# Structured Programming Project

## Part 1 – Develop a game

You will be supplied with a number of project specifications. Choose one of these and develop a game that meets the requirements of that specification.

There is some flexibility for customisation – you must meet the operational requirements listed in the specification, but anything not explicitly written there can be decided by you. Also, you may use graphics supplied by your tutor or develop your own, but you will not be assessed on the quality of your artwork.

Whichever game you choose, it must be a graphical game, interactive, and have some kind of scoring system.

Your solution MUST include:

* Variables (integers, strings, Booleans etc; with names according to the organisational standard below)
* Operators (++, -=, etc)
* Comparators (>, <, == etc)
* Sequence
* Selection (if or case statements)
* Iteration (for or while loops)
* Methods (using the methods supplied with XNA objects such as SpriteBatch.Draw() is acceptable)
* Arrays
* file input OR output (e.g. for saving the high score after each game)
* a user defined function
* comments (to the organisational standard below)
* proper indentation (to the organisational standard below)

You should keep a written account of problems as you come across them in the development process.

### Organisational Standard for development

**Line length**. Try to avoid lines of code that are too large for the screen (typically about 80 characters). You can split lines of code over two or more lines if necessary.

**Variable names**. Keep variable names short, but not too short – single character variable names like ‘x’ and ‘y’ are only acceptable for loops. Also, try to choose names which make it obvious what the variable is for.

**Comments**. Place a comment before each new function, explaining its purpose and parameters. You should also use a comment before each significant loop, selection or other code block explaining its purpose.

**Indentation**. Please follow the default indentation levels as suggested by the Visual Studio 2008 interface: indent all code inside each loop, selection or other code block.

## Part 2 – Test and Debug the game while developing it

The testing of the game does not need to be extensive. However you must create a written test plan of some sort, and then record the test results on it. If the test results prompt you to make changes to your program, you must record what was changed.

## Part 3 – Demonstrate your game

You will need to demonstrate your game to the lecturer. Don’t worry – it will be an informal presentation, and the only ‘checklist’ that will be used at the presentation will be one that *you* should provide, showing that you’ve met the requirements of the specification.

The lecturer should ask you a few questions, but that’s just to make sure that the work is your own.

Your presentation should demonstrate the game itself, but also the most important parts of the development you did: coding, testing, and debugging.

**Part 4 – Evaluation**

Your evaluation is a short (minimum 300 words) written report, and should be consistent with what you said during the demonstration. In it, you need to:

* Show that your solution meets all the requirements of the specification. (In other words, take your checklist from the presentation, and just write a few words saying what you did that met each requirement.)
* Reflect on the work carried out. (eg, which parts did you have difficulty with, and which parts do you think went well)
* Reflect on changes made. (eg, what changes did you make to the game and how well did they work out)
* Recommend future improvements to your game itself, or anything that goes with it (specification, test plan, demonstration etc)

### Development Log

|  |  |  |
| --- | --- | --- |
| **Date** | **Actions / Problems encountered** | **Resolution of problems** |
| 22/10/12 | Started to make game for test. Added variables, arrays[], and set the screen resolution . | None encountered |
| 23/10/12 | Wrote code for loading and drawing sprites on the screen also added player control | Sourced graphics from the internet |
| 24/10/12 | Added player bullets to the game added them to work with gamepad and drew them | Sourced graphics from internet |
| 25/10/12 | Added enemy[] 1 to the game loaded them ,and drew them. placed them outside the screen and set a position with a Randomiser() | Relooked at nc game for intuition on the use of randomisers; |
| 26/10/12 | Added enemybullet1[] to the game loaded them and set it to fire when player is in its sights. Drew enemybullet1[] | None. |
| 27/10/12 | Added sound Content to the game for all actions | Sourced sound from college and internet |
| 28/10/12 | Added enemy2[] to the game drew them and loaded them in | None used the same formula as enemy 1 |
| 29/10/12 | Added sound effect to the game for enemybullet2[] | none |
| 30/10/12 | Added enemybuulet2[] loaded them drew them | None used the same fire mechanism for as enemy1[]. |
| 31/10/12 | Added spider sprite to the game loaded it and drew it and tried to update it | Lots of problem with the randomiser but still got the spider to work but the randomiser never worked with the time delay. |
| 1/11/12 | Loaded animation to the game at point of impact of bullets | Animation don’t seem to play with the spider death tried a couple of ways the problem is the way I have got the spider to move. None proved fruit full moved on and come back if I have time |
| 1/12/12 | Loaded web for spider added array for it drew it updated it and applied timing to it | Lives went down drastically when hit by the array of webs so I made tehm all invisible if one hit the player’s ship. |
| 2.12.12 | Took points away from player if he fired a bullet that hit nothing | none |
| 3/12/12 | Added animation to play shp when he died | The animations position dis not follow the player’s ship position when it played. Solution unfound. |
| 4/12/12 | Added flashing effect to the player’s ship when he is invulnerable | None. |
| **11/12/12** | **Started to make code shorter by adding methods the methods that I added were for loading full size images and smaller objects,**  **Some of background images do not load properly with the method.** | **Retuned some of the values in the method and only one is not working now but that has values of its own and that might be why** |
| 18/12/12 | Started adding comments to my code so that it is easy to read for people, Some of my code was hard to figure out | Played game and worked out the code |
| 25/12/12 | Added scorpion to the game it was hard but in the end all it needed was a for loop of own | Scorpion dint spawn right |
| 09/1/13 | Added scorpion sting to the game and updated more comments | none |
| 11/01/13 | Tried to reset the level shortly after the spider is killed no ships come back tried a few different ways of calling reset level | Never got the reset level to work at all but did get the ships coming back(no spider) |

**Test plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test no.** | **Item being tested** | **Test Method** | **Expected result** | **Actual result** | **Comments** | **Module** |
| 1 | ship | Plug in gamepad to computer | Player ship moves according to left Thumb Stick | Player ship moves according to left thumb Stick |  | Update();  Draw(); |
| 2 | Enemy moving | Plug in game pad and watch for the enemy | Enemy appear on screen at the left and move left from the left of screen | Enemy appear on the screen at the left and move left |  | Update();  Draw(); |
| 3 | Player firing | Plug in game pad and push fire button | Player ship fires bullet which moves left | Player ship fires bullet which moves left |  | Update();  Draw(); |
| 4 | Enemy firing | Play game and look for enemy firing | When player is in line with enemy ship the enemy ship fires a bullet | When player is in line with enemy ship the enemy ship fires a bullet |  | Update()  Draw(); |
| 5 | Player bullet Collision response | Play game and look for enemy being shot | If player bullet hits enemy ship the enemy is invisible and so is the player bullet and points are added on to the player score | Enemy ship is invisible when player bullets hit ship; |  | Update (); |
| 6 | Enemy bullet Collision response | Play game and get hit by enemy bullet | Player gets hit by enemy bullet becomes invisible, the enemy bullet becomes invisible the player loses a life and comes back out the left hand side of the screen | Player gets hit by enemy bullet becomes invisible , the enemy bullet becomes invisible the player loses a life and comes back out the left hand side of the screen |  | Update(); |
| 7 | Enemy2 moving | Plug in game pad and watch for enemy2 | Enemy2 appears on screen at the left and move left from the left of screen | Enemy2 appear on the screen at the left and move left |  | Update();  Draw(); |
| 8 | Enemy2 firing | Play game and look for enemy2 firing | When player is in line with enemy2 ship the enemy2 ship fires a bullet | When player is in line with enemy2 ship the enemy2 ship fires a bullet |  | Update()  Draw(); |
| 9 | Enemy2 bullet Collision response | Play game and get hit by enemy2 bullet | Player gets hit by enemy2 bullet becomes invisible, the enemy2 bullet becomes invisible the player loses a life and comes back out the left hand side of the screen | Player gets hit by enemy2 bullet becomes invisible , the enemy2 bullet becomes invisible the player loses a life and comes back out the left hand side of the screen |  | Update();  Draw(); |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **10** | **Player bullet collision enemy2** | **Play game get enemy 2 kill enemy 2** | **When player bullet connects with enemy2 the enemy2 is longer visible** | **Player hit enemy 2, enemy 2 is no longer visible** |  | **Update();** |
| 11 | Player bullet invisible after it hits enemy 2 | Hit enemy 2 with a bullet | the bullet should now be invisible | Hit enemy 2 with bullet, bullet is no longer visible. |  | Update();  Draw(); |
| 12 | Player gets points for killing enemy 2 | Kill enemy 2 | Player got points | Player got points |  | Update();  Draw(); |
| 13 | Player game is over by to many hit from enemy 2 | Play game get to enemy 2 die at least 5 times | When the player runs out of lives the game over screen appears | The game over screen appears when the game is over |  | Update();  Draw(); |
| 14 | Spider moves up and down when enemy 2 is past | Play game get past enemy2 wait for spider to emerge | When the player is past enemy 2 the spider should start going up and down at regular intervals | The spider started going up and down at regular intervals |  | Update();  Draw(); |
| 15 | When spider intersect player the player losses a life and spawns | Play game reach spider get hit off spider | When the spider hit the player the player should lose a life and spawn if he has lives remaining | The spider hit the player the player losses a life the player spawned when he had lives remaining |  | Update();  Draw(); |
| 16 | The spider web is firing | Get to spider go underneath it | when underneath the spider the spider should fire its web | Went underneath spider the spider fired its web |  | Update();  Draw(); |
| 17 | The spider web kill player if it hits him | Go to spider get underneath it and get hit off the web | When you get hit off the web you should lose a life | Went to spider went underneath it got hit off web and lost a life |  | Update();  Draw(); |
| 18 | Check player can fire upward when he reaches the spider | Go to spider press y on controller | A bullet should travel up the screen begging at the player ship position | Got to spider pressed the y button a bullet appeared and travelled up the screen |  | Update();  Draw(); |
| 19 | Check player can kill spider | Go to spider try to defeat spider by using the y button bullet | The spider should not come after 50 hits with the y button bullet | Defeated spider by pressing y button bullet the spider didn’t come down any more |  | Update();  Draw(); |
| **20** | **Check background stops scrolling when player reaches spider** | **Go to spider** | **The background should stop scrolling** | **Got to spider the background stopped scrolling** |  | **Update();**  **Draw();** |
| **21** | **Check the background moves after killing spider** | **Kill spider** | **The background should start moving** | **Killed spider the background started scrolling** |  | **Update();**  **Draw();** |
| 22 | Check the scorpion comes along with the spider | Get to spider | A scorpion should appear out the bottom every 2nd time the spider comes | Got to spider a scorpion appeared out the bottom |  | Update();  Draw(); |
| 23 | Check the scorpion fire it sting | Get to spider | The scorpion should fire its sting every time you are level with it | Got to spider the scorpion fired its sting every time I got level with it |  | Update();  Draw(); |
| 24 | Check the scorpion sting kills the player when it hits | Get to spider, get hit from the scorpion sting | Should lose a life when the scorpion sting it’s the player ship | Got to spider got hit from the scorpion sting lost a life when I got hit |  | Update();  Draw(); |
| 25 | Check the scorpion takes a life from the player when it hits him | Get to spider get hit from the scorpion | Getting hit off the scorpion should make you lose a life | Got to spider got hit from scorpion lost a life |  | Update();  Draw(); |
| 26 | Check the level goes round again when the spider is defeated | Get to spider kill the spider | After killing the spider the background should start to move again and after 4 seconds or so the game should reset all the enemy’s | Got to spider killed spider background started to scroll  Enemy started to come back but no spider or scorpion appeared |  | Update();  Draw();  Resetlevel(); |

**Demonstration checklist**

* Player can fire by pressing the A button
* Player can fire by Pressing the y button( at a certain time);
* Player cannot go past the half way mark on the screen
* When player hit enemy ship he is destroyed
* When player gets hit by enemy bullet he is destroyed.
* Different alien fire different bullets.
* when player loses a life and respawns at the start.
* The previous highscore is loaded and displayed throughout the game.
* If the player beats the highscore then it is saved at the end of the game.
* Player can move ship with xbox controller

**Evaluation**

Here is how my solution meets all the requirements of the specification…

My game meets all the criteria as it has all the required aspects asked for such as if statements. For loops, Variables, Operators, Comparators, Iteration, it also uses Methods and arrays it has user defined function such as the ability to fire bullets in different directions. It also has the ability to load and save a high score it has gamepad vibration, and the game also stops scrolling for a while to allow different types of enemy to attack the player the scrolling then continues my game also has sound effects and back ground music throughout the game, the background music pauses for a while when the background is stationary to allow the player the opportunity to realise he is at a turning point in the game. The back background music then resumes as does the scrolling of the background I also have animations for death of enemy ships. I also have the player lives and score displayer on the screen as well as the current high score and when the game is over a game over screen appears. The enemy bullets travel from right to left as do the enemy ships the enemy ship are in random formation the game gets harder as the game continues as there is a boss at the end of the level. The player receives points for destroying enemy’s and loses a life if he collides with the enemy ship or the enemy bullets connect with the player’s ship. The player also controls the ship and start with a predetermined number of lives (3). The player bullets travel from left to right as requested and has the option of firing upwards to kill other enemy’s at certain points in the game

The parts of the program that I had difficulty with were…

The spider moving down, up and out of the screen I could get him do it once then it got stuck.

Getting the animation to follow the player ship

Animations ( forgot to update them)

Positioning of enemy ships.

Getting the spider to function correctly

Getting the scorpion to work correctly(only works with certain size)

Resetting the level

The parts of the program that went well were…

Drawing

&& statements

if statements

for loops

making variables arrays etc.

getting some of bullets to fire.

Loading.

Scoring.

Collision response.

Moving the player ship

Adding pints

Saving high score

Drawing score and strings in general

The parts of the program that I had to change after testing were…

Animations following player

Spider web is not as intended but look good at first attempt so I kept it .

The spiders functions several times.

The scorpion several time

The reset level I never got working

Had to use two for loops for different code (moving enemy ship1 and moving enemy ship2) as one didn’t work(same with bullets).

If I had more time, I would like to make these improvements…

Sound animations add a turning point in game so it scrolled down ward (maybe level 2).

Add a collectable for the player to enable firing upward in steamed of just giving him that ability

Reset level,

Make game go quicker if level 2 got implemented

Add better graphics.

Make 2 player

## **Hand-in checklist**

**You need to hand all these in to pass:**

* Printout of all program code.
* Development log showing problems encountered
* Completed test plan
* Demonstration checklist
* Evaluation

**These aren’t mandatory but if you have them you might as well give us a copy:**

* Any other planning you have done: storyboards, flowcharts, concept art etc.
* Printed copy of presentation if you did it in powerpoint.
* Screenshots of gameplay.
* Any other documents you might have that relate to your program.

# Specifications

## Option 1: Side scrolling shoot em up

Inspired by the classic arcade game ‘R-Type’.

* The player must control a spaceship or similar object at the left of the screen.
* The player will begin with a set number of lives, for example 3 lives.
* You can move the ship around, and shoot missiles which go from left to right until they hit something or go off the right hand side of the screen.
* Enemy spaceships or similar objects will appear on the right hand side of the screen, either randomly or in a predetermined pattern.
* The enemy ships will move from right to left until they are destroyed by hitting a missile, or go off the left edge of the screen.
* If the player’s ship hits an enemy ship he will lose a life.
* If the player runs out of lives the game should end.
* If the player shoots an enemy he will gain points.
* Both lives and score must be displayed on the screen.
* As the game progresses the difficulty should increase, for example by making enemies more numerous or faster.

## Option 2: Single-screen shoot-em-up

Inspired by the classic arcade game ‘Robotron: 2084’.

* The player controls a robot or similar character, he can move up, down, left, right, or diagonally.
* The robot begins in the middle of the screen.
* The robot may shoot a missile in any direction.
* Enemy robots will continually spawn in random locations on the screen.
* If an enemy robot would spawn too close to the player, it should choose a new respawn location.
* The robots will move toward the player until they are destroyed or touch him.
* Enemy robots can be destroyed by shooting them. If this happens the player should gain points.
* If the enemy robot touches the player, the game is over.
* Helpless humans will also spawn randomly.
* If an enemy robot or a player missile touches a human, he will die.
* If the player robot touches a human, the player gains points and the human is rescued.

## Option 3: Top-down arcade game

Based on the classic arcade game ‘Frogger’.

* The player controls a frog or similar character, he can move up, down, left, or right.
* The frog begins at the bottom of the screen.
* The player will begin with 3 ‘frogs’ (a.k.a. lives).
* Dangerous vehicles and objects will spawn at set locations on the screen at the start of gameplay.
* Each dangerous object will move in a set direction, either left or right, across the screen.
* When a dangerous object reaches the edge of the screen, it will ‘warp around’ to the opposite side and begin moving across again.
* If the frog touches a dangerous object he will lose a frog and respawn at the bottom of the screen.
* If the frog reaches the top of the screen, he gains points, and that frog is removed.
* If the player loses all three frogs, the game is over.
* If the player runs out of frogs but has at least one frog at the top of the screen, he will advance to the next level and get bonus points.

## Option 4: Side-view arcade game

Based on the classic arcade game ‘Tapper’.

* The player controls a waiter serving sushi, or a similar character, at the side of the screen. He can move up or down between four bars.
* The player starts with a number of ‘lives’.
* The waiter may ‘serve’ food on any of the four bars, launching it as a projectile along the bar.
* If the food reaches the end of the bar, it falls off the bar and smashes, and the player loses a life.
* At random intervals, customers will appear at a random bar, at the opposite side of the bar from the player.
* Customers will gradually move toward the player’s side of the screen.
* If a customer makes it to the player’s side of the screen, the player loses a life.
* If food reaches a customer, he will go away and the player will gain points.

## Option 5: Design your own game.

If none of the previous four options suit you, you can come up with your own game design. HOWEVER! Make sure you create your own brief, like those on the previous pages, AND check with your tutor that it’s suitable, before going ahead and making the game.

Whatever your design is, make sure it follows these rules:

* The brief must be specific about how the game plays, but the graphics and sound don’t need to be so specific.
* It must be interactive – no demos!
* There must be something in the game that uses an array.
* It must load and/or save data to/from a file, for example a high score table, map, or so on.
* There must be some kind of scoring system.