

# Ugradbeni sistemi

Dokumentacija implementacije

Tema: Space Impact

Članovi tima:

Agić Ahmed

Zukić Muhamed

Juli 2021

Dokumentacija će prvo posmatrati klase koje su neizostavne za funkcionisanje projekta, a potom će pojasniti ostale funkcije koje učestvuju u toku igre. Za početak klasa [Artillery](#) je model klase za artiljeriju (naoružanje) te je bazna klasa za klase [CannonBall](#), [Projectile](#), a u nastavku ćemo prikazati listu metoda [Artillery](#):

1. [Artillery\(int power,int speed,Translation translation\)](#)  
Konstruktor koji prima kao parametre vrijednosti snage, brzine, smjera translacije respektivno.
2. [virtual void move\(\)=0](#)  
Osigurava pomjeranje metka u prostoru za nasljeđene klase.
3. [virtual void show\(\)const=0](#)  
Osigurava prikaz metka u prostoru za nasljeđene klase.
4. [virtual void unshow\(\)const=0;](#)  
Osigurava sakrivanje metka u prostoru za nasljeđene klase.
5. [virtual int getX\(\)const=0](#)  
Vraća centar metka X kordinate.
6. [virtual void getY\(\)const=0](#)  
Vraća centar metka Y kordinate.
7. [int getPower\(\)const=0](#)  
Vraća snagu metka.
8. [Translation getDirection\(\)const=0](#)  
Vraća smjer, direkciju kretanja metka.

Sljedeća značajna apstraktna klasa je [BattleShip](#) koja opisuje brod koji se kreće u prostoru, tako da su njegove metode izložene u nastavku:

1. [BattleShip\(int h,int s,int fp, Point c\)](#)  
Konstruktor koji prima kao parametre vrijednosti života, brzine, pucačke snage, pozicije respektivno.
2. [Point getCornerUL\(\)const](#)  
Vraća gornju-lijevu tačku broda.
3. [Point getCornerLD\(\)const](#)  
Vraća donju-desnu tačku broda.
4. [int getHealth\(\)const](#)  
Vraća broj života broda.
5. [virtual void fire\(\)](#)  
Implementira artiljeriju i prikazuje na displeju.
6. [bool reduceHealth\(int decrease\)](#)  
Dekrementira život parametra za vrijednost parametra.

7. **virtual void show()**  
Prikazuje brod.
8. **virtual void unshow()**  
Sakriva brod.
9. **virtual int move(Translation direction)**  
Pomjera brod u smjeru zadanog parametra, a vraća 1 ukoliko je kretanje u dozvoljenom okviru na displeju, a -1 ukoliko je izvan okvira brod.
10. **virtual bool isItShot(Point test)**  
Provjerava da li je brod pogođen, ukoliko jeste vraća true, ukoliko nije vraća false.
11. **virtual int move(Translation direction)**  
Pomjera brod u smjeru zadanog parametra, a vraća 1 ukoliko je kretanje u dozvoljenom okviru na displeju, a -1 ukoliko je izvan okvira brod.

Nasljedene klase iz bazne klase [BattleShip](#) su [AllyShip](#), [EasyEnemyShip](#), [MediumEnemyShip](#), [HardEnemyShip](#) i [Boss](#). Svaka od navedenih nasljedjenih klasa implementira svoj izgled, način pucanja, izbor artiljerije i smjer kretanja. U nastavku ćemo opisati ostale funkcije u programu:

1. **void moveEnemyShips()**  
Pomjera neprijateljske brodove.
2. **void removeEnemyShip(int i)**  
Brisanje broda u vektoru neprijateljskih brodova koji je na poziciji zadanog parametra.
3. **void spawnEnemyShips()**  
Stvara neprijateljske brodove i zavisno od količine brodova pravi smjenu tipova neprijateljskih brodova, uz uslov da postoji mjesta na ekranu za njih (zbog ograničenosti simulatora u pogledu brzine kretanja maksimalni broj mjesta je 6).
4. **void getMainFrame()**  
Prikaz početnog displeja.
5. **void resetScore()**  
Ažurira rekord.
6. **void updateArtillery()**  
Funkcija za ažuriranje kretanja metaka i provjerava pogođenosti brodova. Implementirana je pomoću mape gdje je primarni ključ Y koordinata, a sekundarni ključ je lista u kojoj se nalaze sve vrste metaka za Y osu primarnog ključa.
7. **void randomFire()**  
Poziva pucanje nasumičnog neprijateljskog broda. U ovom slučaju [EasyEnemyShip](#) ne puca, dok ostali neprijateljski brodovi pucaju.
8. **void turnOnTickers()**  
Uključuje tikere za ažuriranje metaka, pomjeranje brodova, pojavljivanje brodova.
9. **void shutDownTickers()**  
Isključuje tikere za ažuriranje metaka, pomjeranje brodova, pojavljivanje brodova.

10. `void startGame()`  
Započinje igru.
11. `void pauseGame()`  
Pauziranje i nastavljjanje igre nakon pauze.
12. `bool clickPlay(int x1, int y1)`  
Provjerava da li je pritisnuto Play.
13. `bool clickUp(int x1, int y1)`  
Provjera da li se desio klik za pomjeranje broda prema gore.
14. `clickDown(int x1, int y1)`  
Provjera da li se desio klik za pomjeranje broda prema dolje.
15. `clickFire(int x1, int y1)`  
Provjera da li se desio klik za pucanje iz Alijansnog broda.
16. `bool clickPause(int x1, int y1)`  
Provjera da li se desio klik za poziv pauze.
17. `void deleteAllShips()`  
Brisanje svih brodova u igri.
18. `void deleteAllArtillery()`  
Brisanje svih artiljerija u igri.
19. `void showGameOverFrame()`  
Prikaz kraja igre i rekorda igrača.
20. `void intro()`  
Incijalizacija i prikaz početnog zaslona.
21. `void endGame()`  
Prvim pozivom gašenje elemenata igre, drugim pozivom resetovanje postavki za ponovno igranje.
22. `bool clickPlayAgain(int x1, int y1)`  
Provjera da li se desio klik za ponovno igranje.

Main funkcija vrši provjeru da li je došlo do dodira displeja, te provjerava u koju fazi se igra nalazi da može shodno fazi djelovati.

Posebna napomena, mjere za pozivanje tikera su stavljene u skladu sa simulatorom, kao i ograničenje broja metaka u jednoj Y osi zbog ograničenja simulatora tj. da bismo izbjegli gubitke u vidu brzine i konzistentnosti same igrice.