CSC413-01 Term Project

Wingman and Tankgame

By: Elbert Dang

Outline

- Wingman
  - Game Logic
  - Challenges
  - Solutions
  - Demo
  - Class Diagram
- Tankgame
  - Class Diagram
  - Demo

Wingman Game Logic

- 3 Different powerups: Weapon, Health, and Lives
  - Powerups add to score when their function is not needed
  - Max Weapon level is 4, Health is 100, Lives is 10
- Initial slow rate of fire and low damage
  - Weapon powerups increase damage, rate of fire, damage, and number of bullets fired.
  - Weapon level increases by 1 per powerup, decreases by 1 per hit, resets to 0 when killed.
- Colliding with enemy subtracts your health with theirs
  - Possible to not kill enemy when your health is too low

Wingman Game Logic (cont.)

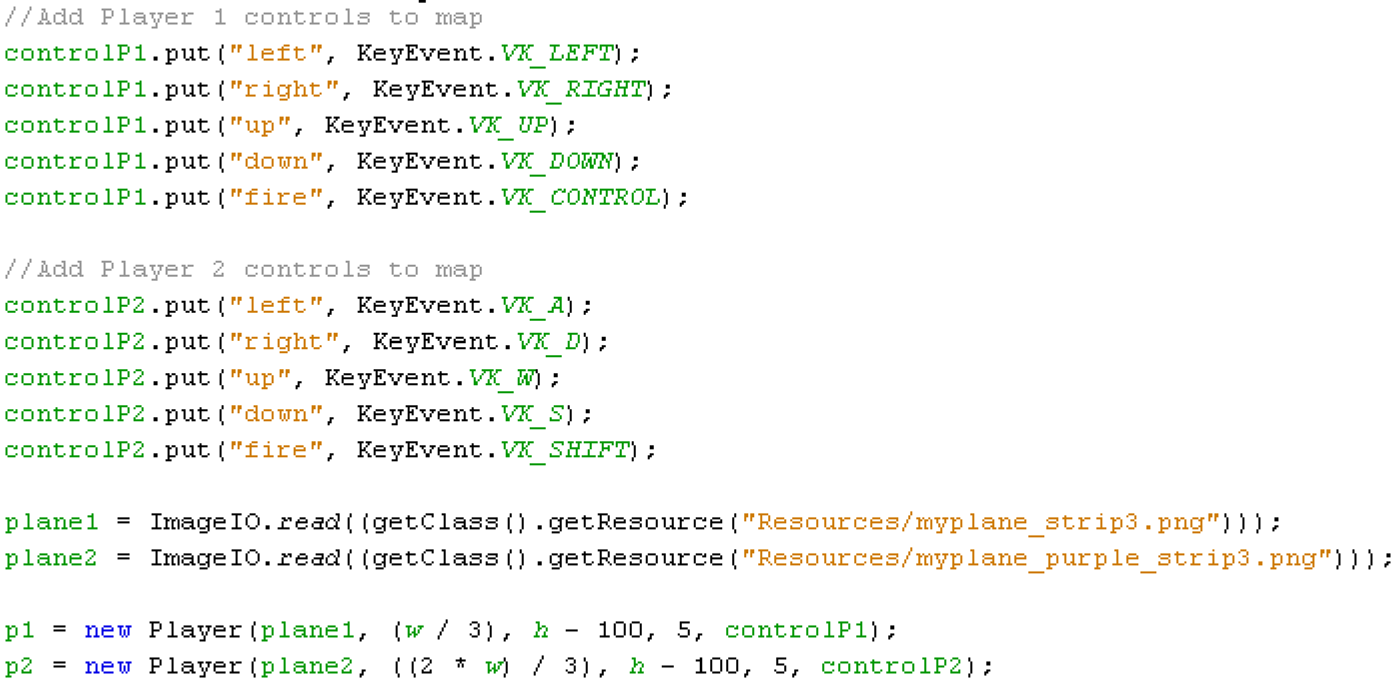
- Objects are removed when they go 250 pixels off screen or when they are “done” (such as explosions)
  - Decreases memory usage for better performance
- Applet opens in the middle of screen
  - Terminates when window is closed

Wingman Challenges

- 2 players, one class
- Smooth controls and movement
- Detect different collisions
- Timeline of enemies

Solution to 2 players, 1 class

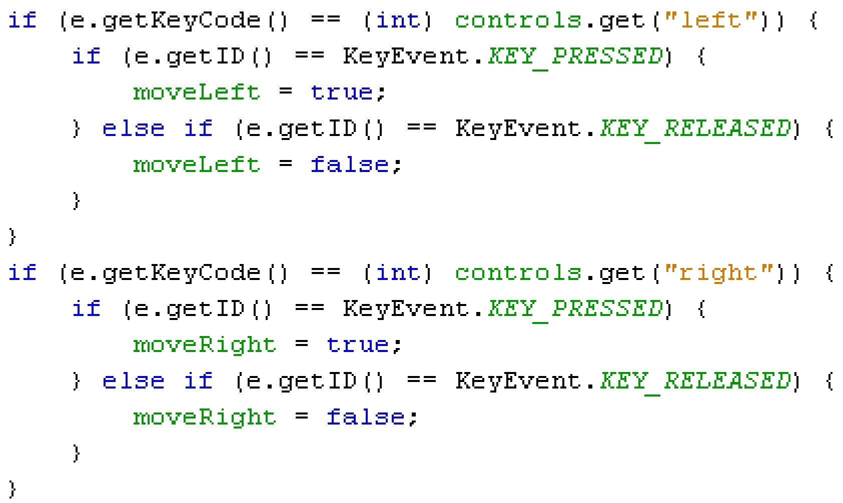
Send Hashmap of controls



```
//Add Player 1 controls to map
controlP1.put("left", KeyEvent.VK_LEFT);
controlP1.put("right", KeyEvent.VK_RIGHT);
controlP1.put("up", KeyEvent.VK_UP);
controlP1.put("down", KeyEvent.VK_DOWN);
controlP1.put("fire", KeyEvent.VK_CONTROL);

//Add Player 2 controls to map
controlP2.put("left", KeyEvent.VK_A);
controlP2.put("right", KeyEvent.VK_D);
controlP2.put("up", KeyEvent.VK_W);
controlP2.put("down", KeyEvent.VK_S);
controlP2.put("fire", KeyEvent.VK_SHIFT);

plane1 = ImageIO.read((getClass().getResource("Resources/myplane_strip3.png")));
plane2 = ImageIO.read((getClass().getResource("Resources/myplane_purple_strip3.png")));

p1 = new Player(plane1, (w / 3), h - 100, 5, controlP1);
p2 = new Player(plane2, ((2 * w) / 3), h - 100, 5, controlP2);
```

Solution for smooth controls

Boolean values:
Default is false
True when keyPressed
False when keyReleased



```
if (e.getKeyCode() == (int) controls.get("left")) {
    if (e.getID() == KeyEvent.KEY_PRESSED) {
        moveLeft = true;
    } else if (e.getID() == KeyEvent.KEY_RELEASED) {
        moveLeft = false;
    }
}
if (e.getKeyCode() == (int) controls.get("right")) {
    if (e.getID() == KeyEvent.KEY_PRESSED) {
        moveRight = true;
    } else if (e.getID() == KeyEvent.KEY_RELEASED) {
        moveRight = false;
    }
}
```

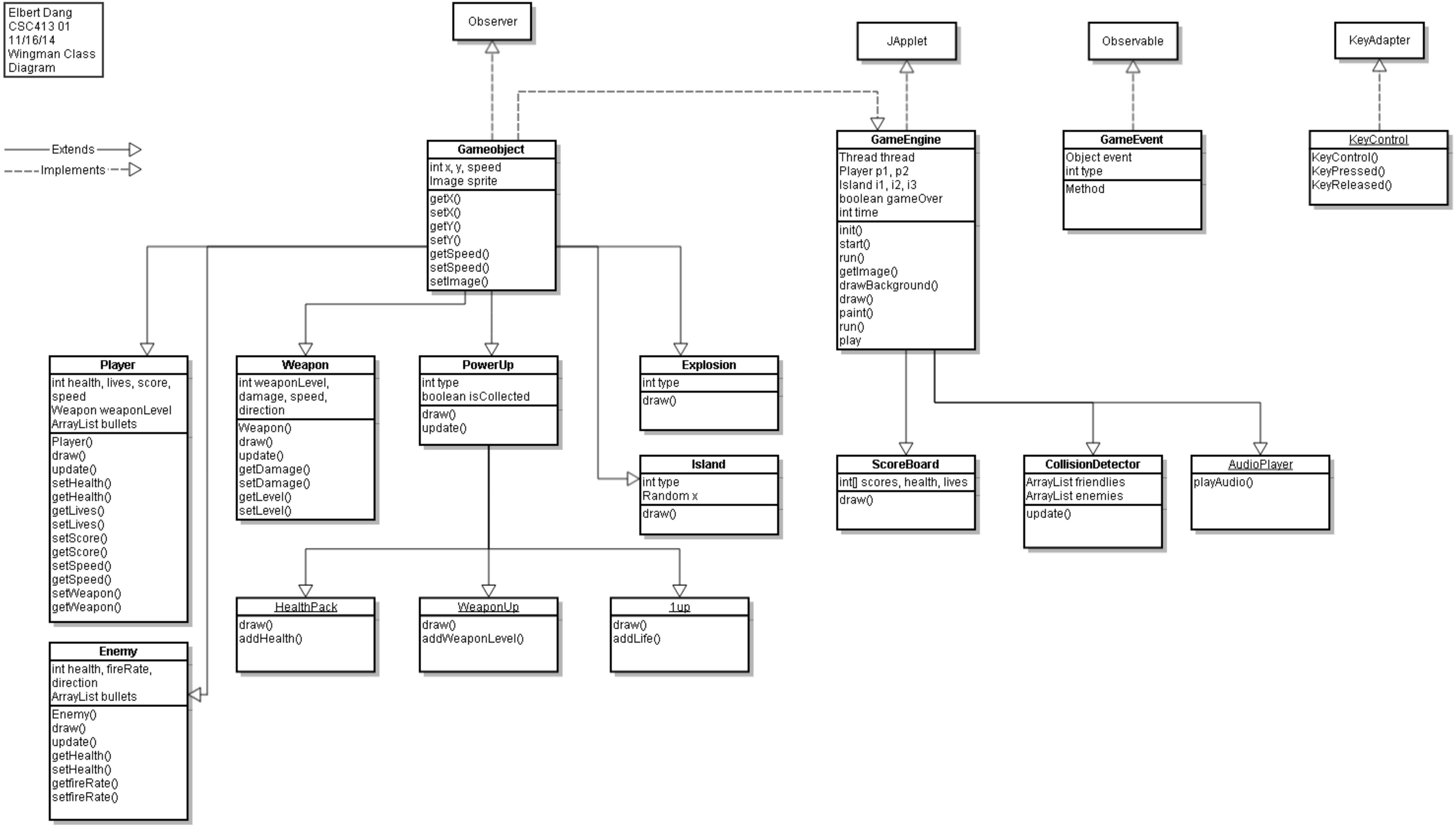
Allows for moving in multiple directions, firing when moving, etc.

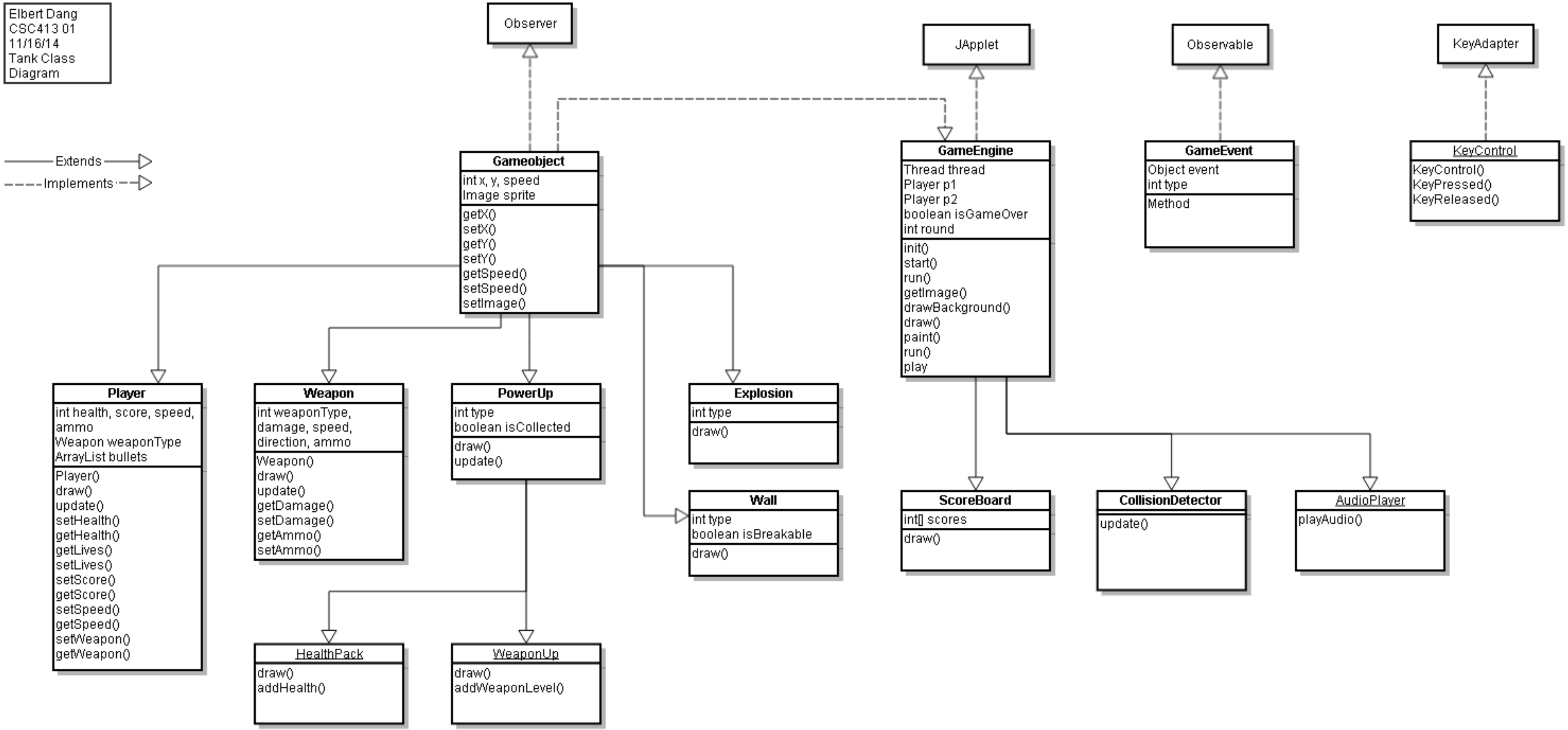
Solution to detecting different collisions

- Have each object used by game extend GameObject.
- GameObject stores bounding rectangle of image, therefore its object.
- Have CollisionDetector go through ArrayList of objects that collide with each other to check what rectangles are intersecting.

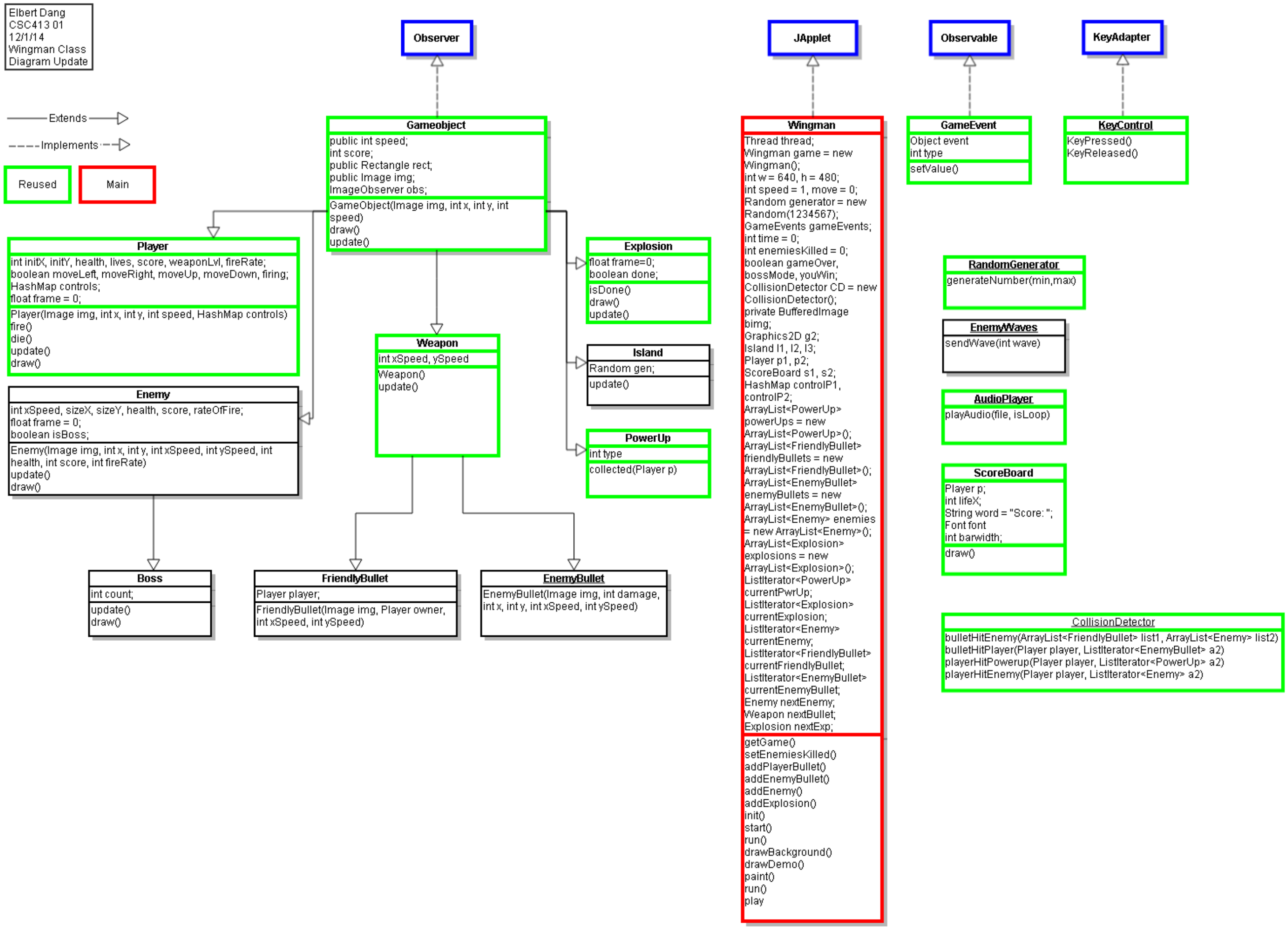
Solution to timeline

- Keep track of time that game is run and enemies killed
- For every 150 frames or so, an enemy wave is sent out
  - RandomGenerator selects from 0 to 6 different wave formations and 4 enemy types.
  - Every 5 enemies killed, a powerup is spawned
- Once 50 enemies are killed, Boss appears
  - Boss has access to new enemy type and 2 new wave formations




Elbert Dang
CSC413 01
11/16/14
Tank Class
Diagram
Extends
Implements
Observer
JApplet
Observable
KeyAdapter
Gameobject
int x, y, speed
Image sprite
getX()
setX()
getY()
setY()
getSpeed()
setSpeed()
setImage()
GameEngine
Thread thread
Player p1
Player p2
boolean isGameOver
int round
init()
start()
run()
getImage()
drawBackground()
draw()
paint()
run()
play
GameEvent
Object event
int type
Method
KeyControl
KeyControl()
KeyPressed()
KeyReleased()
Player
int health, score, speed,
ammo
Weapon weaponType
ArrayList bullets
Player()
draw()
update()
setHealth()
getHealth()
getLives()
setLives()
setScore()
getScore()
setSpeed()
getSpeed()
setWeapon()
getWeapon()
Weapon
int weaponType,
damage, speed,
direction, ammo
Weapon()
draw()
update()
getDamage()
setDamage()
getAmmo()
setAmmo()
PowerUp
int type
boolean isCollected
draw()
update()
Explosion
int type
draw()
Wall
int type
boolean isBreakable
draw()
ScoreBoard
int[] scores
draw()
CollisionDetector
update()
AudioPlayer
playAudio()
HealthPack
draw()
addHealth()
WeaponUp
draw()
addWeaponLevel()

Elbert Dang
CSC413 01
12/1/14
Wingman Class
Diagram Update



Observer

JApplet

Observable

KeyAdapter

Gameobject
public int speed;
int score;
public Rectangle rect;
public Image img;
ImageObserver obs;
GameObject(Image img, int x, int y, int speed)
draw()
update()

Player
int initX, initY, health, lives, score, weaponLvl, fireRate;
boolean moveLeft, moveRight, moveUp, moveDown, firing;
HashMap controls;
float frame = 0;
Player(Image img, int x, int y, int speed, HashMap controls)
fire()
die()
update()
draw()

Enemy
int xSpeed, sizeX, sizeY, health, score, rateOfFire;
float frame = 0;
boolean isBoss;
Enemy(Image img, int x, int y, int xSpeed, int ySpeed, int health, int score, int fireRate)
update()
draw()

Boss
int count;
update()
draw()

Weapon
int xSpeed, ySpeed
Weapon()
update()

FriendlyBullet
Player player;
FriendlyBullet(Image img, Player owner, int xSpeed, int ySpeed)

EnemyBullet
EnemyBullet(Image img, int damage, int x, int y, int xSpeed, int ySpeed)

Explosion
float frame=0;
boolean done;
isDone()
draw()
update()

Island
Random gen;
update()

PowerUp
int type
collected(Player p)

Wingman
Thread thread;
Wingman game = new Wingman();
int w = 640, h = 480;
int speed = 1, move = 0;
Random generator = new Random(1234567);
GameEvents gameEvents;
int time = 0;
int enemiesKilled = 0;
boolean gameOver, bossMode, youWin;
CollisionDetector CD = new CollisionDetector();
private BufferedImage bimg;
Graphics2D g2;
Island I1, I2, I3;
Player p1, p2;
ScoreBoard s1, s2;
HashMap controlP1, controlP2;
ArrayList<PowerUp> powerUps = new ArrayList<PowerUp>();
ArrayList<FriendlyBullet> friendlyBullets = new ArrayList<FriendlyBullet>();
ArrayList<EnemyBullet> enemyBullets = new ArrayList<EnemyBullet>();
ArrayList<Enemy> enemies = new ArrayList<Enemy>();
ArrayList<Explosion> explosions = new ArrayList<Explosion>();
ListIterator<PowerUp> currentPwrUp;
ListIterator<Explosion> currentExplosion;
ListIterator<Enemy> currentEnemy;
ListIterator<FriendlyBullet> currentFriendlyBullet;
ListIterator<EnemyBullet> currentEnemyBullet;
Enemy nextEnemy;
Weapon nextBullet;
Explosion nextExp;
getGame()
setEnemiesKilled()
addPlayerBullet()
addEnemyBullet()
addEnemy()
addExplosion()
init()
start()
run()
drawBackground()
drawDemo()
paint()
run()
play

GameEvent
Object event
int type
setValue()

KeyControl
KeyPressed()
KeyReleased()

RandomGenerator
generateNumber(min,max)

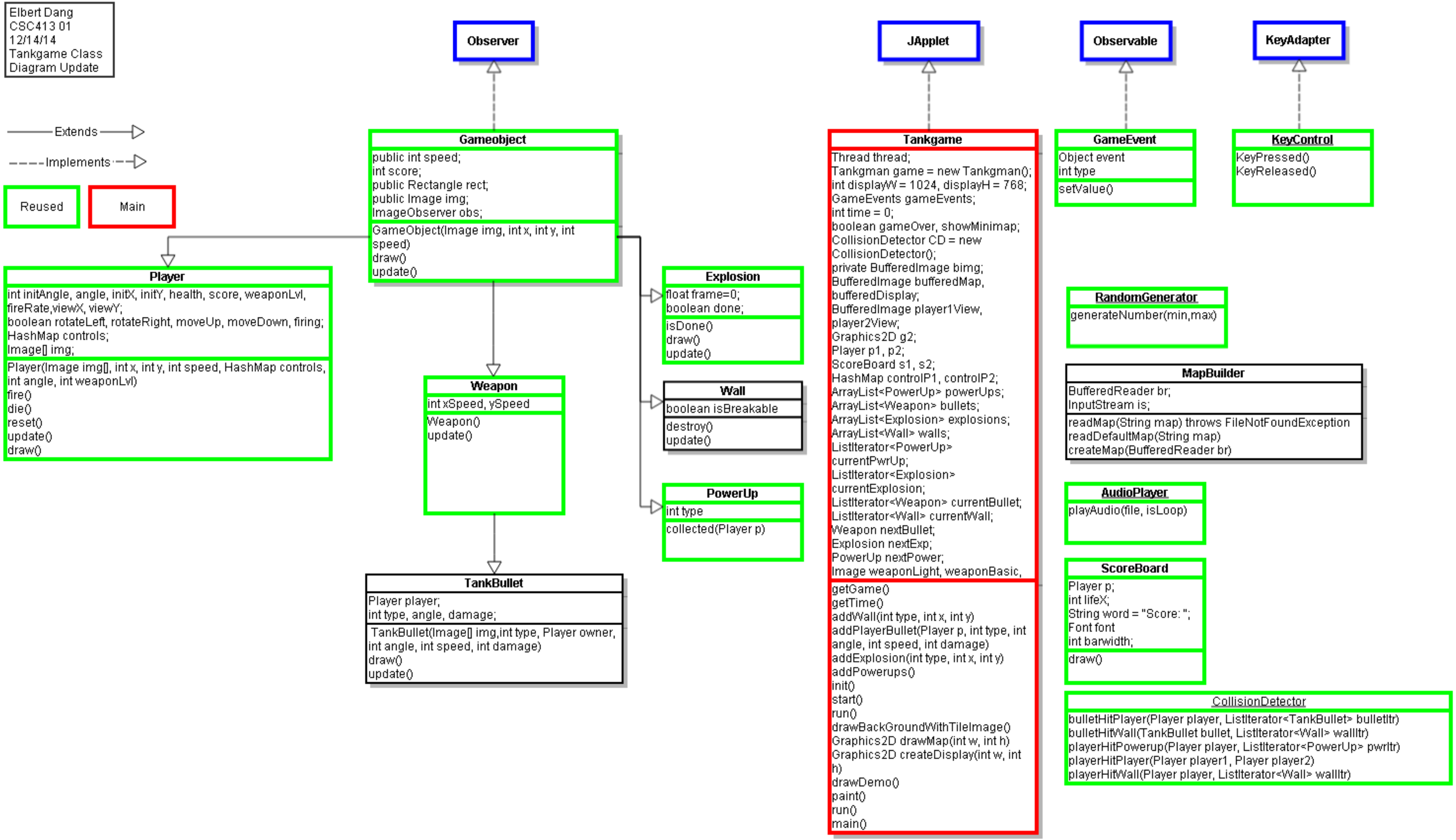
EnemyWaves
sendWave(int wave)

AudioPlayer
playAudio(file, isLoop)

ScoreBoard
Player p;
int lifeX;
String word = "Score: ";
Font font
int barwidth;
draw()

CollisionDetector
bulletHitEnemy(ArrayList<FriendlyBullet> list1, ArrayList<Enemy> list2)
bulletHitPlayer(Player player, ListIterator<EnemyBullet> a2)
playerHitPowerup(Player player, ListIterator<PowerUp> a2)
playerHitEnemy(Player player, ListIterator<Enemy> a2)

Elbert Dang
CSC413 01
12/14/14
Tankgame Class
Diagram Update



Gameobject
public int speed;
int score;
public Rectangle rect;
public Image img;
ImageObserver obs;
GameObject(Image img, int x, int y, int speed)
draw()
update()

Player
int initAngle, angle, initX, initY, health, score, weaponLvl, fireRate,viewX, viewY;
boolean rotateLeft, rotateRight, moveUp, moveDown, firing;
HashMap controls;
Image[] img;
Player(Image img[], int x, int y, int speed, HashMap controls, int angle, int weaponLvl)
fire()
die()
reset()
update()
draw()

Weapon
int xSpeed, ySpeed
Weapon()
update()

TankBullet
Player player;
int type, angle, damage;
TankBullet(Image[] img,int type, Player owner, int angle, int speed, int damage)
draw()
update()

Explosion
float frame=0;
boolean done;
isDone()
draw()
update()

Wall
boolean isBreakable
destroy()
update()

PowerUp
int type
collected(Player p)

Tankgame
Thread thread;
Tankgman game = new Tankgman();
int displayW = 1024, displayH = 768;
GameEvents gameEvents;
int time = 0;
boolean gameOver, showMinimap;
CollisionDetector CD = new CollisionDetector();
private BufferedImage bimg;
BufferedImage bufferedMap, bufferedDisplay;
BufferedImage player1View, player2View;
Graphics2D g2;
Player p1, p2;
ScoreBoard s1, s2;
HashMap controlP1, controlP2;
ArrayList<PowerUp> powerUps;
ArrayList<Weapon> bullets;
ArrayList<Explosion> explosions;
ArrayList<Wall> walls;
ListIterator<PowerUp> currentPwrUp;
ListIterator<Explosion> currentExplosion;
ListIterator<Weapon> currentBullet;
ListIterator<Wall> currentWall;
Weapon nextBullet;
Explosion nextExp;
PowerUp nextPower;
Image weaponLight, weaponBasic,
getGame()
getTime()
addWall(int type, int x, int y)
addPlayerBullet(Player p, int type, int angle, int speed, int damage)
addExplosion(int type, int x, int y)
addPowerups()
init()
start()
run()
drawBackGroundWithTileImage()
Graphics2D drawMap(int w, int h)
Graphics2D createDisplay(int w, int h)
drawDemo()
paint()
run()
main()

GameEvent
Object event
int type
setValue()

KeyControl
KeyPressed()
KeyReleased()

RandomGenerator
generateNumber(min,max)

MapBuilder
BufferedReader br;
InputStream is;
readMap(String map) throws FileNotFoundException
readDefaultMap(String map)
createMap(BufferedReader br)

AudioPlayer
playAudio(file, isLoop)

ScoreBoard
Player p;
int lifeX;
String word = "Score: ";
Font font
int barwidth;
draw()

CollisionDetector
bulletHitPlayer(Player player, ListIterator<TankBullet> bulletItr)
bulletHitWall(TankBullet bullet, ListIterator<Wall> wallItr)
playerHitPowerup(Player player, ListIterator<PowerUp> pwrItr)
playerHitPlayer(Player player1, Player player2)
playerHitWall(Player player, ListIterator<Wall> wallItr)