Design Documentation

Gill, Surinder
1308896

Hu, Joshua
1311940

Lago, Nick
1302613

December 8, 2015

Contents

List of Tables

1 Revision History

Table 1: Revision History: Proof of Concept Plan

DATE	DEVELOPER	CHANGE	REVISION
November 6, 2015	Gill, Surinder	Initial Draft	0
November 6, 2015	Hu, Joshua	Initial Draft	0
November 6, 2015	Lago, Nick	Initial Draft	0
December 7, 2015	Gill, Surinder	Section 1,2,3,4,5,6,7,8	1
December 7, 2015	Hu, Joshua	Section 1,2,3,4,5,6,7,8	1
December 7, 2015	Lago, Nick	Section 1,2,3,4,5,6,7,8	1

1.1 Legend

Blue text indicate removed material. Red text indicated added material. A coloured section title indicates the entirety of the section was affected.

2 Introduction

This document indicates the Module Interface Specifications for the implementation of 'Floppy Fish'. The intent of this document is to facilitate design and maintenance of the project through the clear specification of the interfacing between modules in the program.

Complementary documents include the System Requirement Specifications and Module Guide.

The project "Floppy Fish" requires the MIS as it is a software project that is dependant on its internal behaviour and the way its modules interact with each other. The "Floppy Fish" project has at this stage completed critical phases and as such benefits by providing a detailed description of its design to act as a record of its model. This supports the other available documentation by providing a complete and comprehensive encapsulation of the project.

The document consists of a section entailing the module hierarchy, giving an overview of the modules in the project, a section detailing the MISs of each specific module, a section providing anticipated and unlikely changes, and a traceability matrix from the elicited requirements.

3 Module Hierarchy

Table 2: Module Hierarchy

Level 1	**Level 2**	**Level 3**
Hardware-Hiding Module		
Behaviour-Hiding Module	Rendering Module	Animation Module
Software Decision Module	Initializer Module	Collision Module

4 MIS's of Main.js

4.0.1 `Flappy_Fish.Draw`

Name	In	Out	Exceptions
clear	-	-	-
rect	int, int, int, int, int	-	-
circle	int, int, int, int	-	-
image	image, int, int	-	-
sprite	image, int , int, int, int, int, int, int, int, int	-	-
semiCircler	int, int, int, int	-	-
text	string, int, int, int, int	-	-

4.0.2 Assumptions

No assumptions

4.0.3 State Variables

4.0.4 Environment Variables

screen: computer/device display

4.0.5 Access Program Semantics

Input: Each function takes in an assortment of integers used for the location of the drawing to take place. In addition to this some take in a string or image depending on what the function is drawing.
Transition: All of these functions will *only* modify the state of the environment variable screen.

4.0.6 `Flappy_Fish.Input`

Name	In	Out	Exceptions
set	function	-	-

4.0.7 Assumptions

No assumptions

4.0.8 State Variables

x: int
y: int
tapped: boolean

4.0.9 Environment Variables

4.0.10 Access Program Semantics

Input: This will take in data in the form of a function.
Transition: This does not modify anything. Later another function will use the data from input to modify the screen displaying the game.

4.0.11 `Flappy_Fish.BottomBar`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-
respawn	-	-	-

4.0.12 Assumptions

No assumptions

4.0.13 State Variables

x: int
y: int
r: int
vx: int
bg: image
name: string

4.0.14 Environment Variables

Screen: Display Device

4.0.15 Access Program Semantics

Update:
Transition: Uses the velocity varaible (vx) to update the location variable (x) of the bottom bar.
Render:
Transition: Modifies the screen, drawing the bottom bar at the locations specified.
Respawn:
Transition: Modifies the variable x which controls the location of the bottom bar on the horizontal axis.

4.0.16 `Flappy_Fish.Pipe`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-
respawn	-	-	-
randomIntFromInterval	int, int	int	-

4.0.17 Assumptions

No assumptions

4.0.18 State Variables

centerX: int
coin: int
w: int
h: int
vx: int
type: string

4.0.19 Environment Variables

Screen: Display Device

4.0.20 Access Program Semantics

Update:
Transition: Uses the velocity varaible (vx) to update the location variable (centerx) of the pipe.
Render:
Transition: Modifies the screen, drawing the pipe and coin (if the coin hasn't been obtained) at the locations specified. (Using draw.sprite)
Respawn:
Transition: Modifies the varaible that controls the x axis locaiton of the pipe and calls randomIntFromInterval to modify where the gap will be.
randomIntFromInterval: Takes in a min and max value both as integers.
Output: Returns a number chosen at "random" from inbetween the min and max values.

4.0.21 `Flappy_Fish.Bird`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-

4.0.22 Assumptions

No assumptions

4.0.23 State Variables

img: image
gravity: int
width: int
height: int
ix: int
type: string
iy: int
fr: int
vy: int
vx: int
velocity: int
play: int
jump: int

rotation: int

4.0.24 Environment Variables

Screen: Display device
Speakers: Devices speakers if applicable

4.0.25 Access Program Semantics

Update:
Transition: Updates the location and rotation of the bird sprite. Modifies environmental variable speaker if applicable.
Render:
Transition: Modifies the screen, drawing the birdat the locations specified.

4.0.26 `Flappy_Fish.Particle`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-

4.0.27 Assumptions

No assumptions

4.0.28 State Variables

x: int
y: int
r: int
col: int
type: string
name: string
dir: int
vx: int
vy: int
remove: boolean

4.0.29 Environment Variables

Screen: Display device

4.0.30 Access Program Semantics

Update:
Transition: adds velocity in the x (vx) and in the y (vy) to the particle. Updating its location varaibles (x,y).
Render:
Transition: Modifies environmental variable screen by using the draw function to draw a star or a circle.

4.0.31 `Flappy_Fish.Collides`

Name	In	Out	Exceptions
collides	`Flappy_Fish.Bird`, `Flappy_Fish.Pipe`	c1 or c2	-

4.0.32 Assumptions

No assumptions

4.0.33 State Variables

bx1: int
by1: int
bx2: int
by2: int
upx1: int
upy1: int
upx2: int
upy2: int
1px1: int
1py1: int
1px2: int
1py2: int
c1: int
c2: int

4.0.34 Environment Variables

4.0.35 Access Program Semantics

4.0.36 `Window.Splash`

Name	In	Out	Exceptions
init	-	-	-
update	-	-	-
render	-	-	-

4.0.37 Assumptions

No assumptions

4.0.38 State Variables

banner: image

4.0.39 Environment Variables

Screen: Display Device
Speaker: Device Speakers (if applicable)

4.0.40 Access Program Semantics

init:
Transition: Modifies Floppy Fish's varaible distance and Floppy Fish's variable storing taps and Flappy Fish's variable storing coins gathered (to an initial value of 0). Pushes the bottombar (using the BottomBar function) at an initial coordinate. Lastly modifies the environmental variable speaker by playing a sound.
update:
Transition: calls all the update functions for evey entitiy pushed. Will change the state to play if the player has tapped.
render:
Transition: modifies the screen by using the Draw funciton in Floppy Fish to draw the banner.

4.0.41 `Window.Play`

Name	In	Out	Exceptions
init	-	-	-
update	-	-	-
render	-	-	-

4.0.42 Assumptions

No assumptions

4.0.43 State Variables

checkCollision: boolean

4.0.44 Environment Variables

Screen: Display Device
Speaker: Device Speakers (if applicable)

4.0.45 Access Program Semantics

init:
Transition: Pusheds 3 pipes, a bird and the counter.
update:
Transition: modifies the gradient of the game.
render:
Transition: Modifies the count varaibles and draws them to the screen.

4.0.46 `Window.GameOver`

Name	In	Out	Exceptions
getMedal	-	string	-
getHighScore	-	`FloppyFish.score.coins`	-
init	-	-	-
update	-	-	-
render	-	-	-

4.0.47 Assumptions

No assumptions

4.0.48 State Variables

checkCollision: boolean

4.0.49 Environment Variables

Screen: Display Device
Speaker: Device Speakers (if applicable)

4.0.50 Access Program Semantics

init:
Transition: Modifies environmental variable speaker by playing sound. Updates state variables from multiple other funcitons.
update:
Transition: Modifies the value of Floppy Fish variable holding if the user has tapped.
render:
Transition: Modifies the environmental variable screen as it uses the draw function to draw images and text.
getMedal:
Output: Outputs a string containing which medal the user has won.
getHighScore:
Output: Outputs the maximum amount of coins the user has received based on the cookies

the user has stored.

5 MIS of cookieHandler.js

5.0.1 `cookieHandler.js`

Name	In	Out	Exceptions
getCookie	string	string	-
setCookie	string, int, int	-	-

5.0.2 Assumptions

No assumptions

5.0.3 State Variables

name: string
ca: string
c: string
d: object
bigNum: long
expires: string

5.0.4 Environment Variables

path: cookie storage

5.0.5 Access Program Semantics

getCookie: Checks for a cookie with the name returns the cookie name as a string.
setCookie: Creates a cookie with a value and expiry date.

6 MIS of sound.js

6.0.1 sound.js

Name	In	Out	Exceptions
play_sounds	string	-	-

6.0.2 Assumptions

No assumptions

6.0.3 State Variables

soundJump: object
soundScore: object
soundHit: object
soundDie: object
soundSwoosh: object
sounds_max: int
sounds_length: int
audiosounds: object

6.0.4 Environment Variables

speaker: computer/device audio output

6.0.5 Access Program Semantics

play_sounds: If the audio objects n the array object are not finished load and play them.

7 MIS of windowSetter.js

7.0.1 window.requestAnimFrame

Name	In	Out	Exceptions
requestAnimFrame	-	function	-

7.0.2 Assumptions

No assumptions

7.0.3 State Variables

7.0.4 Environment Variables

screen: computer/device display

7.0.5 Access Program Semantics

requestAnimFrame: Requests an animation frame for various browsers and times out if no applicable animation frame is available.

8 MIS

8.1 Namespace.js

8.1.1 `Flappy_Fish`

Name	In	Out	Exceptions
init	-	-	-
resize	-	-	-
update	-	-	-
render	-	-	-
loop	-	-	-
changeState	state	-	-

8.1.1.1 Assumptions

requestAnimFrame(input) function is defined in a previous JavaScript class so Animation Frame functions can be called in the Namespace.

8.1.1.2 State Variables

Flappy_Fish.WIDTH: int
Flappy_Fish.HEIGHT: int
Flappy_Fish.scale: int
Flappy_Fish.offset.top: int

Flappy_Fish.offset.left: int
Flappy_Fish.entities: array
Flappy_Fish.currentWidth: null
Flappy_Fish.currentHeight: null
Flappy_Fish.canvas: null
Flappy_Fish.ctx: null
Flappy_Fish.score.taps: int
Flappy_Fish.score.coins: int
Flappy_Fish.distance: int
Flappy_Fish.digits: array
Flappy_Fish.fonts: array
Flappy_Fish.RATIO: null
Flappy_Fish.game: null
Flappy_Fish.currentWidth: null
Flappy_Fish.currentHeight: null
Flappy_Fish.canvas: null
Flappy_Fish.ua: null
Flappy_Fish.android: null
Flappy_Fish.ios: null

8.1.1.3 Environment Variables

Screen: Display Output Device
Mouse: Input Device
Keyboard: Input Device
Speakers: Audio Output Device

8.1.1.4 Access Program Semantics

init: Transition: Initializes multiple game variables and sets the canvas for the desired device. This function initializes the game .
resize: Transition: Maintains the aspect ratio of the game after resizing of the browser window and across multiple platforms.
update: Transition: Update the game and restore the tapped state of the game.
loop: Transition: Iterates the updating and rendering of the game.
changeState: Input: The state of the game that it is being changed to.
changeState: Transition: The state of the game that it is being changed to.

8.2 Main.js

8.2.1 `Flappy_Fish.Draw`

Name	In	Out	Exceptions
clear	-	-	-
rect	int, int, int, int, int	-	-
circle	int, int, int, int	-	-
image	image, int, int	-	-
sprite	image, int , int, int, int, int, int, int, int, int	-	-
semiCircler	int, int, int, int	-	-
text	string, int, int, int, int	-	-

8.2.1.1 Assumptions

No assumptions.

8.2.1.2 State Variables

8.2.1.3 Environment Variables

screen: computer/device display

8.2.1.4 Access Program Semantics

Input: Each function takes in an assortment of integers used for the location of the drawing to take place. In addition to this some take in a string or image depending on what the function is drawing.
Transition: All of these functions will *only* modify the state of the environment variable screen.

8.2.2 `Flappy_Fish.Input`

Name	In	Out	Exceptions
set	function	-	-

8.2.2.1 Assumptions

No assumptions

8.2.2.2 State Variables

x: int
y: int
tapped: boolean

8.2.2.3 Environment Variables

8.2.2.4 Access Program Semantics

Input: This will take in data in the form of a function.
Transition: This does not modify anything. Later another function will use the data from input to modify the screen displaying the game.

8.2.3 `Flappy_Fish.BottomBar`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-
respawn	-	-	-

8.2.3.1 Assumptions

No assumptions

8.2.3.2 State Variables

x: int
y: int
r: int
vx: int
bg: image
name: string

8.2.3.3 Environment Variables

Screen: Display Device

8.2.3.4 Access Program Semantics

Update:
Transition: Uses the velocity varaible (vx) to update the location variable (x) of the bottom bar.
Render:
Transition: Modifies the screen, drawing the bottom bar at the locations specified.
Respawn:
Transition: Modifies the variable x which controls the location of the bottom bar on the horizontal axis.

8.2.4 `Flappy_Fish.Pipe`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-
respawn	-	-	-
randomIntFromInterval	int, int	int	-

8.2.4.1 Assumptions

No assumptions

8.2.4.2 State Variables

centerX: int
coin: int
w: int
h: int
vx: int
type: string

8.2.4.3 Environment Variables

Screen: Display Device

8.2.4.4 Access Program Semantics

Update:
Transition: Uses the velocity varaible (vx) to update the location variable (centerx) of the pipe.
Render:
Transition: Modifies the screen, drawing the pipe and coin (if the coin hasn't been obtained) at the locations specified. (Using draw.sprite)
Respawn:
Transition: Modifies the varaible that controls the x axis locaiton of the pipe and calls randomIntFromInterval to modify where the gap will be.
randomIntFromInterval: Takes in a min and max value both as integers.
Output: Returns a number chosen at ”random” from inbetween the min and max values.

8.2.5 `Flappy_Fish.Bird`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-

8.2.5.1 Assumptions

No assumptions

8.2.5.2 State Variables

img: image
gravity: int
width: int
height: int
ix: int
type: string
iy: int
fr: int
vy: int
vx: int
velocity: int
play: int
jump: int

rotation: int

8.2.5.3 Environment Variables

Screen: Display device
Speakers: Devices speakers if applicable

8.2.5.4 Access Program Semantics

Update:
Transition: Updates the location and rotation of the bird sprite. Modifies environmental variable speaker if applicable.
Render:
Transition: Modifies the screen, drawing the birdat the locations specified.

8.2.6 `Flappy_Fish.Particle`

Name	In	Out	Exceptions
update	-	-	-
render	-	-	-

8.2.6.1 Assumptions

No assumptions

8.2.6.2 State Variables

x: int
y: int
r: int
col: int
type: string
name: string
dir: int
vx: int
vy: int
remove: boolean

8.2.6.3 Environment Variables

Screen: Display device

8.2.6.4 Access Program Semantics

Update:
Transition: adds velocity in the x (vx) and in the y (vy) to the particle. Updating its location varaibles (x,y).
Render:
Transition: Modifies environmental variable screen by using the draw function to draw a star or a circle.

8.2.7 `Flappy_Fish.Collides`

Name	In	Out	Exceptions
collides	`Flappy_Fish.Bird`, `Flappy_Fish.Pipe`	c1 or c2	-

8.2.7.1 Assumptions

No assumptions

8.2.7.2 State Variables

bx1: int
by1: int
bx2: int
by2: int
upx1: int
upy1: int
upx2: int
upy2: int
1px1: int
1py1: int
1px2: int
1py2: int
c1: int
c2: int

8.2.7.3 Environment Variables

8.2.7.4 Access Program Semantics

8.2.8 `Window.Splash`

Name	In	Out	Exceptions
init	-	-	-
update	-	-	-
render	-	-	-

8.2.8.1 Assumptions

No assumptions

8.2.8.2 State Variables

banner: image

8.2.8.3 Environment Variables

Screen: Display Device
Speaker: Device Speakers (if applicable)

8.2.8.4 Access Program Semantics

init:
Transition: Modifies Floppy Fish's varaible distance and Floppy Fish's variable storing taps and Flappy Fish's variable storing coins gathered (to an initial value of 0). Pushes the bottombar (using the BottomBar function) at an initial coordinate. Lastly modifies the environmental variable speaker by playing a sound.
update:
Transition: calls all the update functions for evey entitiy pushed. Will change the state to play if the player has tapped.
render:
Transition: modifies the screen by using the Draw funciton in Floppy Fish to draw the banner.

8.2.9 `Window.Play`

Name	In	Out	Exceptions
init	-	-	-
update	-	-	-
render	-	-	-

8.2.9.1 Assumptions

No assumptions

8.2.9.2 State Variables

checkCollision: boolean

8.2.9.3 Environment Variables

Screen: Display Device
Speaker: Device Speakers (if applicable)

8.2.9.4 Access Program Semantics

init:
Transition: Pusheds 3 pipes, a bird and the counter.
update:
Transition: modifies the gradient of the game.
render:
Transition: Modifies the count varaibles and draws them to the screen.

8.2.10 `Window.GameOver`

Name	In	Out	Exceptions
getMedal	-	string	-
getHighScore	-	`FloppyFish.score.coins`	-
init	-	-	-
update	-	-	-
render	-	-	-

8.2.10.1 Assumptions

No assumptions

8.2.10.2 State Variables

checkCollision: boolean

8.2.10.3 Environment Variables

Screen: Display Device
Speaker: Device Speakers (if applicable)

8.2.10.4 Access Program Semantics

init:
Transition: Modifies environmental variable speaker by playing sound. Updates state variables from multiple other funcitons.
update:
Transition: Modifies the value of Floppy Fish variable holding if the user has tapped.
render:
Transition: Modifies the environmental variable screen as it uses the draw function to draw images and text.
getMedal:
Output: Outputs a string containing which medal the user has won.
getHighScore:
Output: Outputs the maximum amount of coins the user has received based on the cookies

the user has stored.

8.3 cookieHandler.js

8.3.1 `cookieHandler.js`

Name	In	Out	Exceptions
getCookie	string	string	-
setCookie	string, int, int	-	-

8.3.1.1 Assumptions

No assumptions

8.3.1.2 State Variables

name: string
ca: string
c: string
d: object
bigNum: long
expires: string

8.3.1.3 Environment Variables

path: cookie storage

8.3.1.4 Access Program Semantics

getCookie: Checks for a cookie with the name returns the cookie name as a string.
setCookie: Creates a cookie with a value and expiry date.

8.4 sound.js

8.4.1 `sound.js`

Name	In	Out	Exceptions
play_sounds	string	-	-

8.4.1.1 Assumptions

No assumptions

8.4.1.2 State Variables

soundJump: object
soundScore: object
soundHit: object
soundDie: object
soundSwoosh: object
sounds_max: int
sounds_length: int
audiosounds: object

8.4.1.3 Environment Variables

speaker: computer/device audio output

8.4.1.4 Access Program Semantics

play_sounds: If the audio objects n the array object are not finished load and play them.

8.5 windowSetter.js

8.5.1 `window.requestAnimFrame`

Name	In	Out	Exceptions
requestAnimFrame	-	function	-

8.5.1.1 Assumptions

No assumptions

8.5.1.2 State Variables

8.5.1.3 Environment Variables

screen: computer/device display

8.5.1.4 Access Program Semantics

requestAnimFrame: Requests an animation frame for various browsers and times out if no applicable animation frame is available.

9 Changes

ANTICIPATED:

1. Additional functionality
2. Magnitude of game controls
3. Game media
4. Increased scope of support

UNLIKELY:

1. Input and Output
2. Game mechanics
3. Execution environment

10 Traceability Matrix

The following requirements reference the functional requirements in the document ”RequirementsDocumentation1.pdf”.

Table 3: Requirements Traceability Matrix

Requirements	Modules
4.4.1	8.1;8.5.1
4.4.2	8.1;8.5.1
4.4.3	8.1;8.2.9,10
4.4.4	8.1;8.2.9
4.4.5	8.2.9,10
4.4.6	8.1;8.2.9,10
4.4.7	8.1;8.2;8.3
4.4.8	8.1;8.2.10
4.4.9	8.1;8.2
4.4.10	8.1;8.2

The following anticipated changes reference section 9 of this document.

Table 4: Anticipated Changes Traceability Matrix

Anticipated Changes	Modules
1	8.2
2	8.2.5
3	8.2;8.4
4	8.5