Christopher Ray, Nathan Vahrenberg

**Text**

Public:

Text();

void update();

Private:

string message;

Professor Scott

CSE 20212

23 March 2014

Project Diagram

Solid lines indicate an inheritance hierarchy. Dashed lines indicate composition.

**Terrain**

Public:

Terrain();

void update();

**PowerUp**

Public:

PowerUp();

void update();

int getPower();

Private:

int power;

**Zombie**

Public:

Zombie();

void update();

void move();

void attack();

void isDead();

Private:

int points;

**Board**

Public:

Board();

void display();

void update();

void Clean();

Private:

int start;

int game;

int pause;

int over;

Player player;

deque<Zombie> zombies;

deque<Text> textboxes;

deque<Button> buttons;

deque<Counter> counters;

deque<PowerUp> PowerUps;

deque<Terrain> boardTerrain;

**Player**

Public:

Player();

void update();

void move();

void attack();

void hitSomething();

void isDead();

**Item**

Public:

Iten();

void display();

virtual void update() = 0;

Protected:

int x;

int y;

**Counter**

Public:

Counter();

void update();

Private:

int count;

string message;

**Button**

Public:

Button();

void update();

Private:

string message;

int isUp;

int isDown;

**Unit**

Public:

Unit();

void display();

virtual void update() = 0;

virtual void move() = 0;

virtual void attack() = 0;

virtual void isDead() = 0;

Protected:

int x;

int y;

int health;

int speed;

int power;

**Box**

Public:

Box();

void display();

virtual void update() = 0;

Protected:

int height;

int width;

int x;

int y;