Feeding Nemo

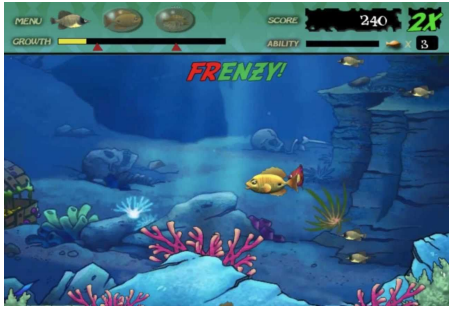
TEAM PINK
Ayush Pandey, Sara Alkhaja,
Fatema Nassar

INTRODUCTION

- Our goal is to implement a simple game design using a processing integrated development environment.
- We decided to make a simplified version of Feeding Frenzy.

What is Feeding Frenzy originally?

There's plenty of fish in the sea, and you're going to eat them all! Play as one of five hungry marine animals and chow down on an ocean full of smaller fish. You get bigger the more you eat, so unhinge that jaw and gnaw your way to the top of the food chain!



Our Modification

We will make a 2-level simplified version of feeding frenzy where the fish, as it moves along the position of the mouse on the screen.Smaller and larger fish will move around randomly along the background.

How Finding Nemo Works?

(strategy, win, and lose)

Level 1

- The Player will control a medium-sized fish that will eat the smaller fish than if it touches.
- If the player was touched by one of the five bigger fish in the area, the player dies.
- If the player eats all the small fish you go to the next level.

Level 2

- Player fish will grow, being able to eat 2 types of fish, and will have 3 fish size larger than it, that can eat the player, and then the game would end.
- Every 30 seconds, a large shark, which the player has to avoid in order to live, will pass at a random y-location eating nemo if it crosses it's way.
- If a player eats all the fish, the game ends and the player will win, otherwise the player restarts the game.

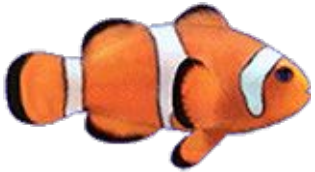
- The score will be incremented by 10 for every small fish eaten, and by 1 for every token eaten.

Our sprites

Preys:

Predators:

Nemo(Player):



Tokens:

- Eat all the preys to win!
- Take care of predator danger!
- Tokens increase your score!

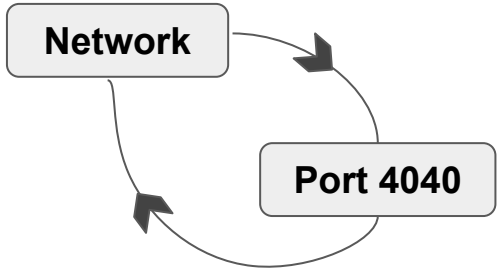
ONLINE MULTIPLAYER MODE

socket, threading

Microsoft Azure

Virtual Machine
OS - Linux [Ubuntu 18.04]

Client - Server
communication using socket



Feeding Nemo UML

Network

- FORMAT: str
- ADDR: tuple
+ player: int
+ scores: dict
+ success: bool
+ serverFull: bool

+ connect(): None
- send(): None
+ updateScore(): None

Token

+ posX: int
+ posY: int
- img: PImage
- size: int
- cropStart: int
- imageCount: int

+ display(): None

Background

+ w: int
+ h: int
- xShift: int
- otherImages: list

+ display(): None

Game

+ w: int
+ h: int
+ multiPlayer: bool
+ network: Network
+ keyHandler: dict
+ screen: int
+ score: int
+ level: int
+ start: datetime
+ bg: Background
+ nemo: Player
+ preys: list
+ predators: list
+ tokens: list
+ shark: Shark
+ background: Minim
- gameOverAudio: Minim
- mainMenuImg: PImage
- level2WelcomeImg: PImage
- winImg: PImage

- addCharacters(): None
+ update(): None
- getTimer(): str
- newGame(): None
- mainMenu(): None
- resetKeys(): None

Fish

+ posX: int
+ posY: int
+ rotation: float
# img: PImage
+ size: int
+ speed: float
+ direction: str
# cropStart: int

+ display(): None
+ getCenterCoordinates():float, float

Player

+ imageCount: int
+ alive: bool
+ eat: Minim
+ tokenEat: Minim

update(): None

Enemy

+ imageCount: int
- increment: list

update(): None
check(): None
getRadius(): float

Shark

+ inGame: bool
+ sound: Minim

update(): None

Roles & Deadlines

AYUSH PANDEY

- UML and Project design
- Fish Class
- Multiplayer

SARA ALKHAJA

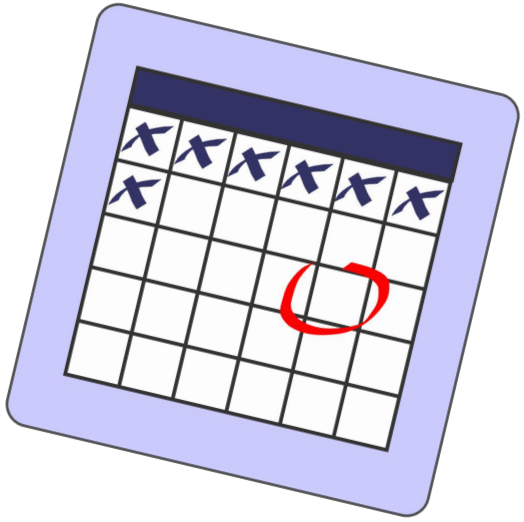
- Presentation slides
- Enemy classes
- Level 2

FATEMA NASSAR

- Proposal PDF
- Level 2, Background and tokens Class
- Backgrounds, sprites & tokens graphics

EVERYONE

- Choosing sprites
- Audio(minim)
- Game class and final Adjustments



- ★ Almost daily meetings discussing roles assigned and going through the code
- ★ Graphics are made on photoshop using separated frames from gifs extracted from the web.