Vincent Zhang | Final Project | June 12, 2019

Disc Game UML Diagram

DiscShooter
+ frame:DiscShooter + canvas:Canvas + toolbar:Toolbar - WINDOWWIDTH:final int - WINDOWHEIGHT:final int
+ DiscShooter() - run():void + void main(args:String[])

CalcScore
+ cols:String[] + rows:String[][] - records:ArrayList<PlayRecord> - data:ArrayList<ArrayList<String>>
- formatData():void + read():void + write(name:String):void + updateScoreBoard(scoreBoard:ScoreBoard, score:int):void

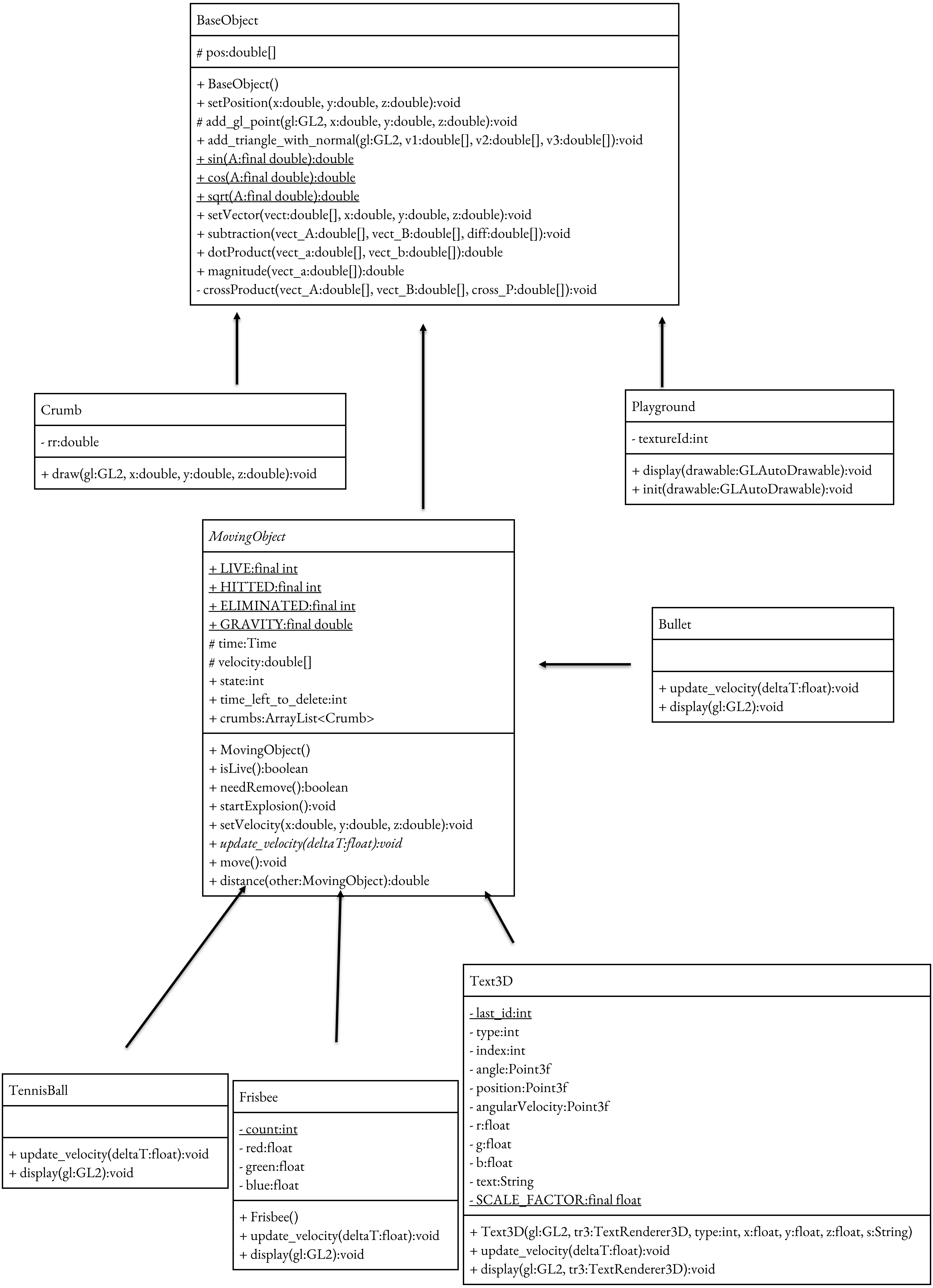
PlayRecord
+ name:String + time:String + numShots:int + numFrisbees:int + numTennisBalls:int + frisbeesHit:int + ballsHit:int + score:int
+ compareTo(o:PlayRecord):int

Toolbar
- stats:CalcScore - firstTime:boolean + numFrisbees:int + numTennisBalls:int + numShots:int + numFrisbeesHit:int + numBallsHit:int + score:int + accuracy:double - settingsBtn:JButton - scoreBtn:JButton - exitBtn:JButton - nDifficulty:JTextField - nShots:JTextField - nFrisbees:JTextField - nTennisBalls:JTextField - nFrisbeesHit:JTextField - nBallsHit:JTextField - nScore:JTextField - nAccuracy:JTextField
+ trySaveScore():void + updateScoreBoard(scoreBoard:ScoreBoard):void + updateStatistics(numFrisbees:int, numTennisBalls:int, numShots:int, numFrisbeesHit:int, numBallsHit:int, difficulty:int):void - buttonImage(int num, String name):void + Toolbar()

Canvas

- glu:GLU
- rotation:float
- timeLastTennisBall:long
- lastTimeFrisbeeAdded:long
- canvas_width:double
- canvas_height:double
+ BallSpeed:double
+ frisbeeSpeed:double
- frisbees:ArrayList<Frisbee>
- bullets:ArrayList<Bullet>
- balls:ArrayList<TennisBall>
- playground:Playground
- scoreBoard:ScoreBoard
- readyToPlay:ReadyToPlay
- textflow:TextFlow
- numShots:int
- numFrisbees:int
- numTennisBalls:int
- frisbeesHit:int
- ballsHit:int
- scene:int
- distance_for_hit:double
- difficulty:int
- scoreBoard_need_to_be_updated:boolean
- dragging:boolean
- lastDragX:int
- lastDragY:int
- motionIncr:float
- xRot:float
- yRot:float
- tr3:TextRenderer3D
- LIGHT_POSITION:final float[]
- LIGHT_AMBIENT:final float[]
- LIGHT_DIFFUSE:final float[]
- MATERIAL_SPEC:final float[]
- ZERO_VEC4:final float[]
- gc:GLCanvas

+ keyPressed(e:KeyEvent):void
+ keyTyped(e:KeyEvent):void
+ keyReleased(e:KeyEvent):void
+ mouseClicked(e:MouseEvent):void
+ mousePressed(e:MouseEvent):void
+ mouseReleased(e:MouseEvent):void
+ mouseEntered(e:MouseEvent):void
+ mouseExited(e:MouseEvent):void
+ display(drawable:GLAutoDrawable):void
+ dispose(drawable:GLAutoDrawable):void
+ init(drawable:GLAutoDrawable):void
+ reshape(drawable:GLAutoDrawable, x:int, y:int, width:int, height:int):void
+ prepare_light(gl:GL2):void
+ addTennisBall():void
+ addFrisbee(gl:GL2):void
+ shoot(vx:double, vy:double, vz:double):void
+ gameStart():void
+ gameStop():void
+ getRecord(r:PlayRecord):void
+ Canvas(frame:JFrame)



BaseObject

# pos:double[]

+ BaseObject()
+ setPosition(x:double, y:double, z:double):void
# add_gl_point(gl:GL2, x:double, y:double, z:double):void
+ add_triangle_with_normal(gl:GL2, v1:double[], v2:double[], v3:double[]):void
+ sin(A:final double):double
+ cos(A:final double):double
+ sqrt(A:final double):double
+ setVector(vect:double[], x:double, y:double, z:double):void
+ subtraction(vect_A:double[], vect_B:double[], diff:double[]):void
+ dotProduct(vect_a:double[], vect_b:double[]):double
+ magnitude(vect_a:double[]):double
- crossProduct(vect_A:double[], vect_B:double[], cross_P:double[]):void

Crumb

- rr:double

+ draw(gl:GL2, x:double, y:double, z:double):void

Playground

- textureId:int

+ display(drawable:GLAutoDrawable):void
+ init(drawable:GLAutoDrawable):void

MovingObject

+ LIVE:final int
+ HITTED:final int
+ ELIMINATED:final int
+ GRAVITY:final double
# time:Time
# velocity:double[]
+ state:int
+ time_left_to_delete:int
+ crumbs:ArrayList<Crumb>

+ MovingObject()
+ isLive():boolean
+ needRemove():boolean
+ startExplosion():void
+ setVelocity(x:double, y:double, z:double):void
+ *update_velocity(deltaT:float):void*
+ move():void
+ distance(other:MovingObject):double

Bullet

+ update_velocity(deltaT:float):void
+ display(gl:GL2):void

TennisBall

+ update_velocity(deltaT:float):void
+ display(gl:GL2):void

Frisbee

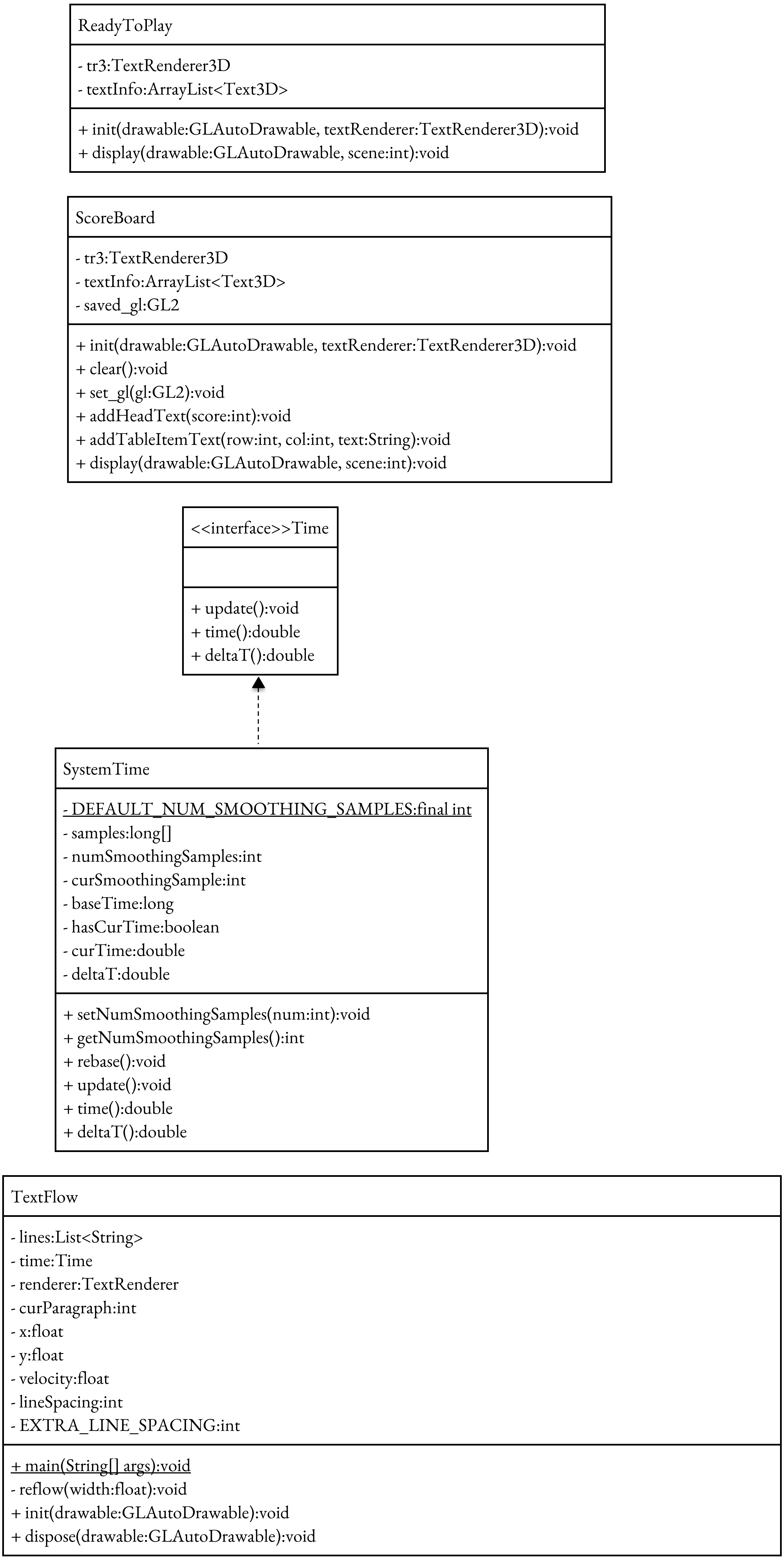
- count:int
- red:float
- green:float
- blue:float

+ Frisbee()
+ update_velocity(deltaT:float):void
+ display(gl:GL2):void

Text3D

- last_id:int
- type:int
- index:int
- angle:Point3f
- position:Point3f
- angularVelocity:Point3f
- r:float
- g:float
- b:float
- text:String
- SCALE_FACTOR:final float

+ Text3D(gl:GL2, tr3:TextRenderer3D, type:int, x:float, y:float, z:float, s:String)
+ update_velocity(deltaT:float):void
+ display(gl:GL2, tr3:TextRenderer3D):void



ReadyToPlay

- tr3:TextRenderer3D
- textInfo:ArrayList<Text3D>

+ init(drawable:GLAutoDrawable, textRenderer:TextRenderer3D):void
+ display(drawable:GLAutoDrawable, scene:int):void

ScoreBoard

- tr3:TextRenderer3D
- textInfo:ArrayList<Text3D>
- saved_gl:GL2

+ init(drawable:GLAutoDrawable, textRenderer:TextRenderer3D):void
+ clear():void
+ set_gl(gl:GL2):void
+ addHeadText(score:int):void
+ addTableItemText(row:int, col:int, text:String):void
+ display(drawable:GLAutoDrawable, scene:int):void

<<interface>>Time

+ update():void
+ time():double
+ deltaT():double

SystemTime

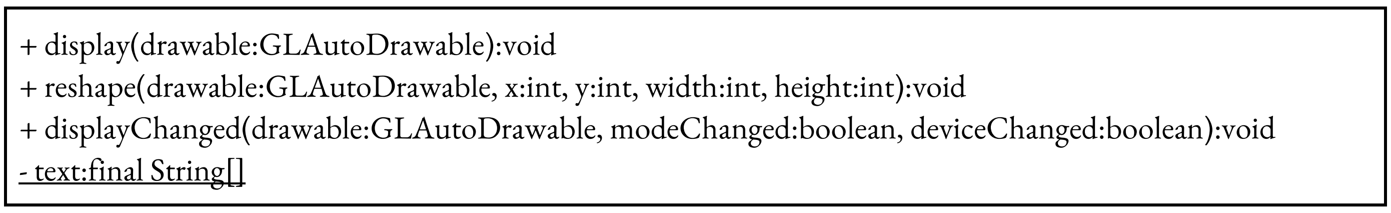
- DEFAULT_NUM_SMOOTHING_SAMPLES:final int
- samples:long[]
- numSmoothingSamples:int
- curSmoothingSample:int
- baseTime:long
- hasCurTime:boolean
- curTime:double
- deltaT:double

+ setNumSmoothingSamples(num:int):void
+ getNumSmoothingSamples():int
+ rebase():void
+ update():void
+ time():double
+ deltaT():double

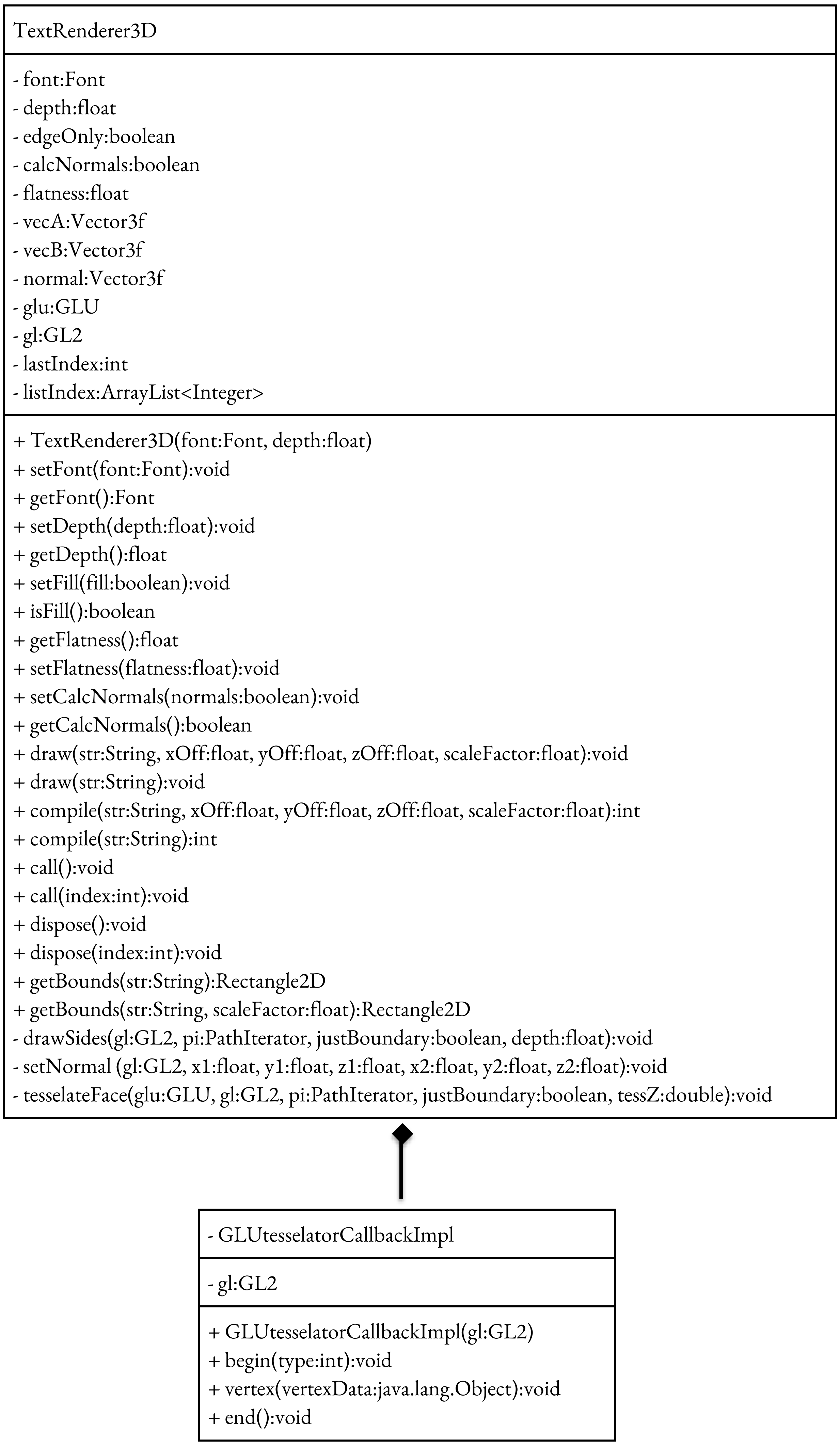
TextFlow

- lines:List<String>
- time:Time
- renderer:TextRenderer
- curParagraph:int
- x:float
- y:float
- velocity:float
- lineSpacing:int
- EXTRA_LINE_SPACING:int

+ main(String[] args):void
- reflow(width:float):void
+ init(drawable:GLAutoDrawable):void
+ dispose(drawable:GLAutoDrawable):void



+ display(drawable:GLAutoDrawable):void
+ reshape(drawable:GLAutoDrawable, x:int, y:int, width:int, height:int):void
+ displayChanged(drawable:GLAutoDrawable, modeChanged:boolean, deviceChanged:boolean):void
- text:final String[]



TextRenderer3D

- font:Font
- depth:float
- edgeOnly:boolean
- calcNormals:boolean
- flatness:float
- vecA:Vector3f
- vecB:Vector3f
- normal:Vector3f
- glu:GLU
- gl:GL2
- lastIndex:int
- listIndex:ArrayList<Integer>

+ TextRenderer3D(font:Font, depth:float)
+ setFont(font:Font):void
+ getFont():Font
+ setDepth(depth:float):void
+ getDepth():float
+ setFill(fill:boolean):void
+ isFill():boolean
+ getFlatness():float
+ setFlatness(flatness:float):void
+ setCalcNormals(normals:boolean):void
+ getCalcNormals():boolean
+ draw(str:String, xOff:float, yOff:float, zOff:float, scaleFactor:float):void
+ draw(str:String):void
+ compile(str:String, xOff:float, yOff:float, zOff:float, scaleFactor:float):int
+ compile(str:String):int
+ call():void
+ call(index:int):void
+ dispose():void
+ dispose(index:int):void
+ getBounds(str:String):Rectangle2D
+ getBounds(str:String, scaleFactor:float):Rectangle2D
- drawSides(gl:GL2, pi:PathIterator, justBoundary:boolean, depth:float):void
- setNormal (gl:GL2, x1:float, y1:float, z1:float, x2:float, y2:float, z2:float):void
- tesselateFace(glu:GLU, gl:GL2, pi:PathIterator, justBoundary:boolean, tessZ:double):void

- GLUtesselatorCallbackImpl

- gl:GL2

+ GLUtesselatorCallbackImpl(gl:GL2)
+ begin(type:int):void
+ vertex(vertexData:java.lang.Object):void
+ end():void