

# Rybmy

Kamilek :D

Listopad 2024

## Contents

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Pomysł . . . . .	2
1.2	Narzędzia . . . . .	2
1.3	Angular . . . . .	2
1.4	MySQL . . . . .	2
<b>2</b>	<b>Frontend</b>	<b>2</b>
2.1	Funkcje . . . . .	2
2.1.1	main-scene.component.ts . . . . .	2
2.1.2	shop.component.ts . . . . .	3
2.2	Klasy . . . . .	4
2.2.1	GameObject . . . . .	4
2.2.2	GameObjectsManager . . . . .	5
2.2.3	Item . . . . .	5
2.3	BaseItem . . . . .	6
<b>3</b>	<b>Backend</b>	<b>7</b>
3.1	Funkcje . . . . .	7
3.2	Klasy . . . . .	7

# 1 Wprowadzenie

## 1.1 Pomysł

Gra webowa o łowieniu ryb, w której gracz za złowione ryby będzie kupował nowy sprzęt i porównywał swoje osiągnięcia ze znajomymi

## 1.2 Narzędzia

## 1.3 Angular

pixi.js express.js

## 1.4 MySQL

# 2 Frontend

## 2.1 Funkcje

### 2.1.1 main-scene.component.ts

`interact() : void`

Funkcja ta jest wywoływana za każdym razem, gdy użytkownik naciśnie przycisk odpowiedzialny za bazowe interakcje ze światem gry. Zależnie od obecnego stanu gracza, kolejno albo gracz zarzuci wędkę, albo ją wyciągnie z wody, lub zatnie rybę, która bierze

`changePulling(newState : boolean) : void`

Funkcja ta odpowiada za zarządzanie przyciągania ryby w świecie gry. Wywoływana jest po przytrzymaniu lub odpuszczeniu przycisku bazowej interakcji

`podbierz() : void`

Funkcja odpowiedzialna za podbieranie ryby, pod warunkiem, że gracz może ją podbrać, co sprawdzane jest na bieżąco w funkcji `update()`

`loadSprite(path : string, x : number, y : number, isVisible : boolean): void`

Funkcja ta wywoływana jest w celu załadowania sprite'a z pliku do świata gry, ustawiając jego pozycję na osiach X i Y, oraz to czy jest on bazowo widoczny

`rabdin(min : number, max : number): number`

Funkcja zwracająca pseudolosową liczbę z zakresu podanego w argumentach funkcji

`fishPulledOut() : void`

Funkcja ta wywoływana jest po wyciągnięciu ryby z wody, która powiadamia gracza o złowionej rybie, oraz przywraca świat gry do bazowego stanu

fishLost(): void

Funkcja ta wywoływana jest po zerwaniu ryby, która powiadamia gracza o zerwaniu zestawu, oraz przywraca świat gry do bazowego stanu

update(time : any): void

Funkcja wywoływana co ticka w świecie gry, przez co wykonywana ona jest ciągle. Do niej dorzucane są funkcje i algorytmy, które muszą zarządzać zmiennymi w czasie rzeczywistym, albo nasłuchują wydarzeń.

durabilityChanged() : void Funkcja ta wywoływana jest za każdym razem gdy zmienia się wytrzymałość zestawu i przekazuje nową wytrzymałość w skali procentowej do obiektu rodzicielskiego, który następnie przekazuje ją do obiektu paska wytrzymałości

### 2.1.2 shop.component.ts

typEkwipunku oraz equipmentTypeArray

Typekwipunku jest customowym typem zmiennej w sklepie, w celu ułatwienia kooperacji i integracji kodu z wyglądem. Wszystkie nowe typy ekwipunku MUSZĄ zostać dodane do listy.

currIds: Map<typEkwipunku, number>

Mapa, której zadaniem jest mapowanie obecnie wyświetlanych w sklepie przedmiotów z ich identyfikatorami.

cash : number

Mamona, kasa, pieniądze, szmal, fundusze, budżet, pitos, kiesz

itemList : Map<typEkwipunku, Array<Item>>

Mapa, której zadaniem jest powiązywanie listy przedmiotów danego typu, wraz z ich typem przedmiotu. Wędki są trzymane osobno od Żyłek, Żyłki od Kołowrotek etc.

calculateMaxDurability() : void

Funkcja, wywoływana przy zmianie obecnie wyposażonego elementu ekwipunku, w celu rekalkulacji bazowej maksymalnej wytrzymałości całego zestawu, która następnie zostaje wysłana do main-scene, przez komponent rodzica

checkItemState(type : typEkwipunku) : string

Funkcja przyjmująca typEkwipunku, którego stan zostaje zwrócony do frontendu

changeItem(type : typEkwipunku, val : number) : void

Funkcja, która wywoływana jest w celu zmiany obecnie wyświetlanego przedmiotu z danego typu we frontendzie.

`getItemOfTypeAtID(type : typEkwipunku, id? : number) ; Item`

Funkcja typu Item, która zwraca przedmiot z danego typu ekwipunku na określonej pozycji. W przypadku braku określenia pozycji, kod uznaje, że chodzi o obecnie wyświetlany przedmiot.

`useItem(type : typEkwipunku) : void`

Funkcja wywoływana w momencie gdy użytkownik chce wykonać obecnie wyświetlaną czynność dla danego przedmiotu. Jeśli przedmiot został uprzednio zakupiony, zostaje on wyekwipowany, w przeciwnym wypadku, po uprzednim sprawdzeniu czy użytkownika stać na dany przedmiot, zostaje on zakupiony. Jeśli natomiast dany przedmiot jest już wyekwipowany, to zostaje on odekwi-  
powany

## 2.2 Klasy

### 2.2.1 GameObject

Obiekty tworzone na bazie tej klasy przechowują dane odnośnie obiektów fizycznych znajdujących się bezpośrednio w grze i które są widoczne dla użytkownika.

`private name : string;`

Prywatna zmienna typu tekstowego przechowująca nazwę obiektu

`private sprite : PIXI.Sprite`

Prywatny obiekt typu Sprite, który reprezentuje użytkownikowi przypisany do obiektu w grze

`public getSprite() : PIXI.Sprites`

Funkcja zwracająca sprite'a obiektu w grze

`changeSprite(path : string) : void`

Funkcja przyjmująca ścieżkę do pliku lokalnego nowego spritea dla danego obiektu w grze

`changeYPos(change : number) : void`

Funkcja przyjmująca wartość typu liczbowego, o którą ma zmienić się pozycja obiektu w grze na osi Y

`setYPos(newPos : number) : void`

Funkcja przyjmująca wartość typu liczbowego, która ustawia nową pozycję obiektu w grze na osi Y

`setSize(newSize : number) : void`

Funkcja przyjmująca wartość typu liczbowego, która ustawia rozmiar obiektu w grze na osi Y

constructor(name? : string, sprite? : PIXI.Sprite)

Konstruktor przyjmujący wartości przypisywane do zmiennych name i sprite

public show() : void

Funkcja, która pokazuje spritea obiektu

public hide() : void

Funkcja, która chowa spritea obiektu

### 2.2.2 GameObjectsManager

Obiekt typu singleton, zarządzający wszystkimi obiektami obecnie istniejącymi w grze, przez który są one dodawane

private static instance : GameObjectsManager

Statyczny i prywatny obiekt typu GameObjectsManager, do którego odwołujemy się, za każdym razem gdy wywołujemy lub tworzymy nowy GameObjectsManager

private gameobjects : Array<GameObject>

Prywatna lista wszystkich obiektów, znajdujących się obecnie w grze

public static getInstance() : GameObjectsManager

Funkcja pilnująca, aby GameObjectsManager był singletonem

public findGameObject(name : string) : GameObject

Funkcja przyjmująca zmienną typu string, która szuka, a następnie zwraca pożądaný obiekt

addGameObject(newObj : GameObject) : void

Funkcja przyjmująca obiekt typu GameObject, która dodaje utworzony GameObject do listy

### 2.2.3 Item

Klasa abstrakcyjna, na bazie której tworzone są klasy dziedziczące, których obiekty reprezentują przedmioty możliwe do zakupu w sklepie

private name : string

Prywatna zmienna typu string, przechowująca nazwę przedmiotu

private value : number

Prywatna zmienna typu liczbowego, przechowująca cenę przedmiotu

private isBought : boolean

Prywatna zmienna typu logicznego, przechowująca to czy przedmiot został zakupiony

```
private isEquipped ; boolean
```

Prywatna zmienna typu logicznego, przechowująca to czy przedmiot jest obecnie wyekwipowany

```
constructor(name : string, value : number)
```

Konstruktor klasy, przyjmujący wartości dla nazwy i ceny przedmiotu, ustawiający automatycznie, że nie został jeszcze zakupiony ani wyekwipowany, oraz wypisujący błąd jeśli cena przedmiotu jest mniejsza od zera

```
public getName() : string
```

Publiczna funkcja typu string, zwracająca nazwę przedmiotu

```
public getVal() : number
```

Publiczna funkcja typu liczbowego, zwracająca cenę przedmiotu

```
public getIsBought() : boolean
```

Publiczna funkcja typu logicznego, zwracająca to czy przedmiot został już zakupiony przez gracza

```
public getIsEquipped() : boolean
```

Publiczna funkcja typu logicznego, zwracająca to czy przedmiot jest obecnie wyekwipowany

```
public buy() : void
```

Publiczna funkcja, wywoływana w celu zakupu przedmiotu

```
public changeIsEquipped(newState : boolean) : void
```

Publiczna funkcja, odpowiedzialna za zmianę stanu wyekwipowania danego przedmiotu

## 2.3 BaseItem

Klasa dziedzicząca z klasy Item, na bazie której tworzone są obiekty przedmiotów sklepowych z kategorii Wędka, Kołowrotek i Żyłka

```
private durability : number
```

Prywatna zmienna typu liczbowego, przechowująca bazową wytrzymałość przedmiotu

```
constructor(name : string, durability : number, value : number)
```

Konstruktor klasy, przyjmujący wartości dla zmiennych name, durability i value. W przypadku podania negatywnej bazowej wytrzymałości, wyświetla błąd w konsoli.

public getDurability() : number

Publiczna zmienna typu liczbowego, zwracająca bazową wytrzymałość przedmiotu

## **3 Backend**

### **3.1 Funkcje**

### **3.2 Klasy**