Battleship имплементација:
Структура на кодот и логика

Ана Андовска 221126
GitHub link: https://github.com/andovskaana/BattleShips_PNVI/

Апстракт

Овој документ обезбедува детален преглед на имплементацијата на модифицирана верзија на играта Battleship (битка со бродови), објаснувајќи ја нејзината структура, механика и уникатни карактеристики што ја разликуваат од стандардната верзија. Документот ги опишува главните компоненти на играта, вклучувајќи ги поврзаноста на датотеките, елементите на корисничкиот интерфејс, однесувањето на компјутерот противник (AI со стратегија на веројатност), нивоата на тежина, механика на маглата и функцијата за совет за помош (hints). Дополнително, се објаснуваат стратегиите зад таргетирањето и поставувањето на бродови од страна на комјутерот, функцијата на магла, и функционалностите на корисничкиот интерфејс, како што се: *fullscreen* режим и анимациите, кои го подобруваат корисничкото искуство.

1. Главни екрани на играта нивните датотеки

Во оваа игра главни екрани се:

- Екран за анимација за приказ на името на креаторот на играта
- Екран за главно мени
- Екран за мени за избор на ниво и тежина
- Екран каде се одвива логиката и играњето на играта

Датотеките на проектот се:

- mкомпјутерп.py
- menu.py
- game.py
- config.py

1.1 main.py – Влезна точка на играта е одговорна за иницијализирање на прозорецот на играта и стартување на менито. Оваа датотека не прикажува директно екран, но ги повикува сите потребни методи од останатите датотеки.

1.2. menu.py – Оваа датотека ги прикажува екраните со мени: почетното мени и менито за избор на ниво на играта. Од овде се праќа параметарот за избраното ниво на играта кон датотеката game.py.

1.3. game.py – Во оваа датотека се одвива целата логика на играта и во повеќе детали е објаснета во следните глави. Таа управува со двете фази на играта: поставување на бродови и фаза на борба.

1.4. config.py – Ги содржи сите глобални променливи кои се користат во другите датотеки и за изгледот, големината на екранот, палетата на бои и фонтот.

Првиот екран прикажува анимација на името на креаторот со цел попрофесионален изглед – повикување на *fade-in* анимација (сл. 1).

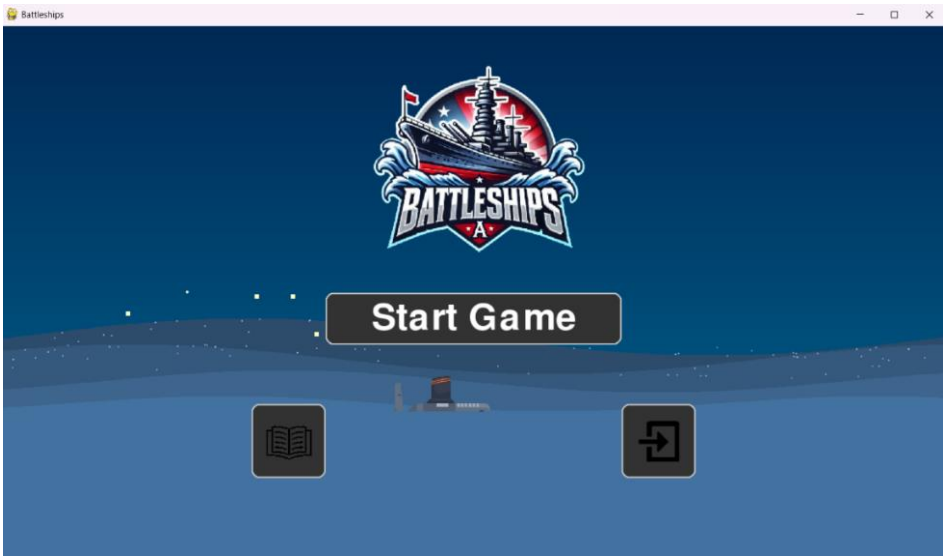
Сл. 1. Анимација за креатор

1.1. Рендерирање на почетно мени

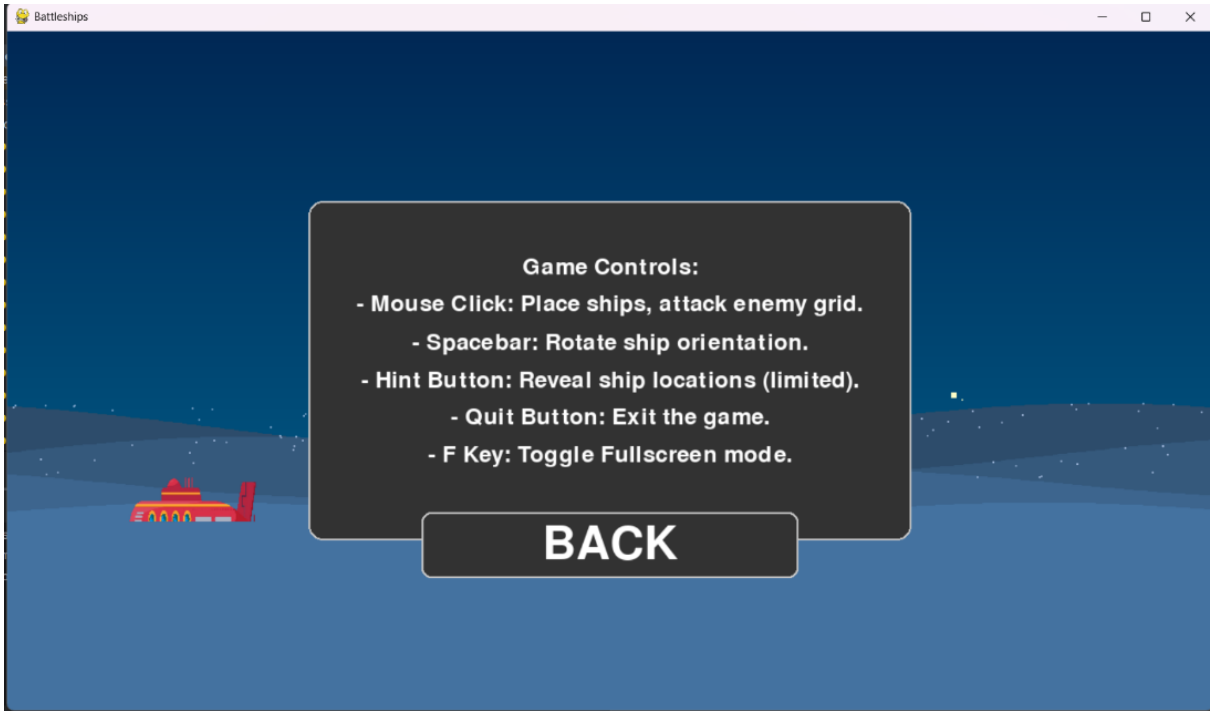
- Копчињата се создадени преку класата *Button*. Тука, на првото мени се копчињата за почеток, контрола на играта (сл. 2), копче за приказ на екран со инструкции (сл. 3) како и копче за гасење на играта.
- **Слоевита анимација на бранови и движење на бродови**

Позадината на главното мени прикажува анимирана океанска сцена со динамично движење на бранови и движење на бродови (сл. 3). Овој ефект е постигнат преку математичка симулација на бранови, комбинирана со движење на спрајтови за бродовите. Тоа е постигнато само со помош на python програмскиот јазик и pygame библиотеката. Подетално објаснување за имплементацијата на анимацијата ќе биде дадена во подоцнежните глави во документ.

- Логото е направено специфично за овој проект. Воочливо е поставена иницијалата буква од на моето име „А“. (сл. 2)



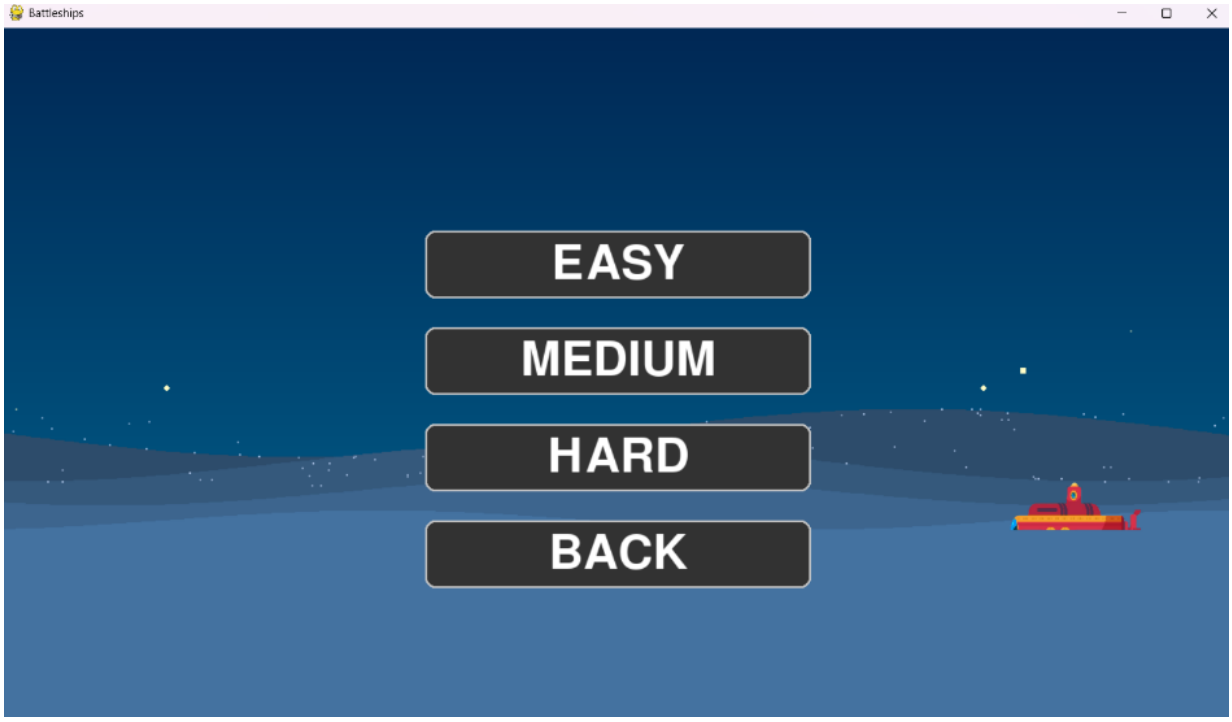
Сл. 2. Почетно мени



Сл. 3. Екран со инструкции

1.2. Рендерирање на мени за избор на ниво со соодветна тежина

- Како надополнување на претходното мени продолжува анимацијата во позадина заедно со копчињата за избор на различни режими на играње, според нивото на тежина: лесно, средно или тешко ниво. (сл. 4).



Сл. 4. Мени за иизбор на тежина на ниво

Нивоа на тежина

1. **Лесен режим:**
    - Компјутерот **ги памети претходните пукања** и се прилагодува.
    - Компјутерот нема активирано повеќе пукање на потег**.**
    - Нема магла; погодоците и промашувањата остануваат видливи.
    - Достапни се **ограничен број на совети за помош (hints)**.
2. **Среден режим:**
    - Компјутерот **ги памети претходните пукања** и се прилагодува.
    - Активирана е **маглата**, криејќи ја историјата на нападите.
    - Играчот мора **стратешки** да се потпира на неговата меморија за тоа што се случило со претходните негови потези.
3. **Тежок режим:**
    - Компјутерот пука **до 3 пати по потег**, приоритетно таргетирајќи **области со висока веројатност**.
    - Користи **напредно таргетирање, базирано на веројатност**.
    - **Густата магла** го отежнува следењето на позициите на непријателските бродови.

1.3. Фази на играта, game.py

1.3.1. Фази на играта

Фаза на поставување бродови

- Играчот поставува бродови, рачно, со преглед во реално време и ротација.
- Невалидни поставувања (преклопувања, надвор од таблата) се автоматски блокирани.
- Компјутерот ги поставува бродови користејќи алгоритам за распоред.

Фаза на борба

- Напади по редослед помеѓу играчот и компјутерот.
- Визуелни индикатори за погодок (црвени кругови) и промашување (сиви кругови).
- Маглата покрива делови на таблата во средното и тешкото ниво.

1.3.2. Објаснување на датотеката game.py

Основна механика на играта: Управува со поставување на бродови, борба, и рендерирање.

1.3.2.1.. Клучни класи:

- **Ship**: Содржи атрибути на бродовите (големина, ориентација) и вчитува спрајтови. Ја користи *get_sprite()* за враќање на скалирани слики врз основа на оштетувањето од погодоците во полињата.
- **GameState**: Ги следи таблите (*player_board, ai_board*), погодоците, анимациите, маглата и резултатите. Го управува поставувањето на бродовите на комјутерот преку *place_ai_ships().*
- **Animation**: Управува со ажурирање и прикажување на експлозии и промашувања.

1.3.2.2. Функции:

- *menu_game():* Главна контролна функција која управува со фазите на игра (подготовка, играње и крај), интеракција и рендерирање.
- *ai_turn():* Ја имплементира логиката на компјутерските пукања врз основа на тежината.
- *draw_grid():* Ги рендерира таблите на играчот и компјутерот со бродови, погодените позиции и маглата.

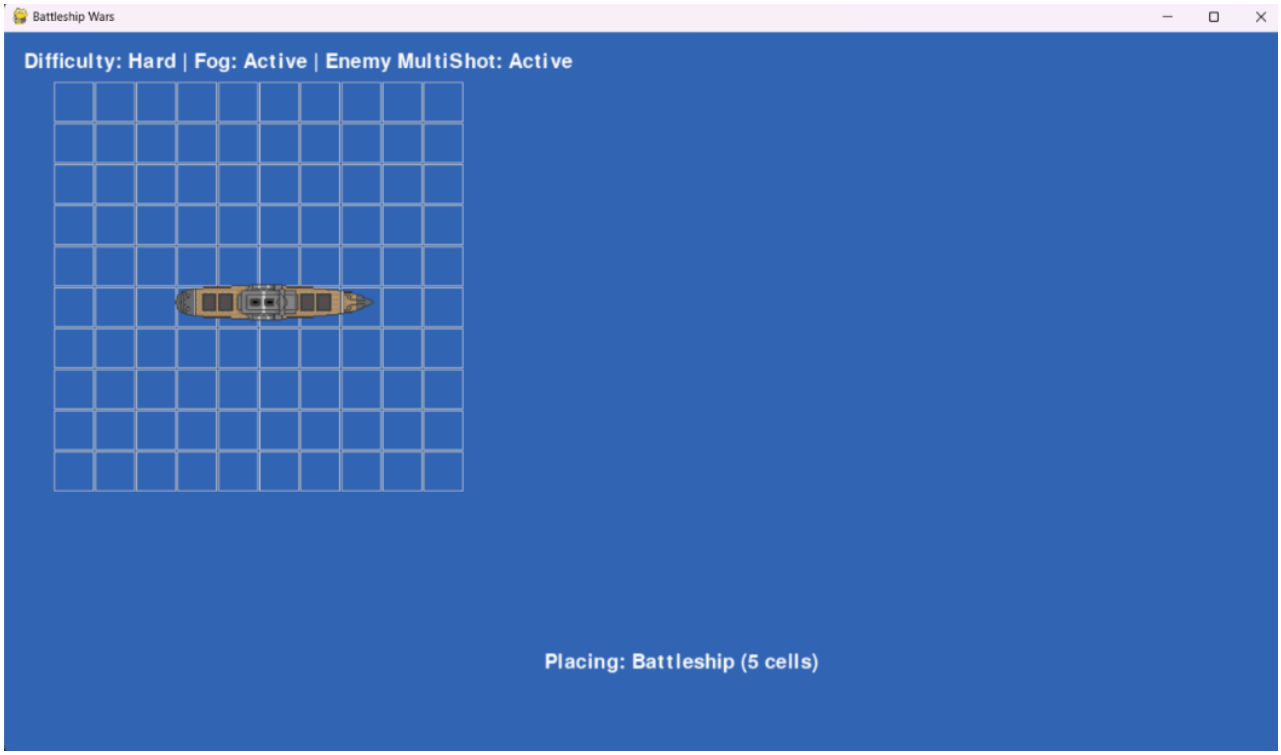
1.3.3. Објаснување на датотеката config.py

Во оваа датотека се поставени сите глобални променливи кои се константни и потребни на сите други датотеки. Вакви променливи се ширината и должината на екранот, боите со нивните rgb вредности како и фонтот.

2. Поставување на бродовите

2.1. Поставување на бродовите од страна на играчот

При поставување на бродовите, самата слика што е репрезентација на бродовите се движи со движењето на глушецот се додека не се притисне на истата и да се фиксира нејзината позиција (сл. 5). Ако позицијата е невалидна (надвор од рамката или врз друг брод) тогаш бродот воопшто не се прикажува.



Сл. 5. Поставување на бродови од играчот

2.2. Поставување на бродови од компјутерот и неговиот систем за правична дистрибуција

Системот за поставување на бродови на компјутерот следи структурирана **случајна валидација** за да се обезбеди фер распределба. Механизмот на поставување на компјутерот се состои од:

2.2.1 Случаен избор на позиција

- Компјутерот избира **случаен почетен ред и колона**.
- Случајно избира **ориентација** (H за хоризонтално, V за вертикално).

2.2.2. Проверка на граници и преклопувања

- Системот проверува **гранични ограничувања**:
    - Ако orientation == H, се осигурува дека col + ship_size <= GRID_SIZE.
    - Ако orientation == V, се осигурува дека row + ship_size <= GRID_SIZE.
- Системот спроведува **проверка на преклопувања**:
    - **Ги поминува сите полиња кои бродот ги зафаќа** и проверува дали тие се веќе зафатени.
    - Ако се открие преклопување, повторно се обидува со нови случајни вредности.

2.2.3. Стратегија за спречување на групирање

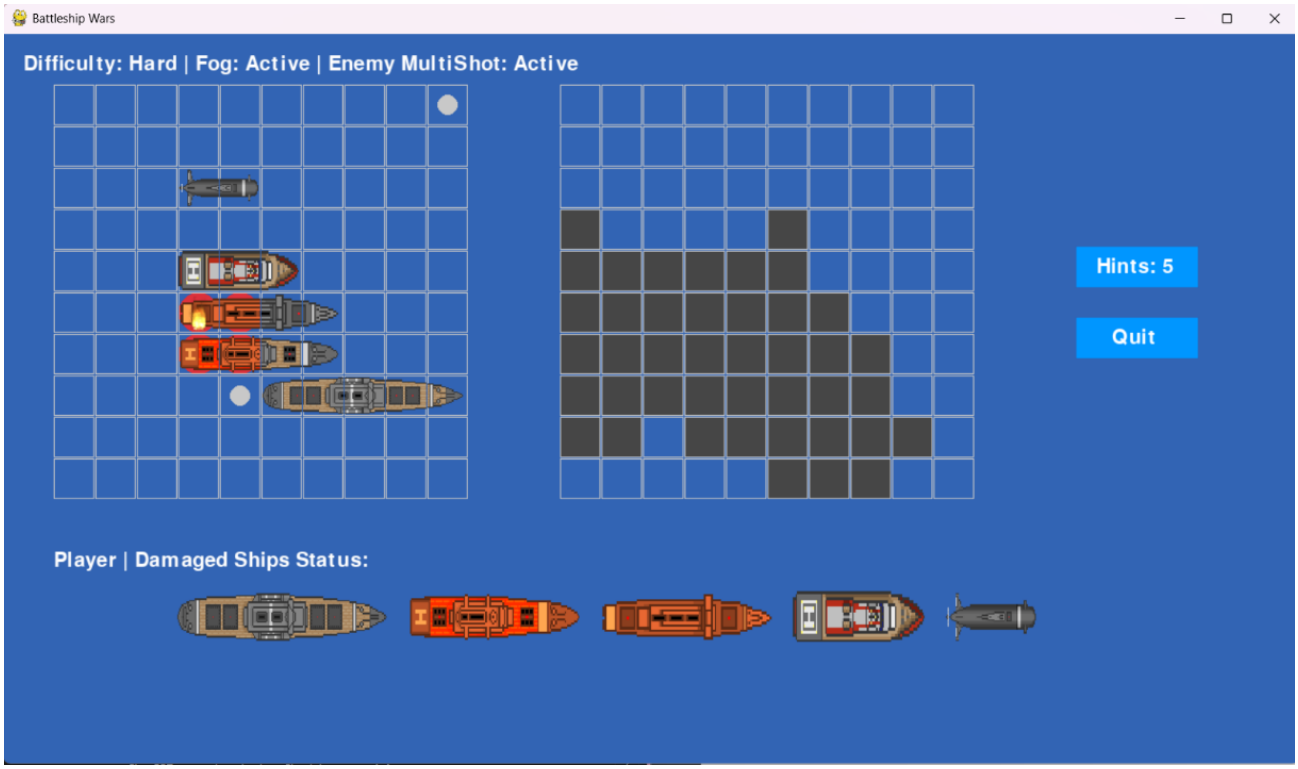
- По успешното поставување на бродовите, **правилото за минимално растојание** осигурува дека бродовите не се поставени премногу блиску еден до друг.
- Компјутерот поставува **празни полиња околу секој брод**, што ја зголемува фер распределбата.
- Ова го спречува компјутерот **да групира бродови заедно**, што би направило потешко напаѓање.

3. Визуелни анимации и функционалности, музика и звуци

3.1. Погодок и промашување на брод

При клик на глушецот на поле на рамката на непријателот (компјутерот), во случајов десната рамка, се прави обид за погодок на непријателскиот брод, што ние досега го нарекувавме „пука“ или „шут“. Ако е погодено поле на кое се наоѓа брод, на играчот му се прикажува црвен круг, анимација на оган (постигната со прикажување на повеќе слики една по друга) и звук на експлозија. Ако пак, шутот е промашување, тогаш се прикажува сиво мало кругче, анимација и звук на удар во вода. Во позадина има две различни песни, едната за менито, другата за играта (сл. 6).

Ако компјутерот погоди брод на играчот, тогаш тој дел од сликата на бродот се бои црвено, и во долната статусна линија (во која се прикажани сите бродови) се бои оној брод кој барем еднаш бил погоден (сл. 6).



Сл. 6. Приказ на анимациите со целосниот изглед на екранот за игра

4. Стратегија на компјутерот за пукање и топлотна мапа на веројатности

Компјутерот наместо случајно погодување на бродовите, користи **таргетирање засновано на веројатност**.

4.1. Генерирање на топлотна мапа на веројатност

На секоја ќелија (поле) на таблата (рамката) ѝ се доделува **вредност на тежина**, базирана на:

- **Погодоци и промашувања од претходните напади**.
  Веројатноста се пресметува на следниот начин:
- **За секоја празна ќелија (x, y):**
  probability_map*[y][x] += *base_value
- **Ако е соседна на познат погодок:**
  probability_map*[y][x] += *adjacency_bonus
- **Ако ќелијата е блиску до центарот на таблата:**
  probability_map*[y][x] += *central_bias (бидејќи играчите често избегнуваат да поставуваат бродови на рабовите).

4.2. Систематско таргетирање и перпендикуларно (ортогонално, под 90 степени) испитување

Кога ќе се регистрира погодок, компјутерот **престанува со случајно пукање и преминува на систематско таргетирање**:

- **Регистрирање погодок:**
  - Компјутерот детектира погодок ако целното поле содржи брод.
  - Погодокот се евидентира во *ai_hits*[y][x] како **2** (означувајќи успешен шут).
- **Продолжување во права линија (систематско таргетирање):**
  - Компјутерот одбира **соседни ќелии** (лево, десно, горе, долу) за следниот напад.
  - Ако се пронајде уште еден погодок, **продолжува во таа насока** додека бродот не биде целосно уништен.
- **Префрлување на ортогонално испитување:**
  - Ако не се пронајдат повеќе погодени полиња во почетната насока, компјутерот **ја менува насоката** (на пример, ако првичната насота е хоризонтална, сега пробува вертикално).
  - Ова осигурува дека компјутерот **не претпоставува дека бродот продолжува во една насока бесконечно**.
- **Ако бродот е потопен (погоден)**, компјутерот се ресетира и продолжува со таргетирање базирано на веројатност.

4.3. Систем на повеќекратни напади според нивото на тежина на играње

- **Лесен режим:** Пука **еден целосно случаен** шут по потег.
- **Среден режим:** Пука **еден шут базиран на веројатност** по потег.
- **Тежок режим:**
  - Пука **до три пати по потег**.
  - Секој шут е избран **од ќелиите со највисока веројатност**.
  - **Ако открие брод, сите шутеви се насочени** за брзо потопување.

5. Магла на војната и како ја менува играта

Оваа верзија на **Battleship воведува механика на магла**, која не постои во дигиталните верзии на оригиналната игра, но концептуално е инспирирана од физичката верзија (каде што играчот мора да заклучи каде погодил без визуелно следење туку само звуци). На почетокот на играта (само во средно или тешко ниво на играње), **кластерите на магла се доделуваат случајно** со пример: ***random.randint* (1, 3) кластери од (15-50) ќелии.**

Секој кластер **се шири во случајни насоки**, обезбедувајќи **неуниформна дистрибуција на маглата**. Маглата **останува активна сè додека играчот не ги открие скриените делови** на бродот со неговите шутеви.

Успешните погодоци и промашувањата остануваат невидливи за играчот, и играшот може да искористи помош (hint).

Маглата спречува играчот да ги следат неговите претходни напади визуелно, принудувајќи **стратешко планирање на нападите**.

Маглата **ја зголемува тежината на играта**, ограничувајќи ја можноста играчот да ги погодува сите полиња по случаен избор. Оваа функционалност, играчот go поттикнува да користи **паметно следење на нападите и стратешко користење на функцијата помош (hints)**.

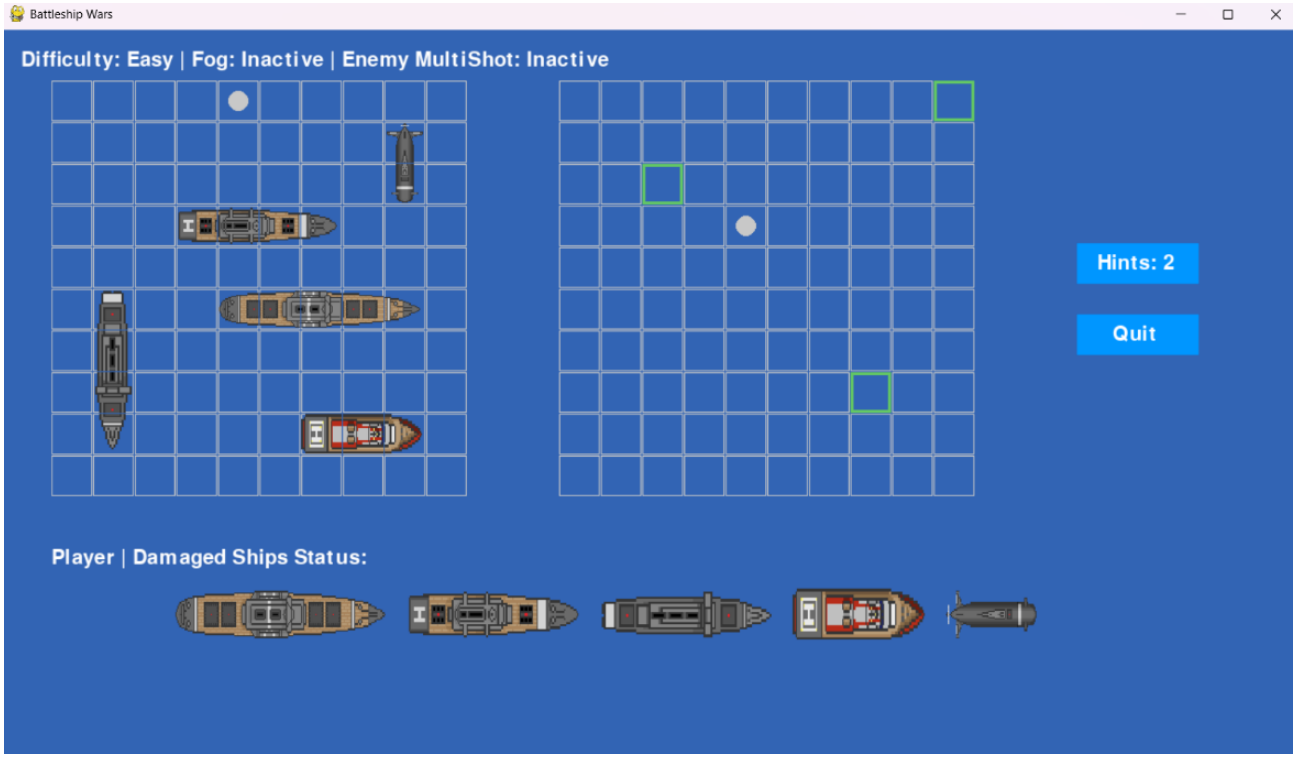
6. Функционалност на помош (hints) и нивното значење во услови со магла

Системот на помош е особено корисен во нивоата на **среден и тежок режим на играње**, каде што маглата го покрива бојното поле.

6.1 Објаснување на функцијата за помош (hints) (Сл.7):

- **Помошта осветлува три случајни полиња на таблата**, од кои **едно содржи дел од брод**.
- Останатите две осветлени полиња се **случајно избрани празни полиња**.
- Играчот мора **да заклучи кое од трите полиња содржи брод**, додавајќи елемент на стратегија наместо директно откривање на непријателскиот брод.

Бидејќи **маглата ги прикрива претходните напади**, функцијата за помош ја обезбедува потребната видливост, намалувајќи го ризикот од непотребни и неефикасни напади.



Сл. 7. Приказ на трите полиња при помош

7. Имплементација на приказ на цел екран (fullscreen)

Функционалноста за прикажување на играта на цел екран е имплементирана со *pygame.FULLSCREEN*. Воведен е и механизам за вклучување/исклучување на оваа функционалност:

1. **Детектирање на команда за приказ на цел екран**:
    - Играта „слуша“ за настанот „притискање на F тастерот“ (*pygame.KEYDOWN* настан) и во *menu.py* и во *game.py*.
2. **Префрлување на играта од и во цел екран**:
    - Ако се притисне тастерот F, **променливата *screen_mode* се менува** помеѓу вредностите *pygame.FULLSCREEN* и *pygame.RESIZABLE*.
    - Прозорецот се ресетира со ***pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT), screen_mode)***.
3. **Одржување на конзистентност низ датотеките**:
    - Променливата *screen_mode* **се пренесува од *menu.py* во *game.py***, осигурувајќи дека изборот за големината екран се задржува при менување на интерфејсот.
4. **Прилагодување на UI елементите**:
    - При преминување од и во цел екран, **позициите на рамките, копчињата и другите UI елементи динамички се прилагодуваат** според новата големина на екранот.
    - Логото и копчињата на менито автоматски **се центрираат со релативни координати** *(SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2).*

8. Заклучок

Оваа верзија на играта битка со бродови, **Battleship,** воведува **механика на магла, адаптивен компјутер противник и стратешки систем на функционалност за помош**, што значително се разликува од традиционалните дигитални варијанти. Маглата ги принудува играчите **ментално да ги следат и памтат нивните напади**, а функционалноста за помош обезбедува привремена видливост, подобрувајќи го стратешкото носење на одлуки. Со интеграцијата на функционалноста за префрлање на играта во и од **цел екран на графичкиот кориснички интерфејс UI**, играта нуди интерактивно и поинакво играње на оваа позната Battleship игра.

9. Референци

https://www.pygame.org/project/4842

https://www.pygame.org/project-Battleship-939-.html

The *Ride of the Valkyries* (German: *Walkürenritt* or *Ritt der Walküren*)