw;	Define weather ball and wall collision or not.
double _x	The time to store.
operator ==	Operator overloading to compare to collision.
bool operator <	Operator overloading to compare to collision.

4.2.5 Class Controller

Table 10: Class Controller

Properties	**Descriptions**
MYPlanenine* _plane	Define the floor surfaces.
GMlib::Array _sphereArray	This is the array to contain balls.
GMlib::Array _wallArray	This is the array to contain walls.
GMlib::Array _collisionArray	This is the array to contain collisions.
void addSphere	Add ball into the ball array.
void addWall	Add wall into the wall array.
void findcollisionSW	This is a function to find ball and wall collision.
void findcollisionSS	This is a function to find ball and ball collision.
void collisionSW	This is a function to set new velocity after ball and wall collision.
void collisionSS	This is a function to set new velocity after ball and wall collision.

QT Keyboard control

This part trying to control ball by keyboard.In the GMlibWrapper::keyPressed to set keys. Using up,down,left,right to control ball velocity.

demo

This part to show some interface in the project.

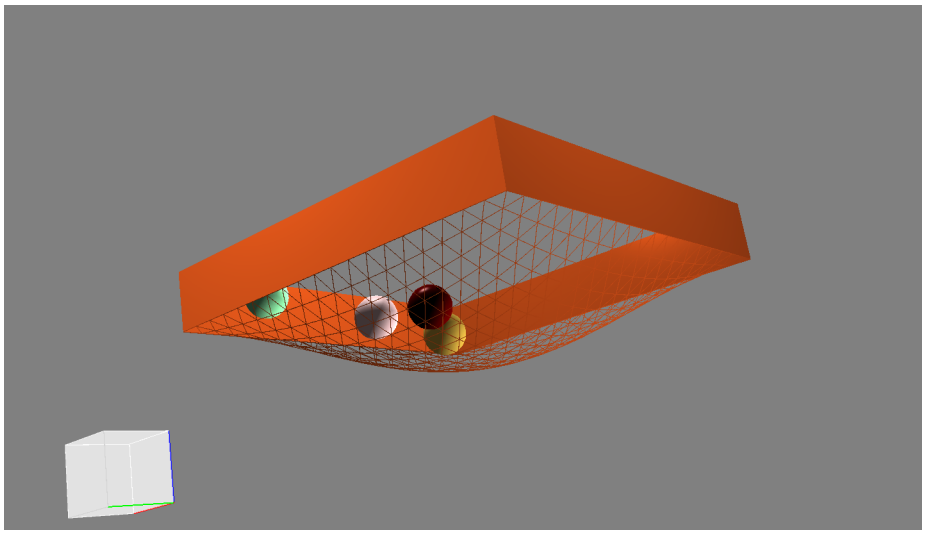

Figure 7: demo 1

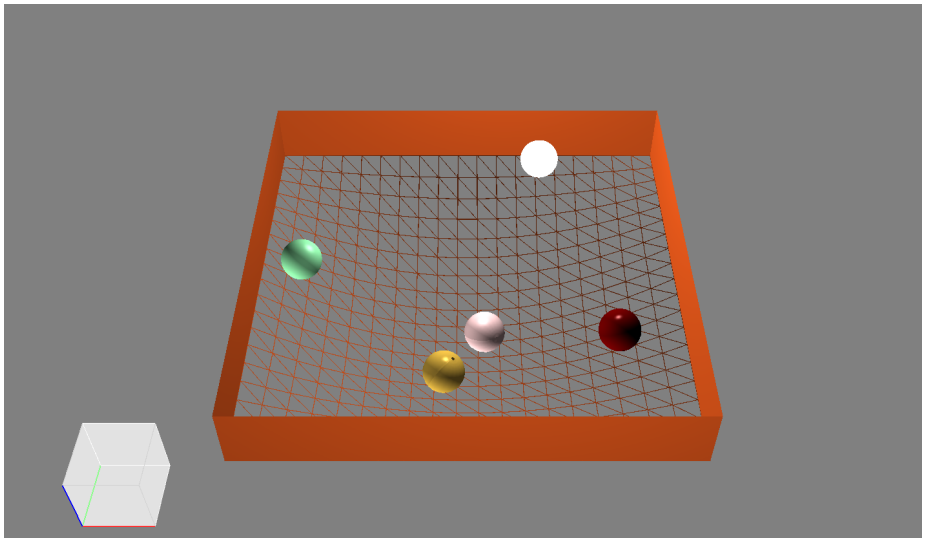

Figure 8: demo 2

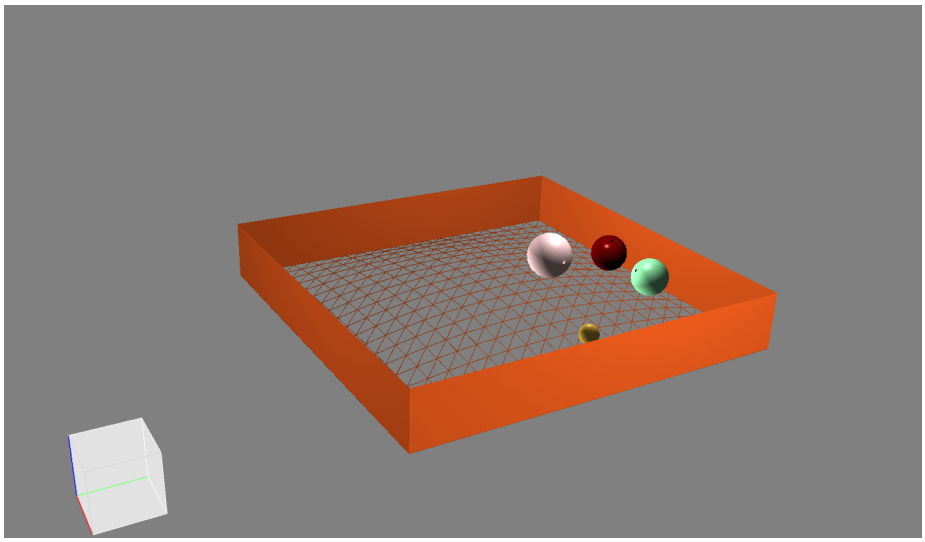

Figure 9: demo 2

Conclusion

In conclusion,this project is based on the lectures in the course: geometric modelling. Througe this project understand the vector better.Practise better than theory.Moreover,with use the GMlib libary and see the code form the teacher. Final need to say thank you to my classmates.They help me a lot.

References

[1] Qt5-feedback (Mailing list), *Concern about removal of QWidget classes*, 7 October 2011.

[2] Knoll, Lars, *Thoughts about Qt 5*, Retrieved 9 May 2011.