

# Technická dokumentácia ku Simulátoru Závodu Plachetníc

Alena Kuklicová, Vladimír Tischler, Filip Béréš, Michal Tomášik

Katalóg požiadaviek .....	4
1. Úvod katalógu požiadaviek .....	4
1.1 Účel katalógu požiadaviek .....	4
1.2 Rozsah systému.....	4
1.3 Slovník pojmov.....	4
1.4 Referencie .....	5
1.5 Prehľad nasledujúcich kapitol .....	5
2. Všeobecný opis systému .....	5
2.1 Perspektíva systému.....	5
2.2 Funkcie systému .....	6
2.3 Charakteristika používateľov .....	6
2.4 Všeobecné obmedzenia .....	6
2.5.....	6
3. Špecifikácia požiadaviek .....	7
3.1 Lod' .....	7
3.2 Trať.....	7
3.3 Sily a javy .....	8
3.4 Kamery .....	8
3.5 Ovládanie .....	8
3.6 Vstupy a Výstupy .....	9
3.6.1 Vstupy .....	9
3.6.2 Výstupy .....	10
Návrh informačného systému.....	11
1. Úvod do návrhu systému .....	11
1.1 Účel návrhu .....	11
1.2 Rozsah a zameranie.....	11
2. Konfiguračné súbory.....	11
2.2 Výstupný súbor .....	13
3. GUI.....	14
3.1 Hlavné menu .....	14

3.2 Konfigurácia simulácie.....	14
3.3 Pohľad počas simulácie .....	15
3.4 Pohľad z ortogonálnej kamery.....	16
4.1 Triedny Diagram .....	17
4.2 Komponentný diagram .....	18
4.3 Sekvenčný diagram.....	19
5. Plán implementácie .....	20
Testovacie Scenáre .....	21

# Katalóg požiadaviek

## 1. Úvod katalógu požiadaviek

### 1.1 Účel katalógu požiadaviek

Tento dokument bol vytvorený ako katalóg požiadaviek pre systém Virtuálnej simulácie závodu plachetníc. Jeho cieľom je popísať dohodnuté požiadavky tohto systému vyvíjaného v rámci predmetu Tvorba informačných systémov FMFI UK Bratislava v akademickom roku 2024/2025.

Katalóg požiadaviek predstavuje záväznú dohodu medzi zadávateľom projektu a jeho vývojového tímu ohľadom rozsahu funkcionality.

### 1.2 Rozsah systému

Cieľom projektu je vyvinúť softvér simulácie závodu plachetníc. Softvér by mal okrem simulácie závodu, byť pripravený aj na rôzne rozšírenia. Rozšírenia, ktoré máme na mysli sú určite autonómne riadenie plachetníc pomocou nejakej konfigurácie, možnosť konfigurácie a pridávania máp. Pripraviť priestor nejakým spôsobom pre možnosť učenia nejakej neurónovej siete na vizuálnych dátach.

### 1.3 Slovník pojmov

**Framework** - štruktúra na zjednodušenie vývoja softvéru s pripravenými komponentami

**Navigačný závod** - preteky na otvorenej trase s rôznymi orientačnými bodmi. Pretekári musia navigovať medzi bodmi a prispôbovať sa meniacim sa podmienkam

**Carousel závod** - preteky na uzavretom okruhu. Obieha sa niekoľko bóji v stanovenom smere a pretekári musia byť rýchli a obratní, lebo sa neustále vracajú na rovnaké body

**Snapshot** - záznam stavu plachetnice a jej okolí, obsahujúci smer vetra, pozíciu lode, prúd vody, nastavenie kormidla a plachiet

**Autonómne riadenie** – riadenie plachetníc pomocou umelej inteligencie

**Keel** - spodná časť lode, plní vyvažovaciu funkciu

**Apparent Vietor / AW** – vietor, jeho smer a rýchlosť ako by bol s relatívnou presnosťou nameraný osobou na danej lodi (Apparent Wind)

**True Vietor / TW** – vietor, jeho skutočný smer a rýchlosť (True Wind)

**Chase Kamera** – kamera, ktorá loď “naháňa”, zhoduje sa s loďou v smerovaní a drží si od nej konštantnú vzdialenosť

## **Anemometer**

### 1.4 Referencie

GitHub repozitár projektu: <https://github.com/TIS2024-FMFI/sailing-simulator>

Repozitár podobného systému v C++: <https://github.com/mkohout/sailing-simulator>

Magisterská práca na túto tému:

<https://www.diva-portal.org/smash/get/diva2:1111560/FULLTEXT01.pdf>

Sailboat 2D - fyzikálne aspekty plachetníc:

<https://www.boatdesign.net/threads/sailboat-physics-simulation.56810/>

### 1.5 Prehľad nasledujúcich kapitol

V kapitole [2](#) všeobecne popisujeme vyvíjaný systém. Kapitola sa pozerá na systém z nadhľadu a stručne vysvetľuje jeho funkcionality, tak aby bolo jasné čo všetko bude robiť. Stručne sa zameriava tiež na charakteristiku používateľov, ako aj obmedzenia a závislosti systému.

Kapitola [3](#) jasne definuje jednotlivé požiadavky na systém, ktoré boli dohodnuté so zadávateľom projektu.

## 2. Všeobecný opis systému

### 2.1 Perspektíva systému

Vyvíjaný systém bude slúžiť ako základná kostra pre virtuálnu simuláciu závodu plachetníc. Cieľom je poskytnúť framework, ktorý umožní nielen jednoduchú simuláciu pretekov, ale aj ich postupné zdokonaľovanie v budúcnosti. V rámci systému budú zadané parametre ako sila vetra a vodné prúdy, ktoré budú simulované v 3D prostredí.

Tento systém umožní používateľovi ovládať plachetnice, nastavovať plachty a manipulovať s kormidlom. Systém bude navrhnutý tak aby bol modulárny a dal sa upravovať/vylepšovať v budúcnosti. V budúcnosti sa plánujú vylepšenia ako pridanie nových funkcií, implementácia pokročilých techník pre analýzu pretekov a vylepšenie modelov plachetníc.

## 2.2 Funkcie systému

Systém na začiatok umožní používateľovi vybrať si medzi dvoma typmi závodu a to Navigačným alebo Carousel. Následne si používateľ zvolí počet plachetníc, ktoré sa závodu zúčastnia. Po výbere typu závodu a počtu plachetníc, používateľ spustí preteky.

Používateľ je schopný prepínať sa medzi plachetnicami s cieľom ich ovládania. Keďže každá loď funguje ako samostatná entita, používateľ môže meniť parametre každej lode bez ohľadu na ostatné. Týmto parametrami budú nastavenia plachiet a kormidla.

Lode budú schopné merať apparent vietor(AW), pričom true vietor(TW) bude definovaný v systéme. Týmto spôsobom dokážu lode vizualizovať AW a následne ho prepočítať na TW. Pre túto funkciu bude na každej lodi anemometer, ktorý presne zachytí aktuálne pôsobiace veterné podmienky. Dané výpočty budú zabezpečené prostredníctvom riadiacich algoritmov a skriptov, ktoré budú súčasťou simulátora.

Každá plachetnica bude plávať po trati určenej bojkami medzi štartovacou a cieľovou líniou. To sa týka obidvoch oboch typov pretekov. V navigačnom závode je trať otvorená a pretekári musia navigovať medzi bójkami rozmiestnenými v rôznych smeroch a vzdialenostiach. Naopak v carousel závode sa plachetnice pohybujú po uzavretom okruhu, kde plachetnice musia obiehať okolo niekoľkých bójok v pevne stanovenom smere. Rozdiel medzi nimi spočíva v tom, že navigačný závod sa odohráva na väčšom území a vyžaduje navigačné zručnosti, zatiaľ čo carousel kladie dôraz na obratnosť a rýchlosť manévrovania v rámci menšieho okruhu.

Bójky sú označené šípkou, ktorá poskytuje vizuálnu informáciu o tom, či sa okolo nich majú točiť lode doprava alebo doľava. Tieto pravidlá platia pre oba typy pretekov.

Vstupom do systému bude súbor, na základe ktorého systém vygeneruje mapu. Počas pretekov budú lode pravidelne vytvárať snapshoty, ktoré budú obsahovať všetky údaje o lodi ako aj o silách ktoré na ňu pôsobia(vietor, vlny). Následne výstupom systému bude víťaz závodu a všetky záznamy o priebehu pretekov, ktoré sa následne v budúcnosti môžu použiť na analýzu a zlepšenie výkonu v budúcnosti.

## 2.3 Charakteristika používateľov

V našom softvéri bude jedna rola a to používateľ, ktorý bude ovládať aj viac plachetníc naraz. Bude mať prístup ku všetkým funkciám, ktoré bude softvér ponúkať.

## 2.4 Všeobecné obmedzenia

Simulácia je obmedzená na dva typy závodov a to Navigačným a Carousel. Súbor s mapami musia byť vo špecifickom definovanom formáte, ktorý bude definovaný pri implementácii systému.

## 2.5 Predpoklady a závislosti

Hardvér, na ktorom má byť simulácia spustená, musí byť dostatočne výkonný pre Unreal Engine 5.4.4.

### 3. Špecifikácia požiadaviek

#### 3.1 Loď

1. Loď pozostáva z :
  - a. Statického trupu
  - b. Hlavnej plachty ktorej napätie sa dá ovládať(vid' 3.5)
    - i. Vysunutie hlavnej plachty do strany vplyvom vetra sa odzrkadľuje vysunutím hlavnej plachty na modeli lode
    - ii. Napätie hlavnej plachty na modeli viditeľné nie je, je však viditeľné na vertikálnom progress bare :
      1. Progress bar je umiestnený na obrazovke po pravoboku lode
      2. Progress bar sa naplňa zdola nahor
      3. Miera 0-100% napätia plachty bude zodpovedať 1:1 miery do ktorej je progress bar naplnený
  - c. Vedľajšej plachty ktorej napätie a vysunutie sa nedajú ovládať
  - d. Kormidla, ktoré sa dá ovládať a ktorým sa dá upravovať smer lode (vid' 3.5)
  - e. Anemometra ktorého vzhľad zodpovedá vetru ktorý na loď v danej chvíli pôsobí

#### 3.2 Trať

1. Trať pozostáva z :
  - a. Štvorcovej vodnej plochy o veľkosti 5 x 5 km
  - b. Mriežky na vrchu ktorej sa bude závod konať
    - i. Políčka mriežky sú štvorcové
    - ii. Mriežka má rovnaký počet riadkov a stĺpcov
    - iii. Veľkosť mriežky a jednotlivých políčok mriežky sa prispôbi údajom v konfiguračnom súbore (vid' 3.6)
    - iv. Každé políčko nesie informáciu o hĺbke vody, smere a rýchlosti vetra a smere a rýchlosti vodných prúdov
      1. Tieto údaje sa počas závodu nemenia
  - c. Štartovacej / Koncovej čiary
    - i. Poloha tejto čiary vo svete bude daná polohou jej dvoch koncov podľa konfiguračného súboru
  - d. Bójí
    - i. Poloha každej bóje je definovaná v konfiguračnom súbore

- ii. Strana po ktorej majú lode bóju oboplávať je definovaná v konfiguračnom súbore
- iii. Strana po ktorej majú lode bóju oboplávať bude používateľovi indikovaná vznášajúcou sa šipkou nad bójou

### 3.3 Sily a javy

1. Loď má smer a rýchlosť (vektor)
2. Vietor sa opiera o hlavnú plachtu lode
  - a. Vietor má smer a rýchlosť (vektor)
  - b. Plachta má napätie (hodnota)
  - c. Vietor loď posúva dopredu ako nejaký produkt týchto dvoch vektorov
    - i. Zmena vektoru lode
  - d. Vietor vysúva plachtu
3. Kormidlo upravuje smer vektoru lode, nie však jeho veľkosť
  - a. Periodicky, s krátkou periódou updatene smer vektoru o nejakú konštantu, ktorá bude bližšie definovaná v štádiu návrhu na základe testovania
4. Odpor vody periodicky jemne loď spomaľuje, znižuje jej vektor, nemení však smer
5. Prepočty budú realizované funkciami, ktoré sa musia v budúcnosti dať modulárne vymeniť za lepšie fyzikálne aproximácie
6. Prepočty budú bližšie definované v štádiu návrhu

### 3.4 Kamery

1. Každá loď bude mať svoju Chase kameru
  - a. Používateľ bude vždy vidieť naraz Chase kameru iba jednej lode
2. Nad celým závodom sa bude pohybovať Kinematická kamera
  - a. Táto kamera sa bude pohybovať po krivke danej konfiguračným súborom, s rotáciou danou vektorom tiež definovaným v konfiguračnom súbore
3. V pravom dolnom rohu bude viditeľná mapa
  - a. Mapa bude emulovať ortogonálnu kameru
  - b. Na mape budú real-time zobrazené všetky lode
4. Možnosť dodefinovať si novú kameru v konfiguračnom súbore
  - a. Týchto kamier môže byť najviac 9
  - b. Tieto kamery majú statickú polohu a smer

### 3.5 Ovládanie

1. Ovládanie lode



- a. Používateľ vždy ovláda práve tú loď, cez ktorej Chase kameru sa pozerá
  - b. Hlavná plachta
    - i. Klávesa “W” plachtu napína
    - ii. Klávesa “S” plachtu uvoľňuje
  - c. Kormidlo
    - i. Klávesa “A” smerovanie lode upraví smerom vľavo
    - ii. Klávesa “D” smerovanie lode upraví smerom vpravo
2. Ovládanie kamier
- a. Prepínanie medzi kamerami definovanými používateľom v konfiguračnom súbore obsluhujú klávesy “1”-“9”
    - i. Počas pohľadu z jednej z týchto kamier nie je možné ovládať žiadnu loď
  - b. Klávesa “R” používateľa prepne na Chase kameru poslednej zvolenej lode
  - c. Klávesa “C” používateľa prepína medzi Cinematickou kamerou a Chase kamerou poslednej zvolenej lode
    - i. Počas pohľadu z Cinematickej kamery nie je možné ovládať žiadnu loď
3. Mapa
- a. Kliknutím ľavého tlačidla myši na loď na mape sa používateľ prepne na Chase kameru tejto lode

### 3.6 Vstupy a Výstupy

#### 3.6.1 Vstupy

1. Každá trať je daná konfiguračným súborom v nasledovnom formáte
2. Každý riadok konfiguračného súboru sa začína nižšie jedným z predom definovaných kľúčových slov podľa ktorý sa riadky pri načítaní spracovávajú
3. Vstupný konfiguračný súbor je členený na tri časti :
  - a. Prvá časť definuje trať, mriežku s poveternostnými a podvodnými podmienkami, pozície bójí a štartovnej línie
  - b. Druhá časť definuje najviac 9 používateľom definovaných kamier, nasledovaných traťou cinematickej kamery pričom táto trať :
    - i. Je definovaná postupnosťou bodou a rotácii kamery v každom bode
  - c. Tretia časť definuje rôzne nastavenia :
    - i. Frekvencia snapshotov vykonávaných loďami
    - ii. Odpor vody
4. Počet lodí v závode si používateľ bude môcť vybrať pri výbere závodu

### 3.6.2 Výstupy

1. Každý závod vytvorí adresár :
  - a. Každá loď má v rámci tohoto adresára svoj podadresár
    - i. V podadresári bude výstupný xml súbor so snapshotmi v tomto formáte:
      1. Každý riadok obsahuje čas od začiatku závodu kedy bol snapshot vyhotovený
      2. Rýchlosť a smer vetra pôsobiaceho na loď v danom momente
      3. Rýchlosť a smer vodných prúdov pôsobiacich na loď v danom momente
      4. Miera napätia a otočenia hlavnej plachty v danom momente
      5. Otočenie kormidla v danom momente
      6. Svoje súradnice v danom momente
      7. Smer lode v danom momente
    - ii. Pre každý záznam v súbore so snapshotmi bude zároveň “screenshot” z Chase kamery danej lode v danom momente
      1. Tieto snímky budú pomenované podľa času od začiatku závodu

# Návrh informačného systému

## 1. Úvod do návrhu systému

### 1.1 Účel návrhu

Tento dokument bol vytvorený ako návrh pre informačný systém Virtuálnej simulácie závodu plachetníc. Jeho cieľom je konkrétne popísať všetky funkcionality a priblížiť implementáciu tohto systému vyvíjaného v rámci predmetu Tvorba informačných systémov FMFI UK Bratislava v akademickom roku 2024/2025.

### 1.2 Rozsah a zameranie

Pre prácu s týmto dokumentom sa predpokladá určitá znalosť Katalógu požiadaviek tohto systému. Dokument špecifikuje a spresňuje všetky požiadavky spísané v Katalógu požiadaviek.

Tento dokument poskytuje podrobný plán implementácie, návrh a vizuál používateľského rozhrania a tvar vstupných a výstupných konfiguračných súborov. Na priblíženie funkcionalít a interakcií systému obsahuje komponentný UML diagram, sekvenčný UML diagram a triedny UML diagram.

## 2. Konfiguračné súbory

### 3.1 Vstupný súbor

Konfiguračný súbor obsahuje rôzne parametre, ktoré definujú nastavenia pre simuláciu plachetníc. Každý riadok predstavuje jeden objekt alebo typ dát v simulácii.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Type	GridRow	GridColumn	WindDirection	WindSpeed	WaterDirection	WaterSpeed	Depth	SpawnIndex	SpawnPosition	Angle	snapfreq	waterres	TileWidth	GridWidth	
2	GRIDDATA_1	0	0	(X=1.0,Y=0.5,Z=0.0)	10	(X=0.2,Y=0.1,Z=0.0)	2	5				10	0.1	50	3	
3	GRIDDATA_2	0	1	(X=0.8,Y=0.4,Z=0.0)	10	(X=0.3,Y=0.2,Z=0.0)	3	7	1			10	0.1	50	3	
4	GRIDDATA_3	0	2	(X=0.6,Y=0.5,Z=0.0)	10	(X=0.4,Y=0.6,Z=0.0)	5	7				10	0.1	50	3	
5	GRIDDATA_4	1	0	(X=1.0,Y=0.5,Z=0.0)	10	(X=0.2,Y=0.1,Z=0.0)	6	5				10	0.1	50	3	
6	GRIDDATA_5	1	1	(X=0.8,Y=0.4,Z=0.0)	10	(X=0.3,Y=0.2,Z=0.0)	7	7				10	0.1	50	3	
7	GRIDDATA_6	1	2	(X=0.6,Y=0.5,Z=0.0)	10	(X=0.4,Y=0.6,Z=0.0)	8	7	2	(X=205.0,Y=0.0,Z=0.0)		10	0.1	50	3	
8	GRIDDATA_7	2	0	(X=1.0,Y=0.5,Z=0.0)	10	(X=0.2,Y=0.1,Z=0.0)	6	5				10	0.1	50	3	
9	GRIDDATA_8	2	1	(X=0.8,Y=0.4,Z=0.0)	10	(X=0.3,Y=0.2,Z=0.0)	2	7				10	0.1	50	3	
10	GRIDDATA_9	2	2	(X=0.6,Y=0.5,Z=0.0)	10	(X=0.4,Y=0.6,Z=0.0)	3	7	3			10	0.1	50	3	
11	START															
12	USERCAMERA_1								4	(X=1.0,Y=1.0,Z=1.0)	(X=1.0,Y=-90,Z=1.0)					
13	USERCAMERA_2								4	(X=1.0,Y=1.0,Z=1.0)	(X=1.0,Y=-90,Z=1.0)					
14	CINEMATICCAMERA_1								5	(X=-120.0,Y=-120.0,Z=1.0)	(X=1.0,Y=1.0,Z=1.0)					
15	CINEMATICCAMERA_2								5	(X=120.0,Y=120.0,Z=1.0)	(X=1.0,Y=1.0,Z=1.0)					
16	CINEMATICCAMERA_3								5	(X=0.0,Y=0.0,Z=1.0)	(X=1.0,Y=1.0,Z=1.0)					
17																

(Type) – Určuje, o aký objekt ide. Ak je to políčko v gride, označuje sa ako GRIDDATA\_(číslo). START slúži na oddelenie GRIDDATA a kamier v kóde. USERCAMERA nastavuje pozície a rotácie kamery na každej lodi.

CINEMATICCAMERA určuje pozície, medzi ktorými sa kamera pohybuje po mape.

(GridRow) – Určuje riadok v 2D gride políček.

(GridColumn) – Určuje stĺpec v 2D gride políček.

(WindDirection) – Určuje smer, ktorým fúka vietor na danom políčku.

(WindSpeed) – Určuje rýchlosť vetra na danom políčku.

(WaterDirection) – Určuje smer vodného prúdu na danom políčku.

(WaterSpeed) – Určuje rýchlosť vodného prúdu na danom políčku.

(Depth) – Určuje hĺbku vody na danom políčku.

(SpawnIndex) – Podľa daného indexu spawnuje jednotlivé objekty:

1 = označuje štartovaciu/cieľovú čiaru, na ktorej sa spawnujú všetky lode,

2 = označuje bojku,

3 = označuje cieľovú čiaru, kde sa hráči nespawnujú,

4 = označuje USERCAMERA,

5 = označuje CINEMATICCAMERA.

(SpawnPosition) – Určuje pozíciu spawnovaného objektu, týka sa to iba kamier.

(Angle) – Určuje rotáciu objektu v Unreal Engine. Pri bojke to určuje smer šípky, ktorá je nad bojkou.

(snapfreq) – Určuje frekvenciu vytvárania snapshotov.

(waterres) – Určuje odpor vody na danom políčku v gride.

(TileWidth) – Určuje šírku a dĺžku jedného políčka.

(GridWidth) – Určuje veľkosť gridu, napríklad 3x3.

## 2.2 Výstupný súbor

Výstupom systému sú snímky (snapshots), ktoré sú uložené v priečinku "Snapshots" v hlavnom priečinku "Saved" v Unreal engine. Pre každú loď je vytvorený následne samostatný priečinok pomenovaný podľa názvu lode, do ktorého sa ukladajú jednotlivé snímky pre každú loď. Tieto snímky sú typu .png a ich názov obsahuje všetky potrebné informácie. Jednotlivé informácie sú oddelené “\_”:

```
BPReggatta0_3_(0-0-40)_10_(X1,Y0.5,Z0)_4_(X0.3,Y0,Z0)_100_0_-20.444_(X2,530.94,Y2,473.67,Z10)_(X-0.26,Y-0.97,Z0).png
```

(BPReggatta0) – názov lode, ktorý jednoznačne identifikuje danú loď. Tiež je to názov priečinku v ktorom sa nachádzajú všetky snapshoty tejto konkrétnej lode.

(3) – číslo snapshotu, ktoré označuje poradové číslo snímky pre danú loď (snímky sa robia každých 10 sekúnd).

(0-0-40) – čas vo formáte (hodiny-minúty-sekundy). Označuje čas kedy bol snapshot spravený a taktiež označuje koľko času prešlo od začiatku závodu.

(10) – rýchlosť vetra pôsobiaceho na loď v danom momente.

(X1,Y0.5,Z0) - smer vetra pôsobiaceho na loď v danom momente.

(4) – rýchlosť vodného prúdu pôsobiaceho na loď v danom momente.

(X0,3,Y0,Z0) - smer vodného prúdu pôsobiaceho na loď v danom momente.

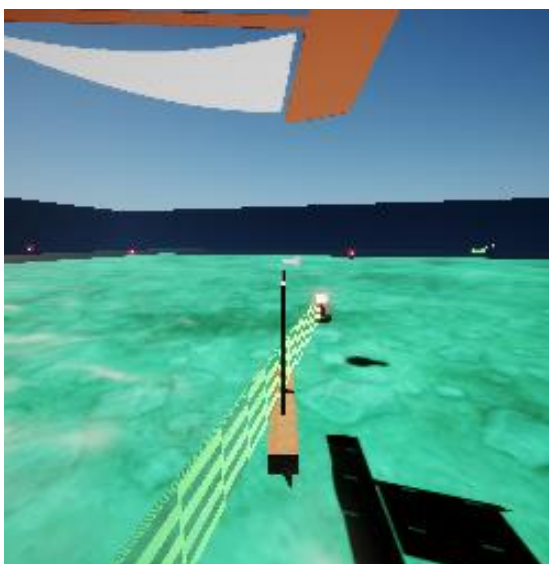
(100) – miera napätia hlavnej plachty v danom momente.

(0) – miera otočenia hlavnej plachty v danom momente.

(-20.444) – miera otočenie kormidla v danom momente.

(X2530.94,Y2473.67,Z10) – súradnice lode v danom momente.

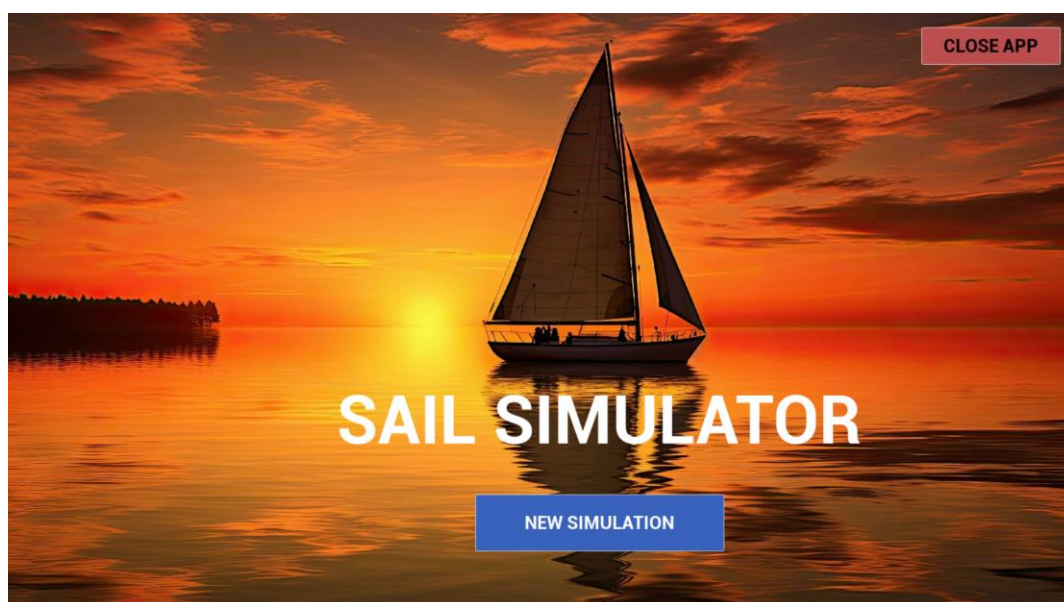
(X-0.26,Y-0.96,Z0) – smer lode v danom momente .



### 3. GUI

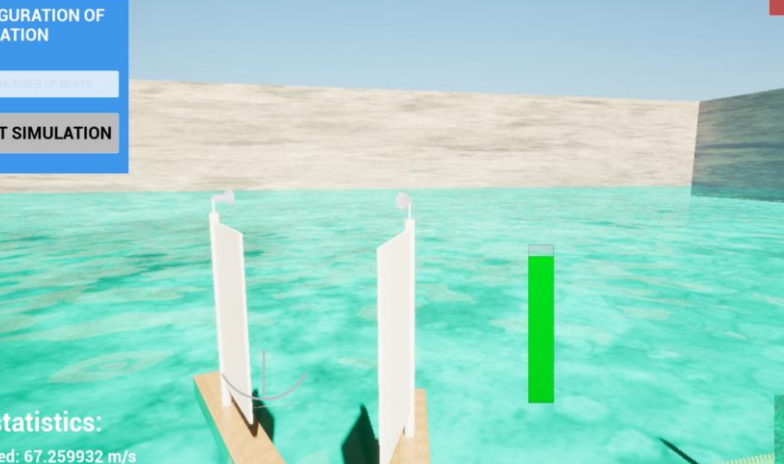
#### 3.1 Hlavné menu

V tejto časti si používateľ môže vybrať, či chce začať novú simuláciu alebo chce zavrieť aplikáciu.



#### 3.2 Konfigurácia simulácie

V tejto časti používateľ už vstupuje do simulácie. Predpoklad je, že už vložil konfiguračný súbor do predvoleného priečinku v predvolenom formáte. Po vložení



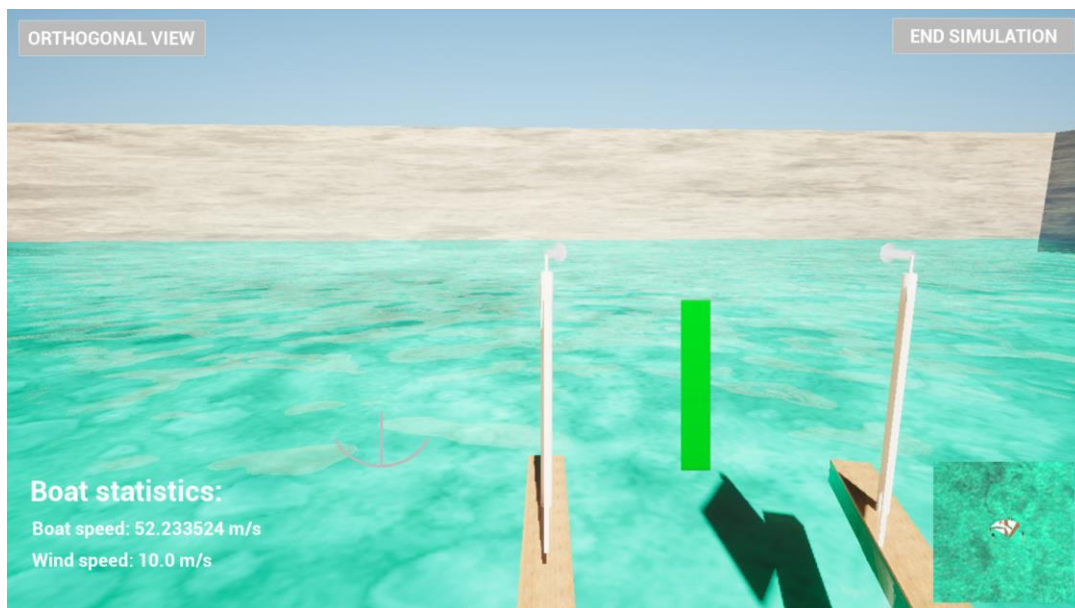
**Boat statistics:**

Parameter	Value
Boat speed:	67.259932 m/s
Wind speed:	0.0 m/s

Close

V tejto časti prebieha už samotná simulácia. Používateľ môže ovládať samotnú plachetnicu. Na plachetnici môže ovládať smer do akého chce ísť (vpravo/vľavo) a napnutie hlavnej plachty. Úroveň napnutia plachty vidíme ako progress bar z pravej strany plachetnice. V ľavom dolnom rohu vidí globálne premenné, ktoré potrebuje vedieť počas ovládania plachetnice ako je rýchlosť lode, smer vetra a rýchlosť vetra. V pravom dolnom rohu je zobrazená mapa kde sú zobrazené všetky lode. Z tejto časti vieme ukončiť simuláciu alebo prejsť do ortogonálneho pohľadu.





### 3.4 Pohľad z ortogonálnej kamery

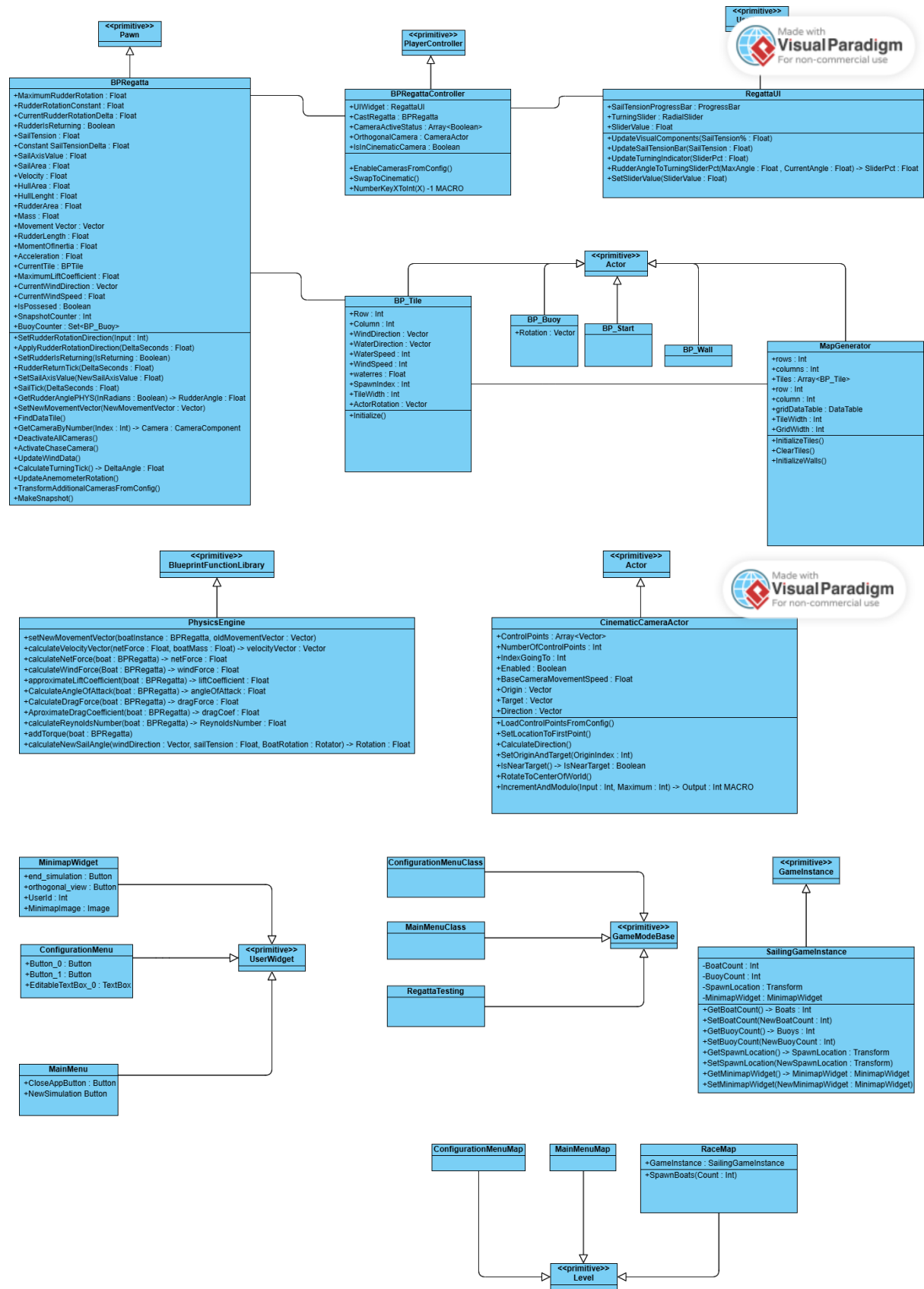
V tejto časti sa prepíname medzi plachetnicami, ktoré chceme ovládať. Výber plachetnice je riešený kliknutím, na konkrétnu plachetnicu.





## 4. UML Diagramy

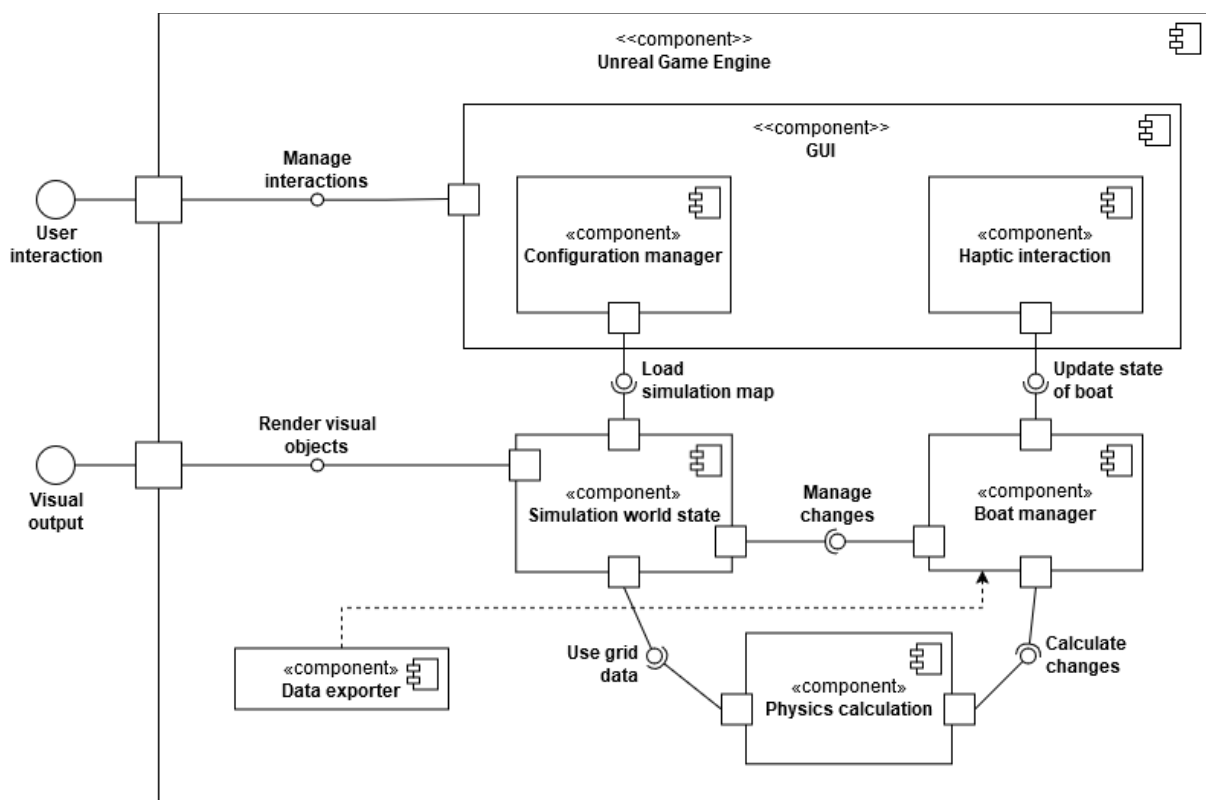
### 4.1 Triedny Diagram



Triedny diagram obsahuje prehľad tried ktoré sme vytvorili a používali v rámci tohoto systému. Triedy označené <<primitive>> sú základné triedy v Unreal Engine a ich atribúty a metódy možno nájsť v oficiálnej dokumentácii. Pre účely poskytnutia základnej predstavy o týchto triedach, **Actor** je objekt umiestnený v scéne, **Pawn** je druh **Actora** ktorý môže byť ovládaný používateľom alebo AI. **PlayerController** je trieda ktorá umožňuje ovládanie **Pawna** používateľom. **UserWidget** je trieda ktorá umožňuje rozširovať UI. **BlueprintFunctionLibrary** je knižnica funkcií, **Level** je scéna v ktorej môžu byť **Actors**, a na každý **Level** sa viaže nejaký **GameMode** ktorý definuje pravidlá hry. **GameInstance** je singleton objekt ktorý existuje počas celého trvania hry. V našom systéme je použitý ako úschovňa globálnych premenných. Triedy, ktoré sú odvodené od **GameModeBase**, sú triedy dátové a dedenie je v tomto prípade využívané výlučne na overrideovanie členských premenných **GameModeBase**.

## 4.2 Komponentný diagram

Komponentný diagram identifikuje najkľúčovejšie komponenty systému, zobrazuje ich vzájomné vzťahy a spôsoby akými komunikujú.



### 4.2.1 Fyzikálny komponent a návrh fyzikálnych výpočtov

Fyzikálny komponent je plný výpočtových funkcií na výpočet zmeny pohybu lode v čase a za vplyvu vonkajších faktorov. Komponent je volaný komponentom Boat manager s potrebnými informáciami lode a jej prostredia.

V čase návrhu vznikol dokument Návrh fyzikálnych výpočtov, ktorého hlavným účelom bolo odhaliť dôležité parametre, ktoré by si loď mala o sebe pamätať a boli by meniteľné podľa potreby v budúcnosti. Samotné fyzikálne funkcie a gettery fyzikálnych

premenných lode sú implementované, avšak sme pre naše účely zvolili jednoduchšie výpočty, ktoré ich nevyužívajú všetky. V nasledujúcich odsekoch sa Vám pokúsime priblížiť implementáciu za otáčaním plachty a zmeny pohybu lode.

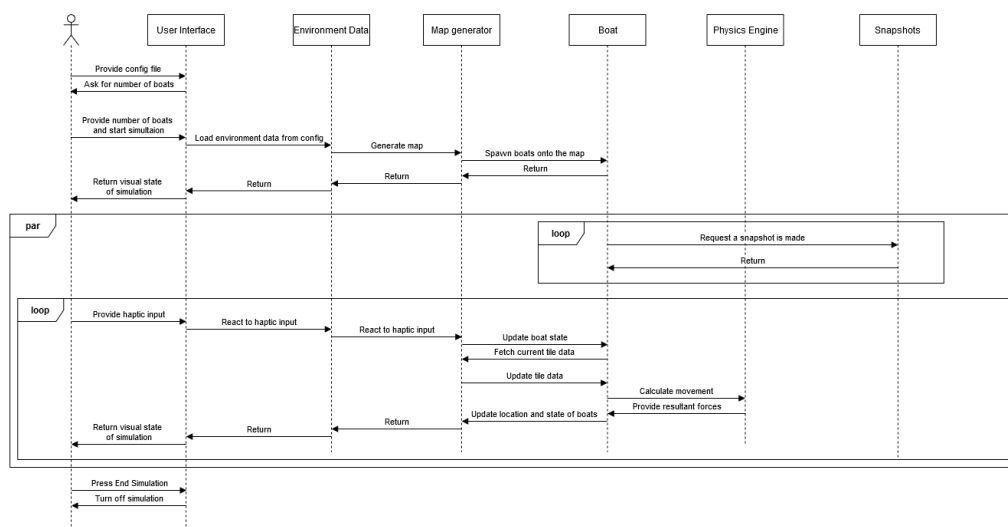
Plachta lode sa otáča na základe vetra a jej napätia. Úplne napätá plachta zostáva v strede, začiatočnej pozícii, bez ohľadu na smer vetra. Úplne uvoľnená plachta je naopak otočená práve tak ako na loď vietor fúka. Postupné uvoľňovanie plachty teda plachtu otvára viac a viac a zvyšuje tým vplyv vetra, ktorý na plachtu má.

Loď má svoj vektor pohybu, na ktorý pôsobí vietor, loď samotná, kormidlo a odpor vody. Pre jednoduchosť ovládania simulácie sa loď sama posúva dopredu, čo môže v budúcnosti byť nahradené inými výpočtami pohybu. Loď sa však neposúva iba dopredu, nakoľko ju posúva aj vietor v smere, v ktorom fúka. Veľkosť vplyvu vetra na pohyb lode sa dá zväčšiť, respektíve zmenšiť. Čím je uvoľnenie plachty bližšie k 50 percentám, tým väčší vplyv vietor na loď má. Dôvod je za tým taký, že vtedy by plachta mala byť otočená kolmo na vektor vetra, ktorý na ňu pôsobí. Úplne uvoľnená, respektíve napätá, plachta spôsobuje, že na loď nevplyva vektor vetra skoro vôbec.

Kormidlo pomaly loď otáča okolo svojej osi avšak pri jeho používaní sa vektor pohybu lode zmenší a to kvôli odporu vody.

### 4.3 Sekvenčný diagram

Sekvenčný diagram, znázorňuje interakcie medzi rôznymi entitami v systéme v priebehu času. Zobrazuje, ako používateľské vstupy (napríklad riadenie plachty alebo kormidla) prechádzajú systémom a ako jednotlivé komponenty spolupracujú na dosiahnutí požadovaného výsledku. Zobrazuje zároveň ako paralelne počas reagovania na používateľské vstupy simulátor pravidelne vytvára snapshoty čím vytvára dáta výstupu.



## 5. Plán implementácie

### 1. Implementácia lode

- 1.1. Kormidlo - jeho vizuál a ovládanie
- 1.2. Sťažnosť - vizuál
- 1.3. Plachta - ovládanie
- 1.4. Pohyb na základe používateľského vstupu
- 1.5. Kamery lode a prepínanie medzi nimi
- 1.6. Zapisovanie stavu lode do súboru

### 2. Mapa

- 2.1. Čítanie dát zo súboru
- 2.2. Vygenerovanie mapy
  - 2.2.1. Štart, a prípadne koniec, závodu
  - 2.2.2. Voda, hĺbka vody
  - 2.2.3. Pevnina
  - 2.2.4. Prúdy - vodné a veterné
  - 2.2.5. Lode
  - 2.2.6. Bóje

### 3. GUI

- 3.1. Hlavné menu
- 3.2. Konfiguračné menu
- 3.3. Ortogonálna kamera
- 3.4. Vizualizácia premenných pre konkrétne lode
- 3.5. Minimapa

### 4. Fyzika

- 4.1. Spracovanie dát objektov na mape
- 4.2. Spracovanie dát prostredia
- 4.3. Implementácia realistických fyzikálnych výpočtov
  - 4.3.1. Výpočet rýchlosti a smeru lode na základe jej stavu a stavu prostredia

## Testovacie Scenáre

### T.S.1 : Spustenie a priebeh závodu

- Predpoklady :
  - Používateľ poskytol validný konfiguračný súbor.
  - Používateľ sa nachádza v hlavnom menu.
  
- Postup :
  - Používateľ klikne na tlačidlo “New Simulation”.
    - ♣ *Očakávaný výsledok* : Používateľ je presunutý na obrazovku kde môže zadať počet lodí ktoré sa majú zúčastniť závodu.
  
  - Používateľ zadá požadovaný počet lodí do textového poľa a klikne na tlačidlo “Start Simulation”.
    - ♣ *Očakávaný výsledok* : Používateľ je presunutý do závodu v ktorom sa nachádza nim určený počet lodí a on ovláda jednu z nich. Ďalej sú v zápode korektne načítané políčka s dátami, bójky, štartovná čiara a cinematická kamera.

- o Používateľ chce kormidlom upraviť smer lode vľavo a stlačí klávesu “A”.
  - ♣ *Očakávaný výsledok* : Kormidlo sa vychýli správnym smerom a loď upraví svoj smer smerom vľavo. (3.5.1.c.i)
  
- o Používateľ chce kormidlom upraviť smer lode vpravo stlačí klávesu “D”.
  - ♣ *Očakávaný výsledok* : Kormidlo sa vychýli správnym smerom a loď upraví svoj smer smerom vpravo. (3.5.1.c.ii)
  
- o Používateľ chce uvoľniť plachtu a stlačí klávesu “S”.
  - ♣ *Očakávaný výsledok* : Plachta sa vychýli viac do strany a z progress baru po pravej strane lode ubudne.
  
- o Používateľ chce napnúť plachtu a stlačí klávesu “W”.
  - ♣ *Očakávaný výsledok* : Plachta sa pritiahne bližšie ku osi trupu lode a do progress baru po pravej strane lode pribudne.
  
- o Používateľ chce prepnúť na inú kameru na pravé ovládanej lodi a stlačí klávesu 1-9.
  - ♣ *Očakávaný výsledok* : Ak bola kamera s daným číslom definovaná v konfiguračnom súbore, prepne sa pohľad na ňu. Ak nie, nestane sa nič.
  
- o Používateľ chce prepnúť naspať na chase kameru práve ovládanej lodi a stlačí klávesu “R”
  - ♣ *Očakávaný výsledok* : Po stlačení klávesy je pohľad opäť z pohľadu chase kamery.
  
- o Používateľ sa chce pozrieť na závod z pohľadu cinematickej kamery a stlačí klávesu “C”. Potom sa chce prepnúť opäť na chase kameru poslednej ovládanej lode a opäť stlačí klávesu “C”.
  - ♣ *Očakávaný výsledok* : Po prvom stlačení sa pohľad na simuláciu zmení na pohľad z cinematickej kamery. Po druhom stlačení sa pohľad vráti na chase kameru poslednej ovládanej lode.

- o Používateľ počas simulácie vidí vplyv síl a javov, ktoré pôsobia na loď a chce upravením kormidla zmeniť smer vektoru lode.
  - ♣ *Očakávaný výsledok* : Loď reaguje na vonkajšie sily. Smer vetra je znázornený anemometrom a otočením plachty v závislosti s jej napätím, ktoré prispieva aj k zmene smeru lode. Otočením kormidla sa smer vektoru lode zmení a vplyvom odporu vody sa zmení jeho veľkosť.
  
- o Používateľ sa po ukončení závodu snaží v “Saved” adresári nájsť snapshoty utriedené do podadresárov podľa lode ktorá ich vytvorila.
  - ♣ *Očakávaný výsledok* : V “Saved” sa nachádzajú podadresáre, v ktorých sa nachádzajú snapshoty v podobe fotografie, ktorá v názve nesie dáta vzťahujúce sa na daný snapshot.