SPACE WARRIOR #1



TECHNICAL MANUAL

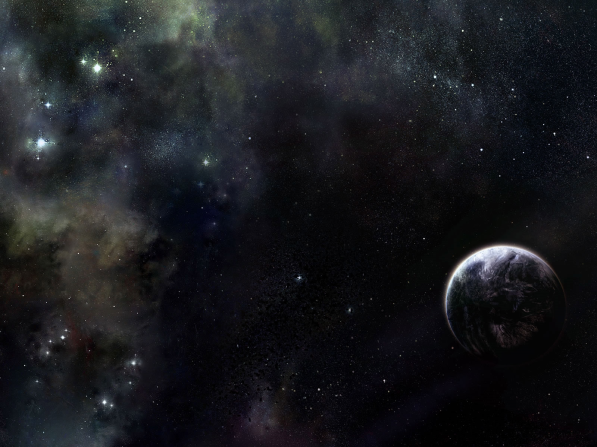
# PART A

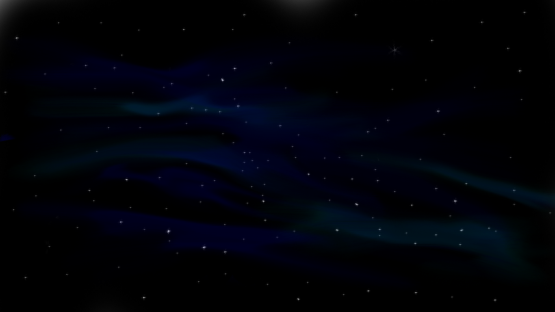
* **TYPE OF FSE**

2D game

* **TYPE OF USER INTERFACE**
* 2D interface

 Game play

 second Game Play in another position

 Level 2 game Play

 miniMap and menu

**Global Variables**

* Bool isExit -It gets the exit button rectangle and checks if it is clicked
* bool isStart – it gets if start is pressed or not.
* Texture2D start1 – Stores an image that will display start in Black
* Texture2D start2 – Stores an image that will display start in Blue.
* Texture2D exit1 - Stores an image that will display exit in Black.
* exit2 -Stores an image that will display start in Blue
* Vector2 startPos – Stores start Position
* Vector2 exitPos – Stores the position of the Texture2d exit.
* Texture2D mainBackGround – Store background image.
* Texture2D welcomeBackGround- Stores welcomed background image.
* Texture2D main,back1,firstBackGround,title,gameOver;
* Texture2D mainSprite – Stores main sprite image that will be user controlled.
* Texture2D backLevel1, backLevel2;
* public int shootEnemies - It is the number of enemies that have been shot.
* Texture2D life1 - life image - stores life image
* Texture2D mainlife1,mainlife2,mainlife3,mainlife4,mainlife5 - Stores image for life
* public float lifeMain = 10 –stores the perntage of the life of the main sprite
* float rotation – rotational movement of the main sprite
* bool isScreen = false –used to get iif the keyboard is pressed before the game starts;
* bool isDraw = false, isDraw2 = false, isDraw3 = false;
* bool state = false;
* SpriteFont font;
* int level = 1;
* public Vector2 mainPosition = new Vector2(300,100) – coordinates of the main sprite
* Vector2 mainPosition2 -position of the second main sprite in the virtual world
* Vector2 mainPosition3 - position of the third main sprite in the virtual world
* public Vector2 allPosition
* Vector2 origin ;
* public Vector2 speed = new Vector2(1, 1), speed2 = new Vector2(1, 1), speed3 = new Vector2(1, 1);
* const float tangetialVelocity = 5f;
* float friction = 0.1f;
* SoundEffect shootingSong;
* Song playingMusic;
* MEMORY BUFFERS

|  |  |  |
| --- | --- | --- |
| PICTURES | SIZE | TOTAL MEMORY |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\sprites\S2.bmp | 35.7KB | 35.7KB  \* 16 =  571.2 KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\sprites\spaceship.png | 5.19KB | 5.19KB \* 3 = 15.57KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\sprites\A6.bmp | 10.5KB | 10.5KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\sprites\alien.png | 19.8KB | 19.8KB \* 2 = 39.6KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\sprites\bullet2.png | 2KB | 2KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\flame.jpg | 47.2KB | 47.2KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\gameOver1.png | 541KB | 541KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\Untitled.png | 21.7KB | 21.7KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\life\life1.png | 4KB | 4KB \* 5 = 20KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\life\mainLife.png | 3KB | 3KB \* 5 = 15KB |
| C:\Users\'Siya\Desktop\DemoFinal\SpaceBrain\SpaceBrainContent\images\camera\miniMap.png | 2.3KB | 2.3KB \* |
| star1.bmp | 1.03KB | 1.03KB |
| Entrance.mp3 | 3.02MB | 3.02MB |
| Explosion | 13.2KB | 13.2KB \* 33 = 435.6KB |
| btn1.bmp | 55.1KB | 55.1KB\* 2 = 102.4KB |
| btn4.bmp | 55.1KB | 55.1KB \* 2 = 102.4KB |
| b3.jpg | 57.6KB | 57.6KB |
| TOTAL MEMORY BUFFERS |  | 1338.27KB |

* VIRTUAL WORLD
* 2D virtual world, with 1900 \* 1900 bounds
* MAP TYPE
* Mini map available to track units
* Its a 345 \* 225 pixels
* Mini map displays position of each unit and selected one’s
  + VIRTUAL WORLD DEFINITION
    - 100% life for each unit
    - Find and kill the enemy ,in order to finish the game
    - All units are in one interface, select one per each selection to have control over it
  + VIRTUAL WORLD SPECIFICS
* Mouse state - to make a functional mini map
* Texture2D – for all pictures
* Vector2 – for all position of textures

PART B:

* INPUT, PROCESS AND OUTPUT

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESS** | **OUTPUT** |
| Mouse click | Looks for a state to be changed. And to move a camera when selected on map | Red unit on mini map, change in camera position |
| Keyboard state | Changing the camera motion. | Moving the camera also |

* DEBUG
* Executable game can only be played on computer
* No cheats
* CTRL + Esc – to exit

IF YOU GET PROBLEMS PLEASE INFROM US,WE WILL LIKE TO HEAR FROM YOU

siyabd@yahoo.com