

## **Fish Boid Project Product Backlog**

Author: Jacob Andrew Broomfield  
Date: 30<sup>th</sup> June 2025  
Version: 1.0  
Status: Done

## CONTENTS

1.	GENERAL DESCRIPTION.....	3
1.1	Product Goal.....	3
1.2	User Characteristics.....	3
2.	PRODUCT BACKLOG.....	4
2.1	User stories.....	4
2.2	Documentation stories.....	7

## **1. GENERAL DESCRIPTION**

### **1.1 Product Goal**

A successful product will provide a fish boid simulation program. There are no rules to follow as this program will simulate boids of fish schooling together and avoiding shark objects.

The different implementations necessary for this program include:

- Select fish type to spawn on the board.
- Select shark to spawn on the board.
- Spawn a fish/many fish of different/the same types onto the board.
- Fish move randomly around the board avoiding collisions with the border and other entities.
- Fish of the same type move towards each other to school together in a group.
- Different fish schools avoid clashing with one another without swimming away from each other.
- Spawn a shark onto the board.
- Shark moves around the board towards the fish avoiding collisions with the border.
- On contact with fish shark eats the fish.
- Fish scatter pushing and breaking the school to escape when near shark before returning to usual.

### **1.2 User Characteristics**

I can assume users have access to a pc and are capable of using a mouse to click buttons and click the simulation board. I will provide the user with information telling them where to click to spawn the selected fish buttons.

## **2. PRODUCT BACKLOG**

### **2.1 User stories**

#### **US1 Select fish type to spawn on the board.**

A user, when the program starts, will need to choose between a selection of 4 different fish types. This is required in order that we know which fish type is to currently be spawned on the board.

**Priority:** Must have

**Conditions of satisfaction:**

1. When the program begins, the user has 4 options to select between when choosing what fish is to be spawned when the board is clicked.
2. Only 1 fish type may be selected at one time, and any other selected fishes or shark should be unselected.
3. It needs to be clear that a specific fish type is currently selected.
4. The user cannot be allowed to interact with the board unless a fish has been selected.

#### **US2 Select shark to spawn on the board.**

A user, when the program starts, will need to have the option to select a shark option alongside the fish. This is required in order that we know when we are spawning the sharks onto the board.

**Priority:** Must have

**Conditions of satisfaction:**

1. When the program begins, the user has an option to select a shark to be spawned when the board is clicked.
2. Only the shark can be selected, and any fish should be unselected.
3. It needs to be clear when a shark has been selected.
4. The user cannot be allowed to interact with the board unless a shark has been selected.

#### **US3 Spawn a fish/many fish of different/the same type onto the board.**

When the user clicks the defined board space with a fish selected, it will spawn a fish entity at the spot clicked onto the board as often as clicked. This is required as a way to spawn entities on the board to carry out the boid simulation so that I can have as many fish as possible of whatever type I like on the board.

**Priority:** Must have

**Conditions of satisfaction:**

1. When the board is clicked, 1 of the selected fish type will spawn at that point.
2. If no fish is selected, no fish will spawn.
3. If anywhere but the board is clicked on the screen, nothing will spawn.
4. If a fish is spawned on top of another fish on the board, the fish will not spawn.
5. If a fish spawned close to another overlaps, the fish will move to align themselves separately.

#### **US4 Fish move randomly around the board avoiding collisions with the border and other entities.**

When spawned on the board, a fish will begin to move randomly in swerves around the board detecting objects like other fish and the board edge making sure to turn away smoothly. This will allow us to simulate many moving fish moving together in one school of fish.

**Priority:** Must have

##### **Conditions of satisfaction:**

1. When spawned a fish must immediately start randomly moving across the board.
2. When moving, the fish can only move in straight lines or large swerves.
3. When approaching other fish, the fish should avoid colliding if possible but slight collisions in the movement process can happen.
4. When approaching the board edge, fish must not exceed the border limits and if necessary, should swerve sharply.

#### **US5 Fish of the same type move towards each other to group together in a school.**

If 2 fish come into closer contact with each other, instead of avoiding each other they should school together. This will allow the different fish groups to form schools with each other to simulate the purpose of the program.

**Priority:** Must have

##### **Conditions of satisfaction:**

1. If 2 fish of the same type are close, they don't have to swerve to avoid collisions.
2. Fish in a school should aim to be separate in the boid space from one another.
3. Fish in a school should aim to have the same average direction of movement within the school.
4. Fish in a school should aim to stay close to the main centre of mass of the school.
5. If 2 schools of the same type of approach they should assimilate into one another and if they separate too much should divide.

#### **US6 Different fish schools avoid clashing with one another without swimming away from each other.**

When there are multiple fish schools of different types, we want 2 schools of different types to avoid one another and not assimilate without scattering but holding the main school structure. This is needed because it will allow us to simulate calm fish movements around one another in their schools.

**Priority:** Must have

##### **Conditions of satisfaction:**

1. When a school of 2 or more fish is present on the board, they will be checked by all other present schools on the board to anticipate the inevitability of collision.
2. If 2 or more opposing schools come within a certain distance of each other, each school close should try to change the average direction of said school towards a free space swerving without going too sharply.
3. If colliding, the swerve towards free space should continue without scattering allowing short temporary collision.
4. If collision is successfully avoided, allow schools to pass around one another not crossing a certain boundary.

### **US7 Spawn a shark onto the board.**

When the user clicks the defined board space with the shark selected, it will spawn a shark entity at the spot clicked onto the board as often as clicked. This is required as a way to spawn hostile entities on the board to carry out the boid simulation so that I can have my fish scatter and regroup.

#### **Conditions of satisfaction:**

1. When the board is clicked, the selected shark will spawn at that point.
2. If no shark is selected, no shark will spawn.
3. If anywhere but the board is clicked on the screen, nothing will spawn.
4. If a shark is spawned on top of another fish on the board, the shark will move over it.

### **US8 Shark moves around the board towards the fish avoiding collisions with the border.**

When spawned, the shark will go towards the closest fish to try and eat it. If the shark gets too close to the boarder, it swerves fairly sharply away from the border. This is needed to move the shark to the schools simulating the scattering.

**Priority:** Nice to have

#### **Conditions of satisfaction:**

1. The shark moves in swerves towards the closest fish entity to it trying to touch.
2. If a shark gets too close to the border it will swerve fairly tight away.
3. If a shark approaches another shark, it will swerve away to prevent collision.
4. Sharks have a 5 second attention span on a specific fish before targeting another to prevent endless following.

### **US9 On contact with fish shark eats the fish.**

When any fish touches a shark even slightly, it will be deleted from the board being eaten. This is to simulate how effective fish are at scattering from the shark by seeing which fish are caught or not.

**Priority:** Nice to have

#### **Conditions of satisfaction:**

1. If a shark touches another fish entity/entities, it/they are deleted and the shark resets its following.

### **US10 Fish scatter pushing and breaking the school to escape when near shark before returning to usual.**

When a fish sees a shark within its vicinity, it will move directly away causing a whole boid to stop its behaviour and temporarily scatter from a shark until safe. This is needed to help simulate the regrouping of fish into boids after chaos.

**Priority:** Nice to have

#### **Conditions of satisfaction:**

1. When a shark is seen, a fish must sharply turn the opposite direction from it.
2. Once far enough away, the fish will continue its grouping behaviour.

## **2.2 Documentation stories**

### **DS1 UI prototype**

I am going to have a UI prototype produced so that I can see the design of the user interface at an early stage in the process as I can then plan to implement functionalities accordingly from the beginning and I have a planned prototype I can follow rigorously.

**Priority:** Must have

### **DS2 Architectural design**

As a product owner, I wish to have an architectural design produced so that I can see the architectural design of the software at an early stage in order to provide feedback to developers and also so that the developers can reason about the functionality of the system and use it as a basis to implement that system.

**Priority:** Must have

## References

- [1] Loftus, C.W., "Software engineering group projects - quality assurance plan. Technical Report SE.QA.01", 3.0, Computer Science Department, 27<sup>th</sup> June 2023

## DOCUMENT HISTORY

<i>Version</i>	<i>Date</i>	<i>Changes made to document</i>	<i>Changed by</i>
0.1	17/06/25	Setting up basic document structure.	Jacob Broomfield
0.2	22/06/25	Laying out basic backlog user stories and other doc setup.	Jacob Broomfield
0.3	23/06/25	Detailing the first few user stories and tweaking document structure.	Jacob Broomfield
0.4	26/06/25	Finishing the rest of the user stories.	Jacob Broomfield
1.0	30/06/25	Finishing up the document.	Jacob Broomfield