

technical support manual

SulAxya lama

kAYLA harrison

Ayça Küçükdurmaz

Space Game

CS 240 Team #3

2016 Edition

Table of Contents

[Game Loop 2](#_Toc400051163)

Game Object ………………………………………………………………………………….......6

SDL Functions...............................................................................................................................10

Game Loop

[Checkbounds 3](#_Toc400051158)

[Run 4](#_Toc400051159)

[GameObjects 5](#_Toc400051160)

Checkbounds

**Prototype:**

void CheckBounds (GameObject\* object);

**Parameters:**

GameObject\* object

**Description:**

Checkbounds function is used for checking bounds to prevent player object from going off screen.

Run

**Prototype:**

void run();

**Parameters:**

None

**Description:**

Run function executes the game. It consist of lot of SDL functions including initiating the screen to placing the object on the screen, movement controls of the game. SDL functions used are described in following pages.

GameObjects

**Prototype:**

vector<GameObject\*> gameObjects

**Parameters:**

None

**Description:**

A vector is used to store all the object images for the game. The image are in bitmap format.

Game Object

[Load Image 7](#_Toc400051158)

[Move 8](#_Toc400051159)

Load Image

**Prototype:**

GameObject(std::string imageFilename)

{

**Parameters:**

String imagefilename

**Description:**

The function is used for loading all the bmp object images.

Move

**Prototype:**

Move(int newX, int newY)/{

**Parameters:**

Int newX;

Int newY;

**Description:**

The function acts as setter for both x and y. It handles moving objects.

SDL Functions

The program was based on SDL. Following code are implemeneted on SDL.

SDL\_Window\* window = 0;

* To create a window

SDL\_Surface\* screenSurface = 0;

* To create a surface to apply images

SDL\_Init(SDL\_INIT\_EVERYTHING); //

* initialize SDL

SDL\_Event e;

* Get and handle SDL events (input, errors, etc)

switch( e.key.keysym.sym )

* Keyboard events

SDL\_QUIT

* to close the window

GameObject\* currentObj = gameObjects.at(i);

* draw a rectangle around the image.

SDL\_Delay(1000.0 / (float)fps)

* for delaying

SDL\_Surface\* ImageLoader::loadImage(std::string filename)

* takes filename as a parameter and load image onto the screen