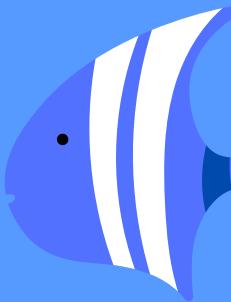




LOVE
LOVE
JELLYFISH



MEMBER

MINI PROJECT SIGNAL



ALEENA
KAEWKRISADA

6601012620135



NOPPASIN
RENRUANG

6601012620119



CHAYADA
VANITCHAKULPITAK

6601012620071

CONTENTS

01

GAME CONCEPT

02

SPRITE

03

COLLISION

04

HARDWARE &
SOFTWARE
CONCEPTS

05

THREADING

06

DEMO GAME



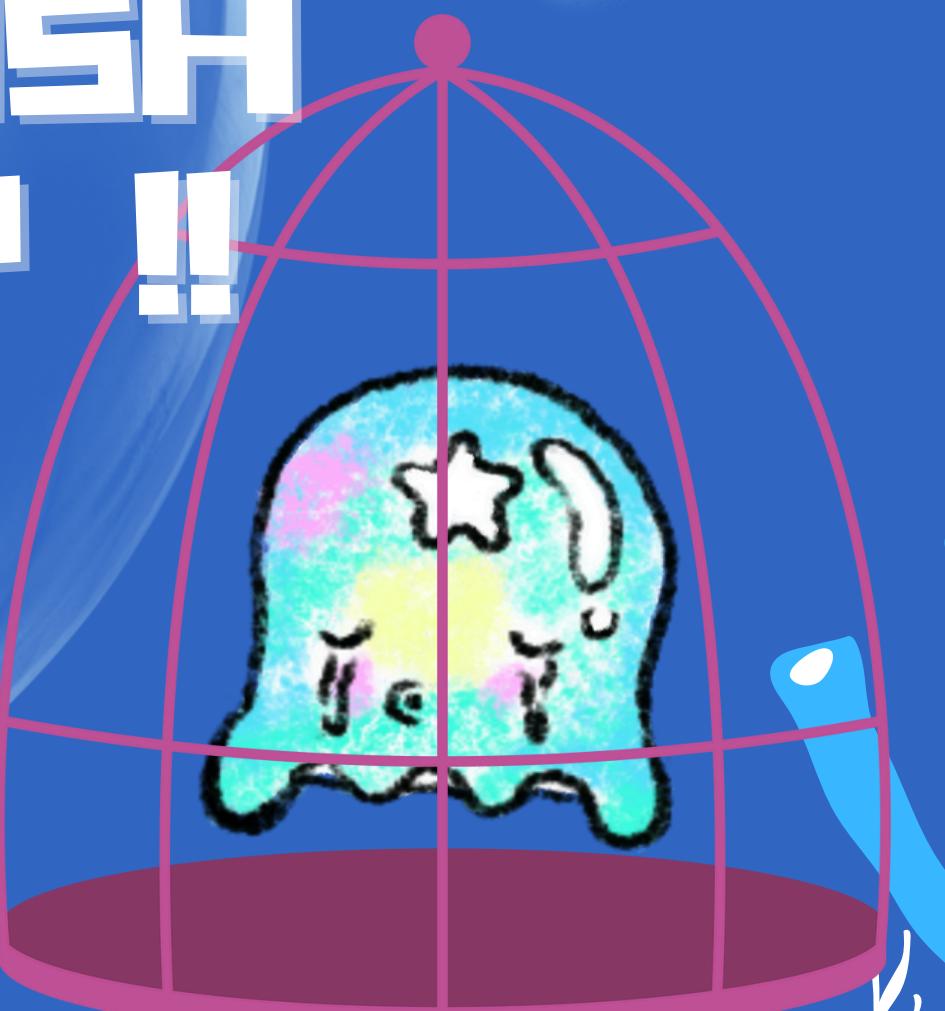
“
66

01

GAME CONCEPT



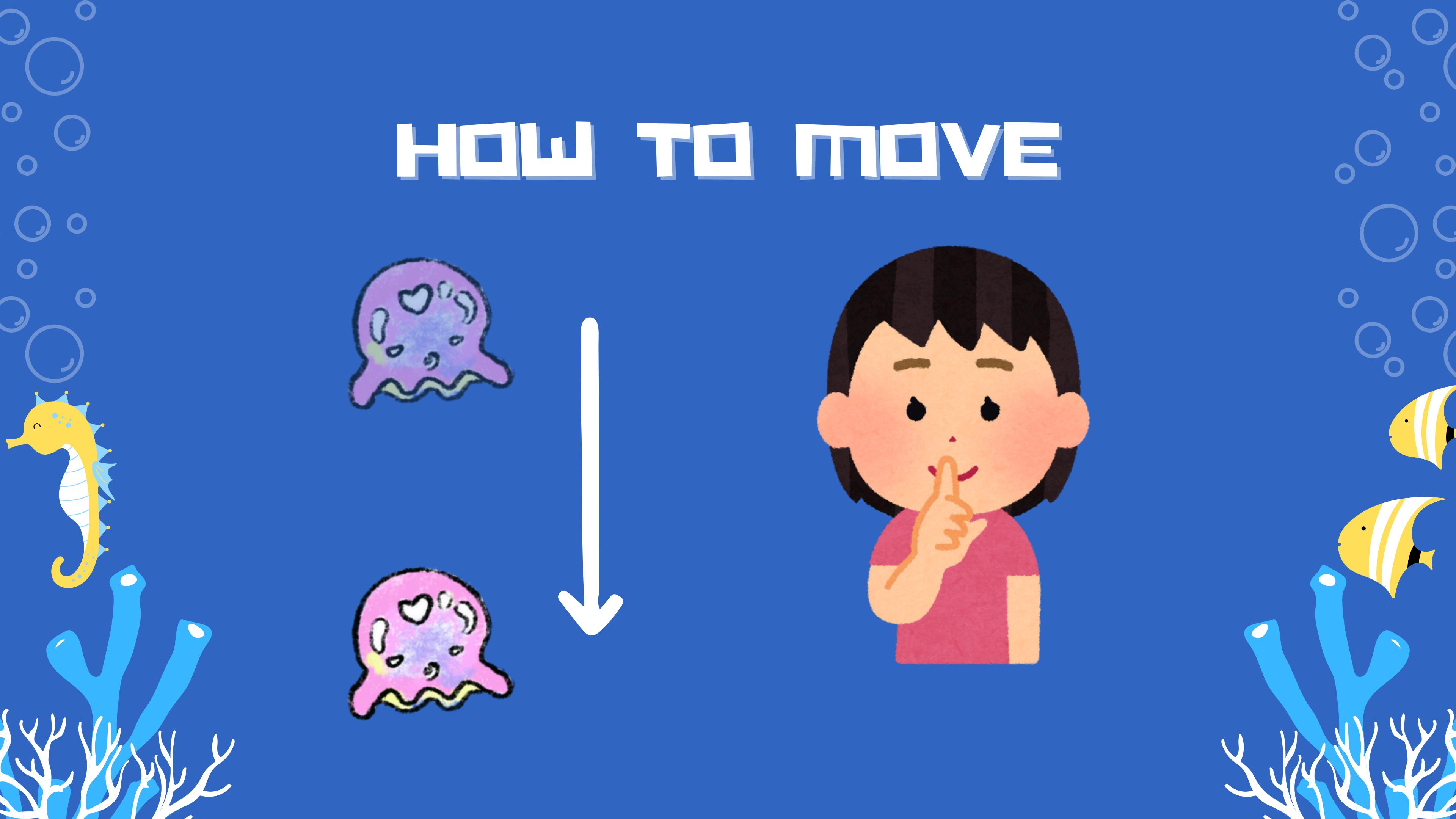
MISS JELLYFISH
NEEDS HELPP !!



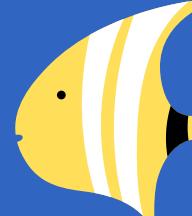
HOW TO MOVE



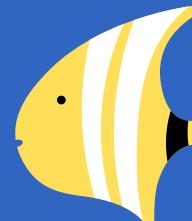
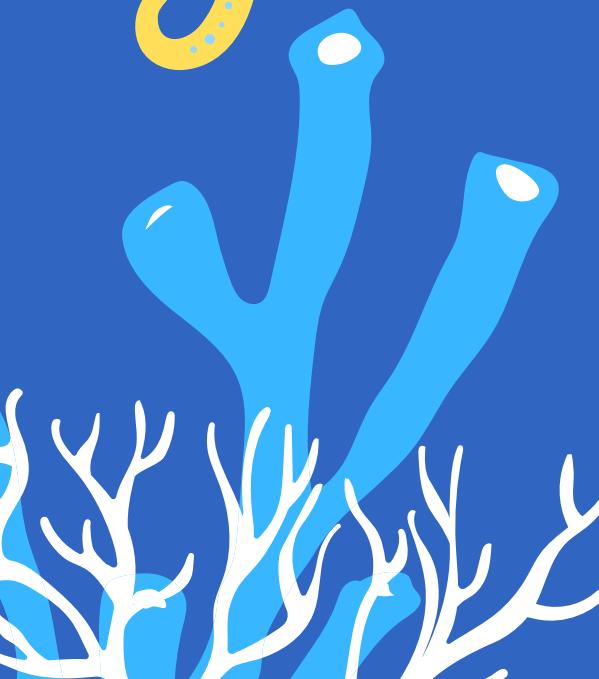
HOW TO MOVE



OBSTACLES DURING GAMEPLAY



END GAME



66

02

Sprite

SPRITE

SUB_IMG_W



SUB_IMG_H



```
1  class Jellyfish(pygame.sprite.Sprite):
2      def __init__(self, jellyfish_img, jellyfish_num_sub_imgs, x_position, y_position):
3          super(Jellyfish, self).__init__()
4
5          # jellyfish_num_sub_imgs = 3
6          jellyfish_sub_img_w = jellyfish_img.get_width() // jellyfish_num_sub_imgs
7          jellyfish_sub_img_h = jellyfish_img.get_height()
8          self.jellyfish_sub_imgs = []
9
10         for i in range(jellyfish_num_sub_imgs):
11             x = i * jellyfish_sub_img_w
12             f = jellyfish_img.subsurface(x, 0, jellyfish_sub_img_w, jellyfish_sub_img_h)
13             self.jellyfish_sub_imgs.append(f)
14
15         self.jellyfish_repeat = FPS // jellyfish_num_sub_imgs
16         self.jellyfish_last_frame = (jellyfish_num_sub_imgs * self.jellyfish_repeat) - 1
17
18         self.image = self.jellyfish_sub_imgs[0]
19         self.rect = self.image.get_rect(center=(x_position, y_position))
20         self.index = 0
21         self.speedx = 5
22         self.distance = 10
23
24     def update_animation(self):
25         if self.index >= self.jellyfish_last_frame:
26             self.index = 0
27
28         i = self.index // self.jellyfish_repeat
29         self.image = self.jellyfish_sub_imgs[i]
30         self.index += 1
```



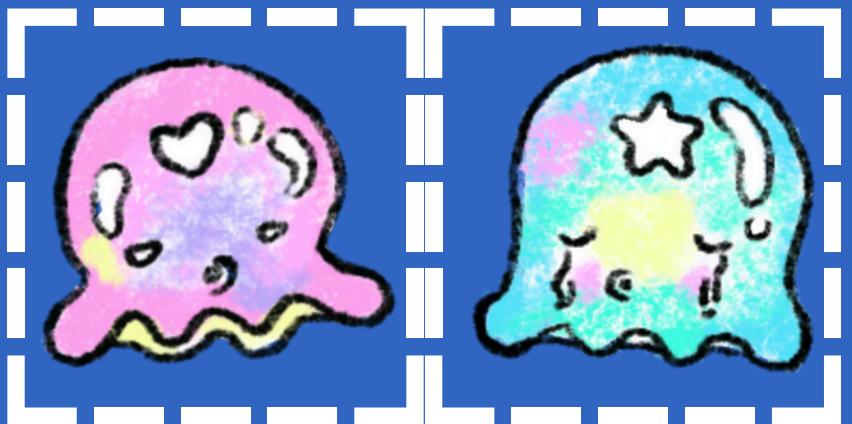
66

03

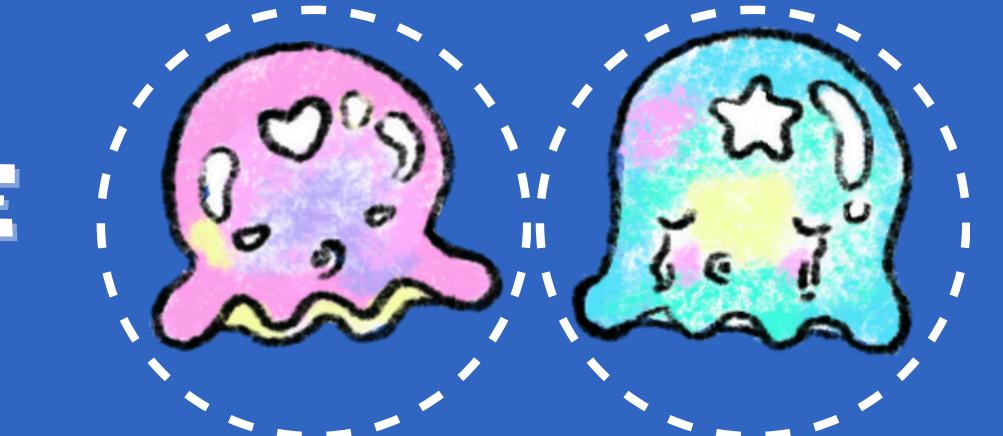
COLLISION

COLLISION

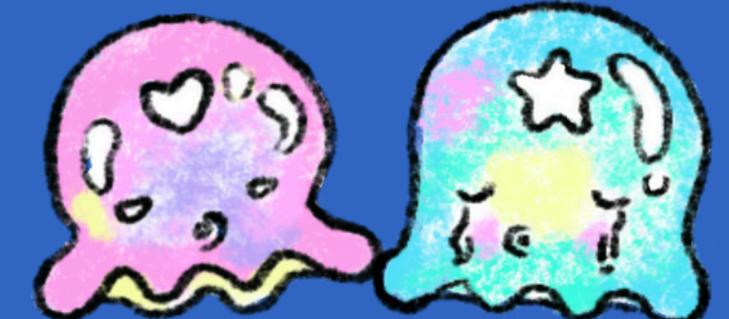
COLLIDE_RECT



COLLIDE_CIRCLE



COLLIDE_MASK



“
66

04

HARDWARE & SOFTWARE CONCEPTS

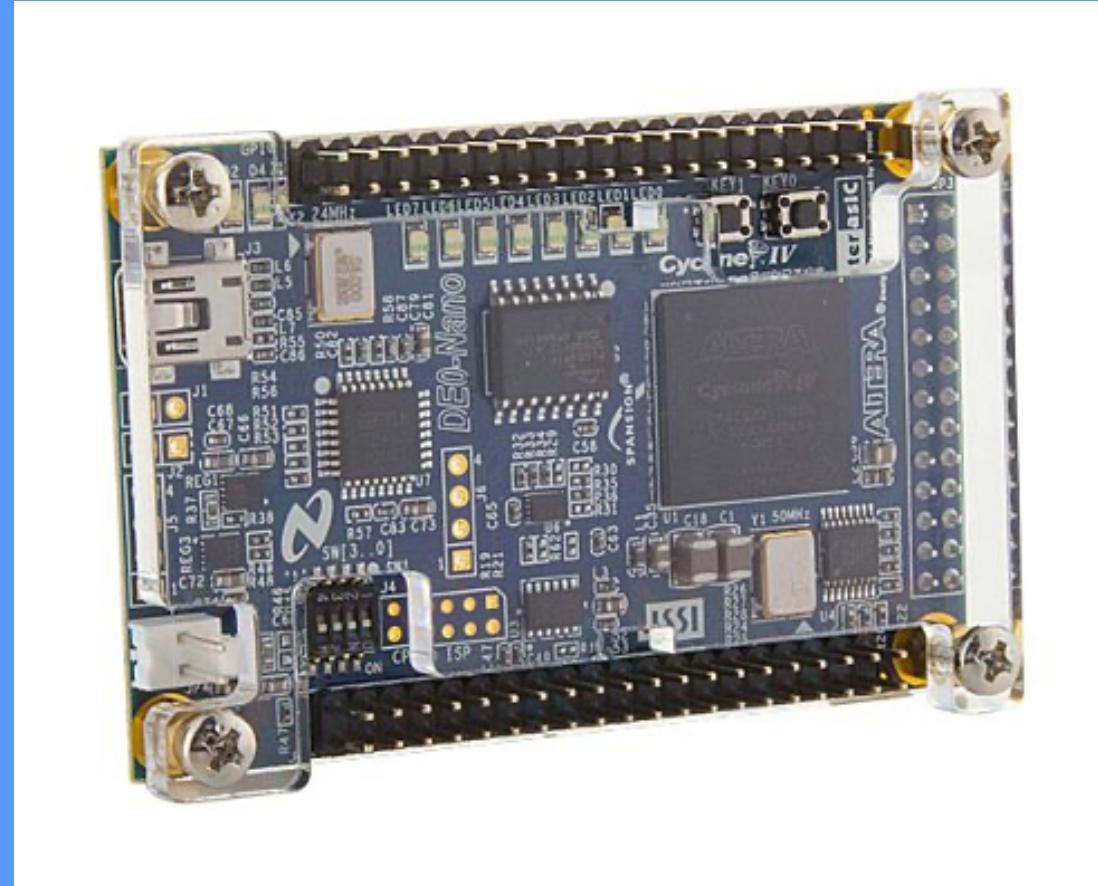
HARDWARE CONCEPTS

MICROPHONE MAX9814



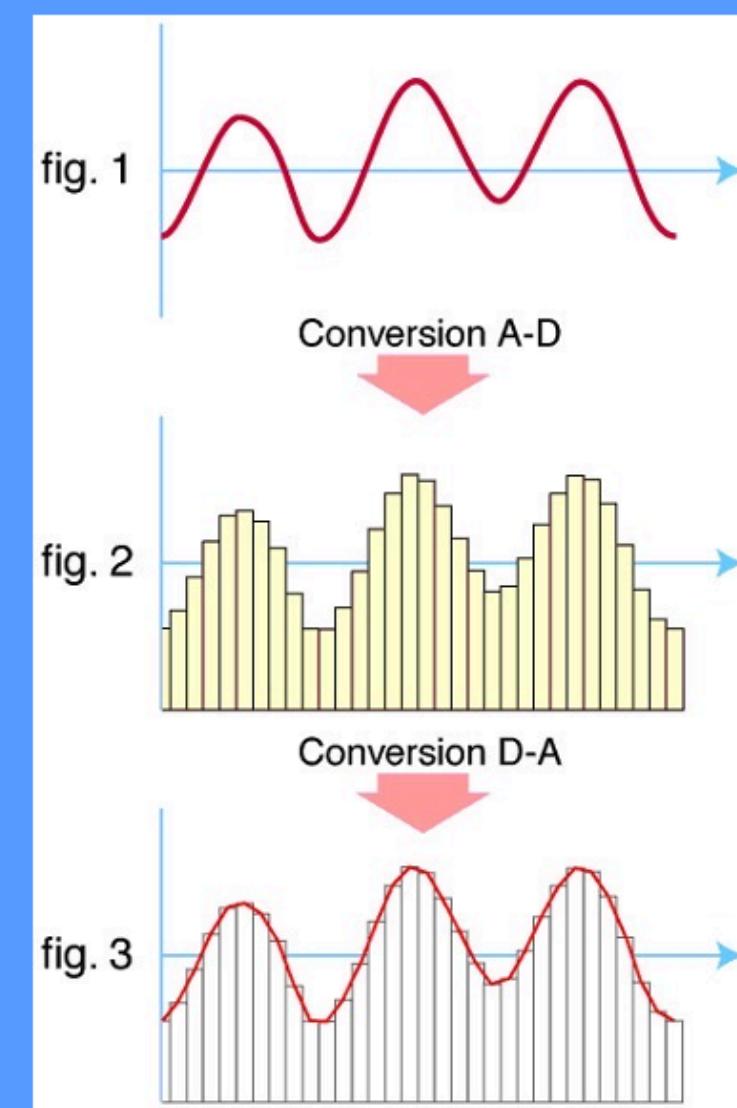
HARDWARE CONCEPTS

TERASIC DEO-NANO CYCLONE IV



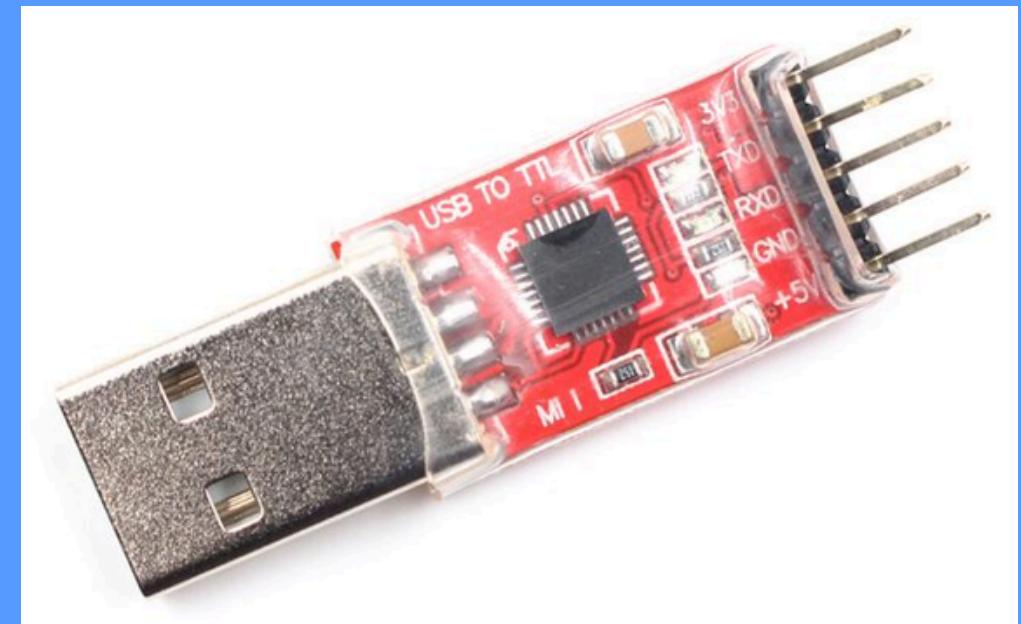
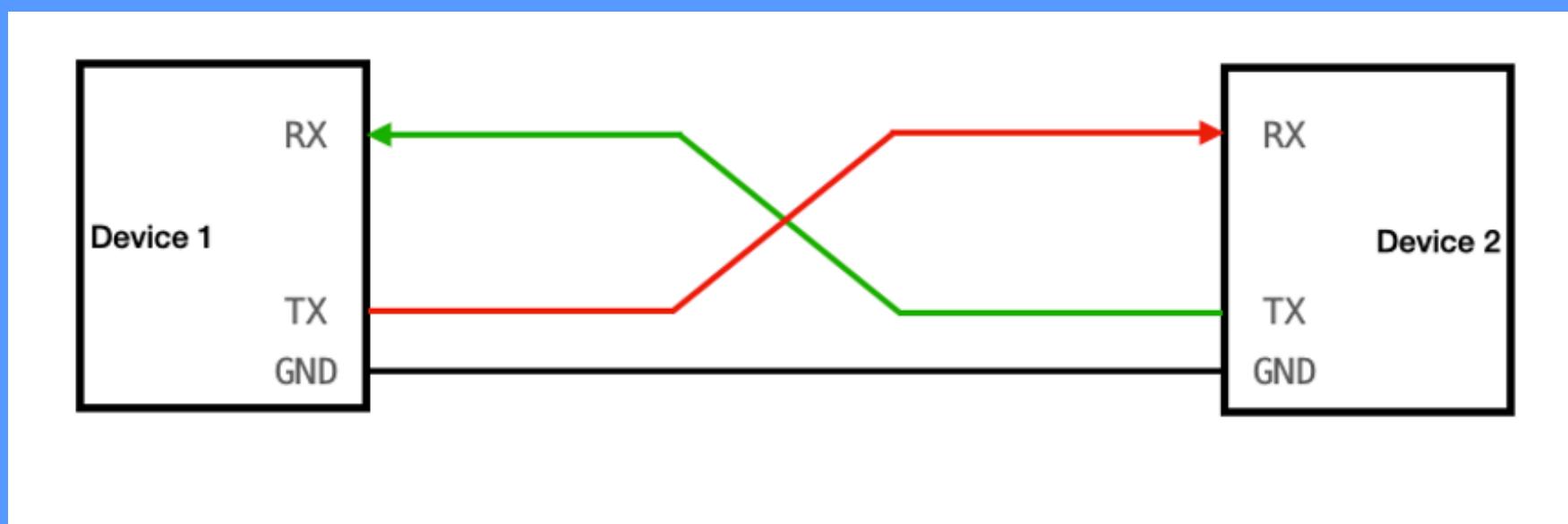
HARDWARE CONCEPTS

ANALOG TO DIGITAL CONVERTER (ADC)



HARDWARE CONCEPTS

UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER (UART)



SOFTWARE CONCEPTS

PYSERIAL



SOFTWARE CONCEPTS

NUMPY FFT

NumPy User Guide API reference Building from source Development Release notes Learn More ▾

Section Navigation

- NumPy's module structure
- Array objects
- Universal functions ([ufunc](#))
- Routines and objects by topic

 - Constants
 - Array creation routines
 - Array manipulation routines
 - Bit-wise operations
 - String functionality
 - Datetime support functions
 - Data type routines
 - Mathematical functions with automatic domain
 - Floating point error handling
 - Exceptions and Warnings ([numpy.exceptions](#))
 - Discrete Fourier Transform ([numpy.fft](#))

 - [numpy.fft.fft](#)
 - [numpy.fft.ifft](#)

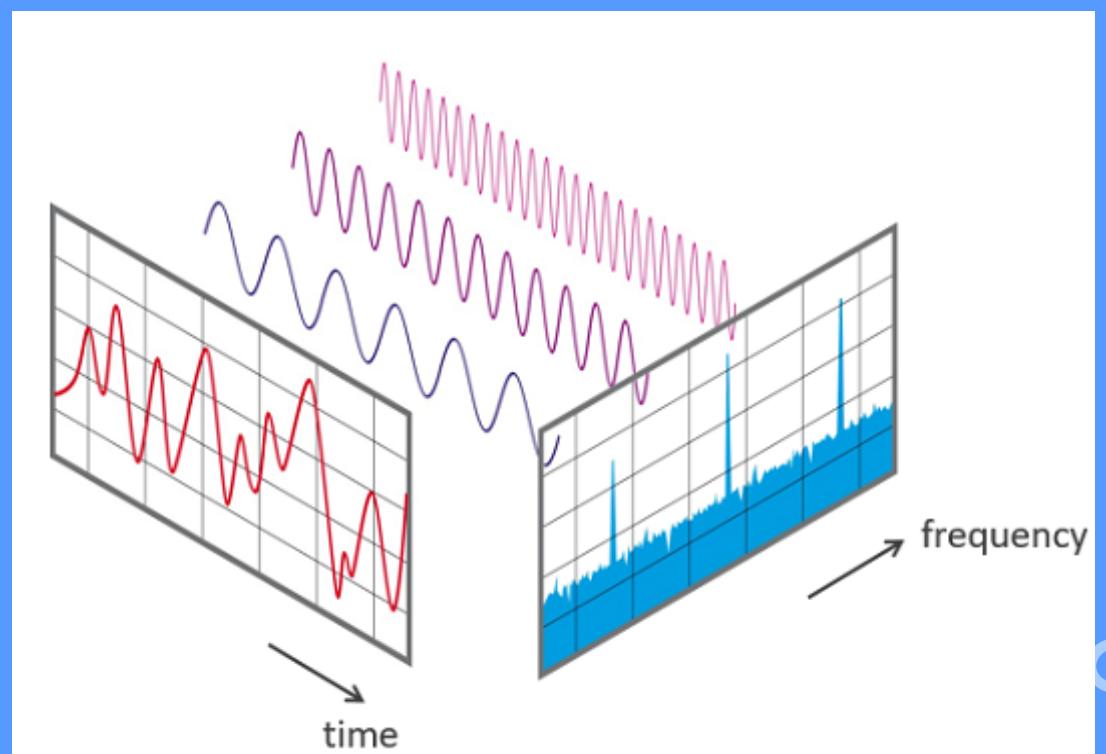
Home > NumPy reference > NumPy's module structure > Discrete...

Discrete Fourier Transform ([numpy.fft](#))

The SciPy module [scipy.fft](#) is a more comprehensive superset of [numpy.fft](#), which includes only a basic set of routines.

Standard FFTs

fft (a[, n, axis, norm, out])	Compute the one-dimensional discrete Fourier Transform.
ifft (a[, n, axis, norm, out])	Compute the one-dimensional inverse discrete Fourier Transform.
fft2 (a[, s, axes, norm, out])	Compute the 2-dimensional discrete Fourier Transform.
ifft2 (a[, s, axes, norm, out])	Compute the 2-dimensional inverse discrete Fourier Transform.
fftn (a[, s, axes, norm, out])	Compute the N-dimensional discrete Fourier Transform.
ifftn (a[, s, axes, norm, out])	Compute the N-dimensional inverse discrete Fourier Transform.

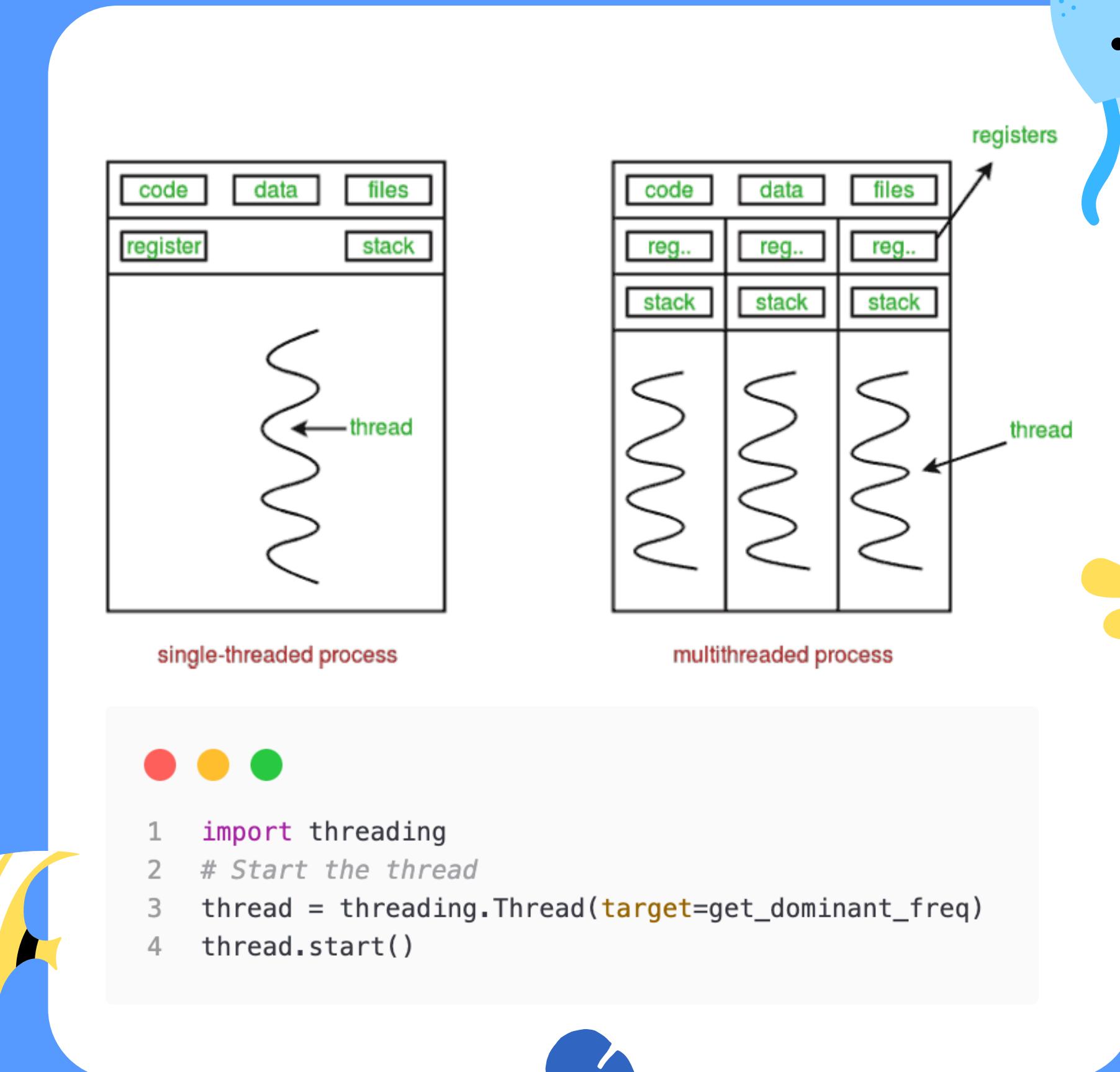
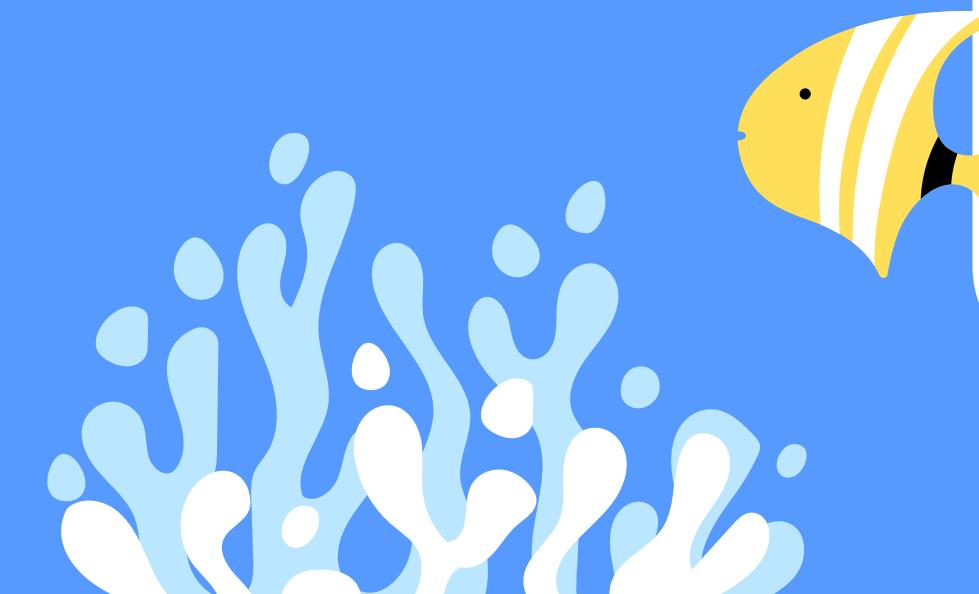


66

05

THREADING

THREADING



“
6

06

DEMO GAME

MAIN SCREEN

Love Love Jellyfish

Game Description:

In Love Love Jellyfish, players step into the translucent world of a determined jellyfish on a mission to find and rescue his beloved. The jellyfish's boyfriend is trapped in the deep sea, and our hero must navigate through an underwater maze of obstacles to reach her. The ocean is filled with floating debris, dangerous sea creatures, and mysterious objects that, if hit, will cause him to lose his energy and restart his journey.

Gameplay Mechanics:

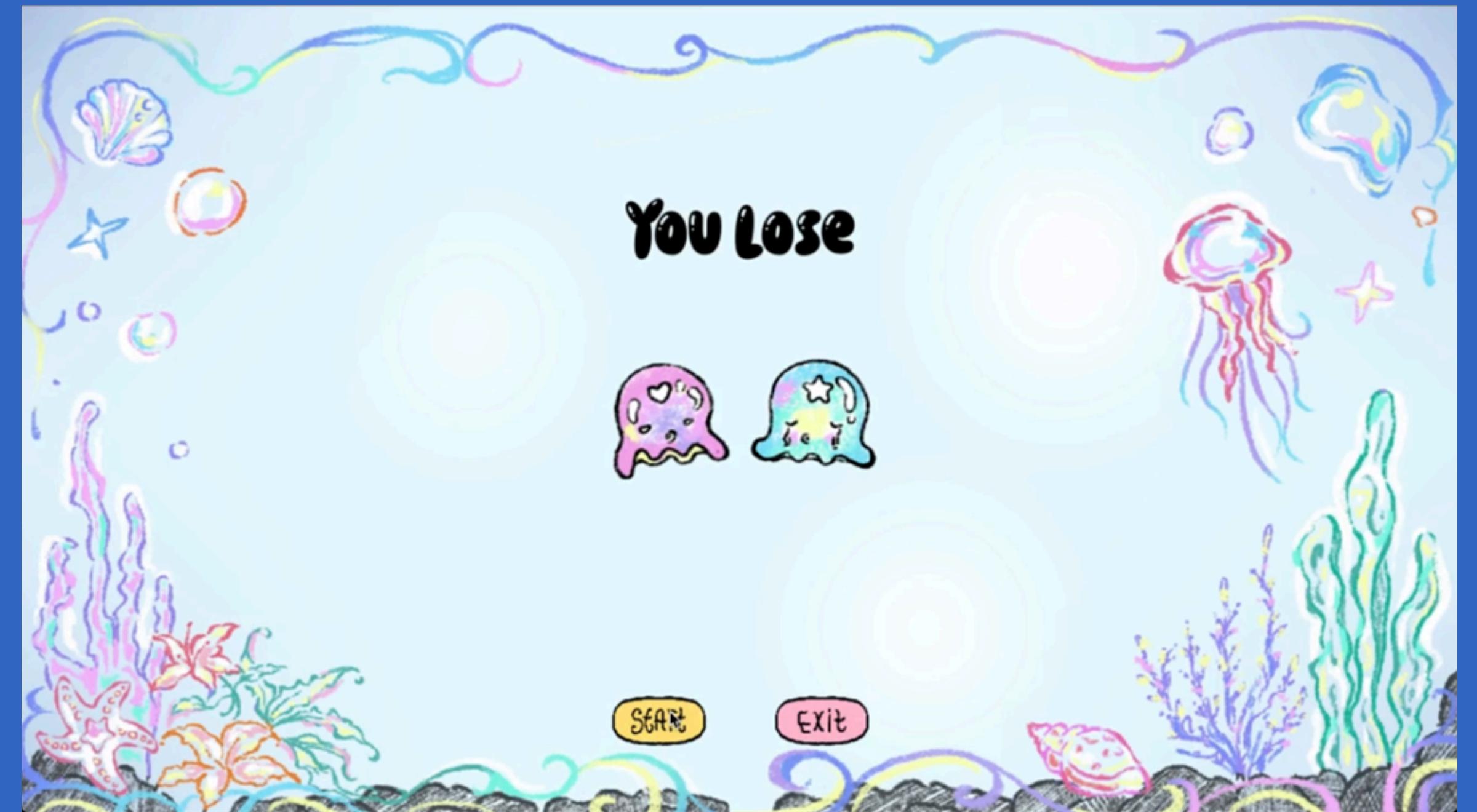
Frequency Control: Players use their own voice or a frequency controller to adjust the jellyfish's movement and frequency, avoiding obstacles. Humming or adjusting pitch lets the jellyfish dodge, dive, or rise as needed.

START

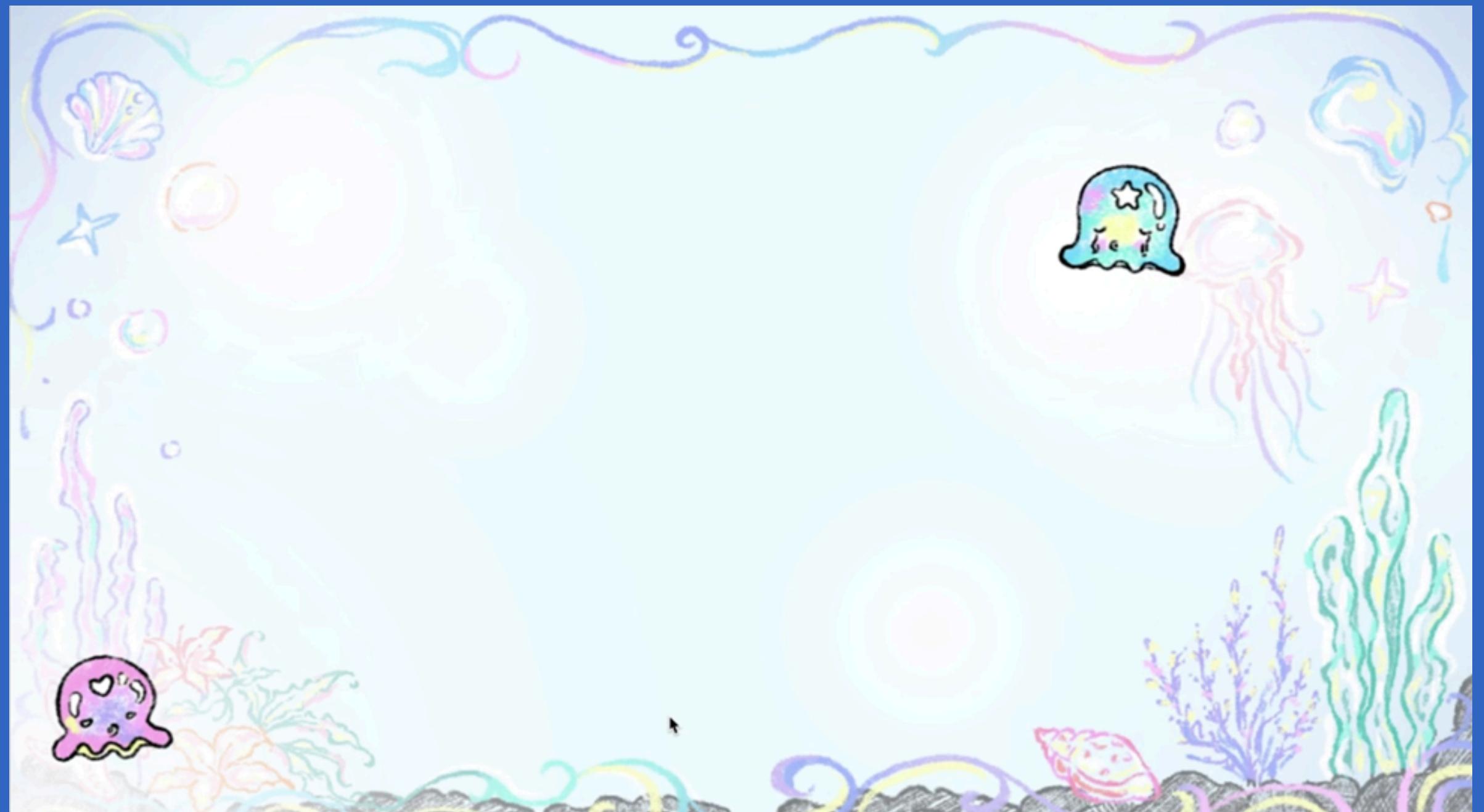
EXIT



**IF THE JELLYFISH COLLIDE
WITH AN OBSTACLE**



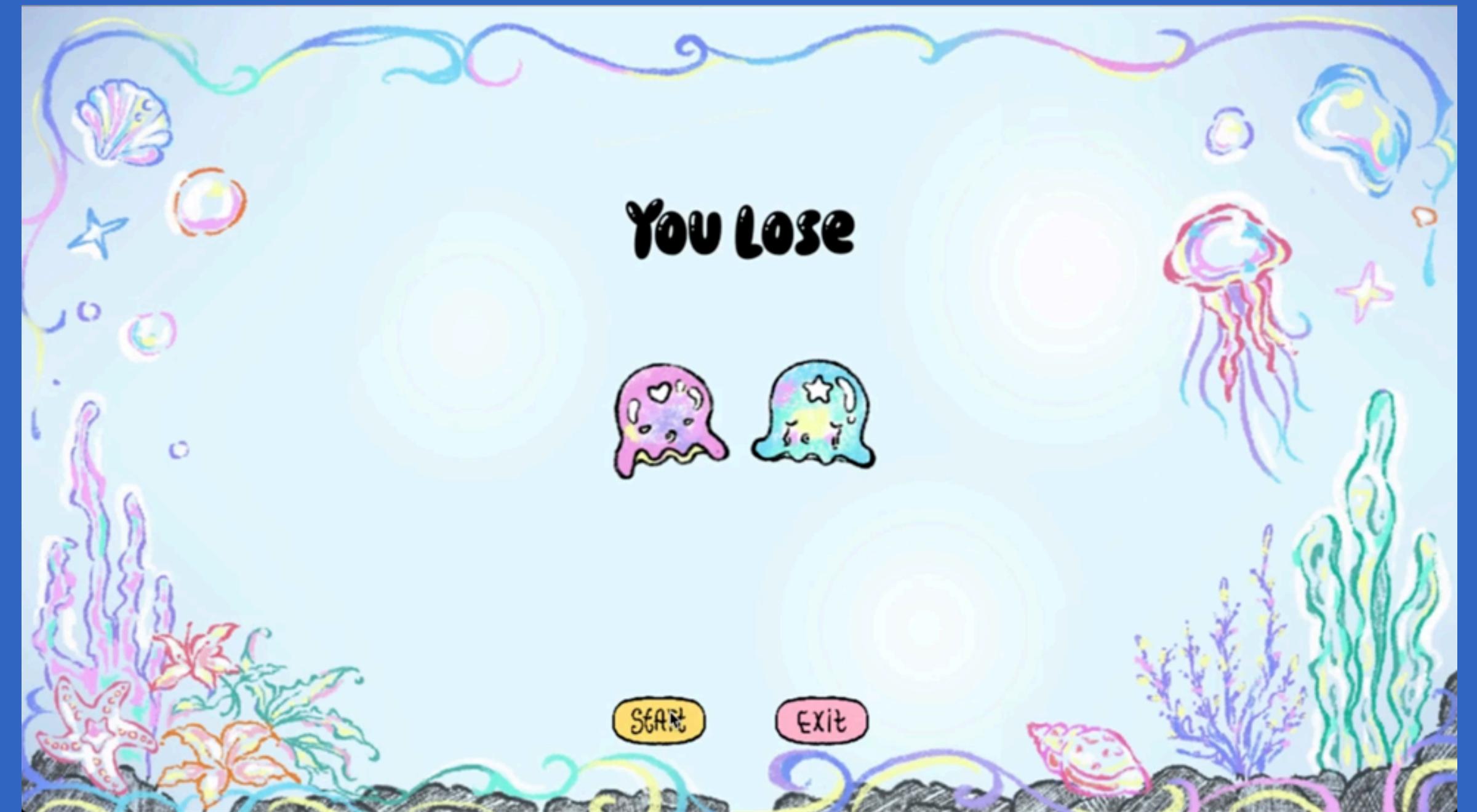
YOU LOSE



ONCE ALL OBSTACLES ARE
AVOIDED, THE JELLYFISH WILL
MEET ITS PARTNER



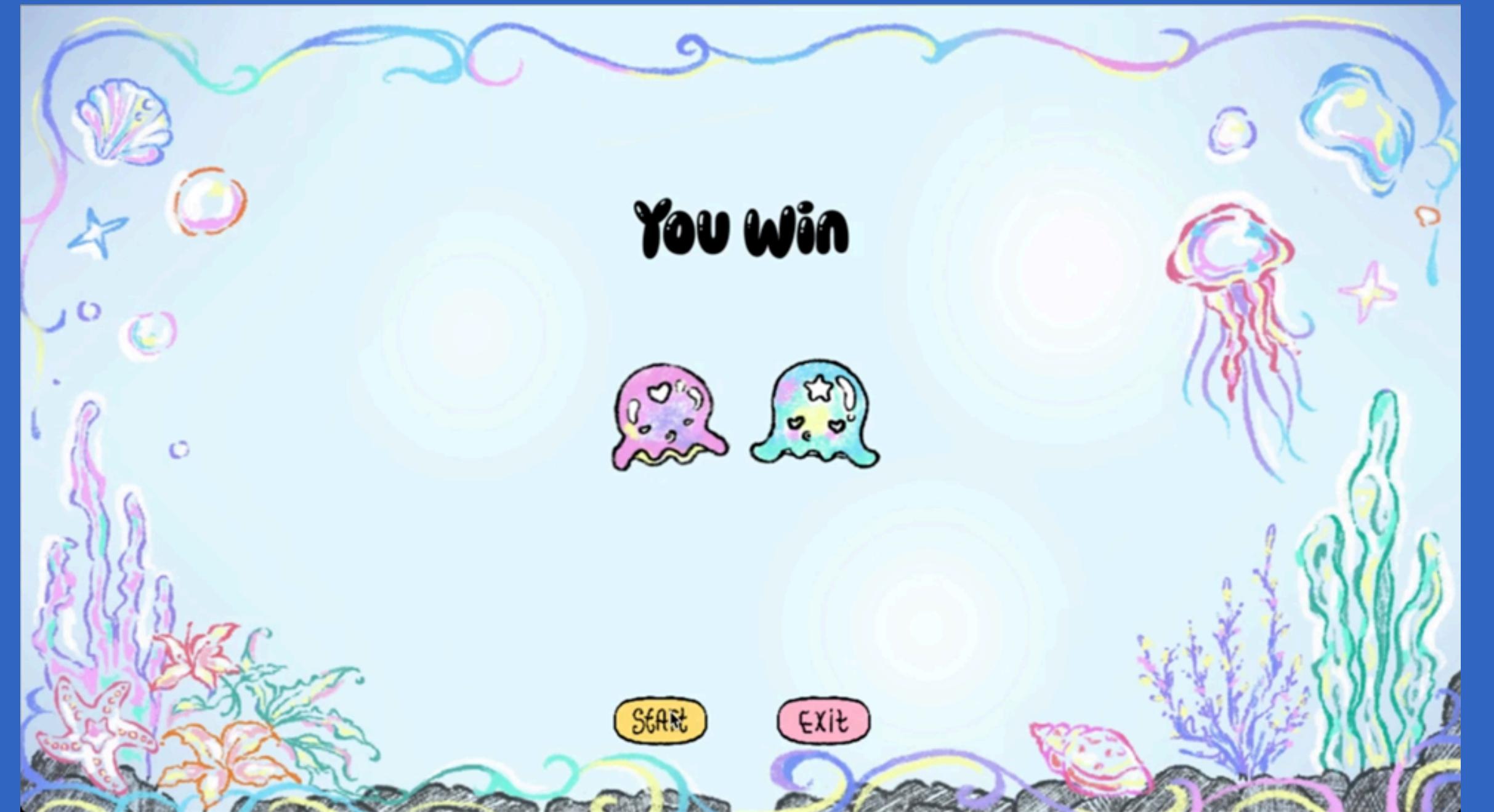
**IF JELLYFISH DON'T COLLIDE
WITH EACH OTHER**



YOU LOSE



**IF JELLYFISH COLLIDE
WITH EACH OTHER**



YOU WIN

CODE

