GameObjects

Ball (Chino encargado ?)

Description: La bola controla el juego. hay que saber

vars:
bool ballgameover
int status; // waiting, playing, game over

Functions:
setup()
update()

bool isInsideScenario(ofBox);
bool setGameOverBall(); // callback from GameOverCollision

SimpleObject **(Kuba)**

Vars:

- Id generic

- size(ofVec3f)
- worldPosition (ofMatrix4x4)
- Centroide (ofVec3f) // for what?
- bvisible (boolean)
- colisionCounter (int)

(Physics Properties)
- enum {Static/dynamic/Kinematic}
- float Damping
- float Friction

(Events Status):
- bool bAnimation
- bool bCollision

objDisplay mygraphicobject

Functions:
- setup(ofxBulletWorldRigid &world, type)

- update() // check distance to ball & call

updateAnimations, etc..
- draw()

SpringObject (Chino encargado ?)

Description: Objeto que multiplica la fuerza con que la bola rebota

vars:
- shapeType (enum ..)
- shape (bullet shape object)

VectorColision (ofVec3f)
EnergyCollision (Float)
Time to pull: (int);
objDisplay animation
functions:
checkDistance2ball(int idBall);

Lever (Chino encargado ?)

Description: Palancas
Vars:
- shapeType (enum ..)
- shape (bullet shape object)
velocity (float)
angle (float)

Functions:
push()
setup()
update()
checkDistance2ball(int idBall);

ContainerObject (Chino encargado ?)

Description: La bola para y rebota después de un tiempo (o un evento?)

vars:
- shapeType (enum ..)
- shape (bullet shape object)
int timebeforeaction;
Hammer myHammer;

functions:
update() // ?
checkDistance2ball(int idBall);

void cachBall(int idBall, ofVec3f stopPos) // capturamos la bola en una pos

Hammer (Chino encargado ?)

Description: Objeto que se usa para empujar la bola (en el estado initial o para los objectos containers)

Vars:
Public
energy (float)
Shape ( )
ofVec3f startmov
ofVec3f endmov
int duration
Private:

Functions:
setup(ofVec3f startPoint, ofVec3f EndPoint, enum Shape, ofVec3f Size, worldPos)
start(Timer t) //
checkDistance2ball(int idBall);

Managers

PinballChinoManager (Chino encargado ?)

Description: La clase mas basica que actualiza el mundo, el estado del juego, y puede crear/tener mas de un escenario.

Vars:
Scenario myScenario1;

ofxBulletWorldRigid world;
ofCamera camera;

GameStatus mygamestatus;

Functions:
setup()
update()
draw()

Ranking (Chino encargado ?)

Description:
Save / get users scores from XML file

Vars:
int bestPlayer;
int bestScore;
int bestNumMissionsCompleted;

Functions:
saveXmlScore(gameNumber, numMissionsCompleted, score)

loadXmlRanking();

GameStatus (singletoon class)
// reference about singleton design pattern "Chapter 6. Implementing Singletons" from Modern C++ Design by Andrei Alexanrescu

Description:
Manages the game score, players, levels, todo

Vars:
int idPlayer;
int score;
int lives;
int currentMission;
enum gamestatus {waintingStart, playing, gameOver};
ranking myRanking;
<int> missionsCompleted;

Functions:
saveRanking();

Scenario

Scenario (Chino encargado ?)

Description: Responsable por inicializar y controlar los elementos de un escenario del juego.
Vars:
ofxBulletWorldRigid * world;

ofBoxBullet wall_bottom, wall_top, wall_left, wall_right;

vector <Ball> myball;

vector<SpringObject> mySimpleSprings;
vector<ContainerObject> mySimpleContainers;

Hammer mypullball;

Lever leftlever;
Lever rightlever;

Functions:
setup(ofxBulletWorldRigid &world)
update();
draw();
loadScenario(xmlfile)

outObjectScenario (Chino encargado ?)

Description: Responsable de informar que una bola se ha salido del escenrio

Vars:
ofBox box;
event outCallback;

Functions:
setup(ofMatrix4x4 worldPosiotion, ofVec3f size)
loadEvent()

Media

objDisplay (Chino encargado ?)

Description: Se encarga de los métodos y atributos que controlan la exhibición del objeto

Vars:
State (normal / collision / animation)
textureFilename (string);
videoFilename (string);
bAnimated (bool);

private:
worldposition (ofmatrix4x4)
shapeType (enum)

setup(worldposition, shape)
update(gameStatus, objectStatus)
eventCollision( )

video_objDisplay (Chino encargado ?)

ofxVideoPlayer player

setup(filename, pos)
update()
draw()

img_objectDisplay (Chino encargado ?)

ofImage image

setup(filename, pos)
update()
draw()

sound_objectDisplay (Chino encargado ?)

ofSound sound

setup(filename, pos)
update()

Missions

simpleMission

Description:
Manages the goals and statuses of a mission within a scenario (time, points to achieve, tasks, etc.)

Vars:
int idMision;
int MissionTimeLimit;

Functions:

virtual setup(int type)
virtual update()

bool isMissionCompleted()

Mission 1

Description: La primera mision sera completar un numero de puntos en un timepo concreto.

Vars:

int MissionTargetPoints;
int MissionTimeLimit;
ofXmlSettings xmdatamision; // num points , tiempo a resolver
ofxTimer timer;

Functions:

loadMision(ofxXmlSettings mySettings) //Actualiza los valores de points, time, goals, etc.
finishMission(); //callback from Timer
update();

ScenarioEditor

?