Raport z testowania prostej gry turowej własnego autorstwa (projekt z przedmiotu Wzorce Projektowe)

# Cel: Zapewnienie braku błędów przed prezentacją obecnego stanu projektu i sprawdzenie sprawności użytych metod i klas w celu dalszego rozwoju aplikacji

# Testowane klasy/przypadki z użyciem unittests oraz/lub mock:

1. Character (z użyciem mock do przyspieszenia czasu między klatkami)
   1. Reset do stanu startowego
      1. Nie znaleziono błędów
   2. Zmiana klatek animacji i powrót do stanu „Idle” lub zostanie w stanie „Dead”
      1. Nie znaleziono błędów
2. DamageText stworzone przez Factory
   1. Długość życia obiektu
      1. Obiekt istniał jedną klatkę więcej niż powinien, błąd naprawiono
      2. Bohater może nie mieć przypisanej fabryki i kod tego nie sprawdzał przez co dochodziło do erroru, błąd naprawiono prostym ifem
3. DamageText stworzone „ręcznie”
   1. Długość życia obiektu
      1. Obiekt istniał jedną klatkę więcej niż powinien, błąd naprawiono
      2. Obiekt stworzony z oczekiwanym życiem równym 0, usunie się dopiero po jednej klatce, potencjalny problem przy dalszym rozwoju kodu w nieprzewidzianych przypadkach
4. HealthBar
   1. Automatyczne aktualizowanie klasy HealthBar\_normal
      1. Nie znaleziono błędów
   2. Automatyczne aktualizowanie klasy HealthBar\_action\_panel przy zmianie przypisanego bohatera
      1. Potencjalny problem: klasa HealthBar\_action\_panel nie jest automatycznie aktualizowana tak jak HealthBar\_normal, może to być nieintuicyjne dla przyszłego powrotu do kodu i rozwijania aplikacji

# Testowanie ogólnego działania aplikacji „czarnoskrzynkowo” przez osobę niepiszącą kodu:

Aplikacja pod nadzorem autora i po przekazaniu instrukcji została oddana do użytku osobie nie związanej z projektem. Ten test nie wykazał żadnych błędów w strukturze kodu ani działaniu aplikacji, jednak dostarczył wartościowe informacje na temat podjętych decyzji w procesie tworzenia, intuicyjności interfejsu oraz pomysłów na dalszy rozwój gry.

# Testy manualne wykonane przez autora:

Autor kodu podczas tworzenia i po ukończeniu pracy wykonał szerokie spektrum manualnych testów aplikacji. Wszystkie błędy znalezione podczas pisania kodu zostały natychmiastowo poprawione.

Znalezione nienaprawione błędy:

1. W rzadkich przypadkach, po wybraniu umiejętności wymagającej wybrania celu spośród przeciwników i nakierowaniu na niego kursora, zamiast poprawnego zamienienia standardowego kursora systemowego na ikonę wybranej umiejętności, na czas pojedynczych klatek wyświetlany jest kursor systemowy na wierzchu ikony wybranej umiejętności powodując irytujące migotanie; błąd do dalszego wyszukiwania przypadków wystąpienia

Poniżej zostaną wymienione szczególne przypadki, które zostały sprawdzone i warto je powtarzać przy dalszym rozwoju aplikacji:

* Niezależnie od tego co zrobiliśmy w innych ekranach, po powrocie do głównego menu, przycisk „Run” powinien przenosić nas do ekranu walki z nowymi bohaterami bez śladu wcześniejszej walki
* Dostępność przycisku menu „Return to Main Menu” w dowolnym momencie walki
* Działanie przycisku „Return to Main Menu” oraz resetowanie walki po ponownym powrocie do niej. Możliwe problemy dla użytkownika: brak ostrzeżenia o utraceniu obecnego stanu gry
* Używanie wszystkich umiejętności dostępnych dla wszystkich postaci, każdorazowo aktywowana powinna zostać następna postać niezależnie od użytej umiejętności
* Użycie wszelkich przycisków w dowolnej kolejności; w przyszłości punkt do odrzucenia przez rozmiar projektu, jednak przy obecnym stanie bardzo realna i bezpieczna metoda
* Brak aktywowania postaci (zarówno sojuszniczych jak i przeciwnika) podczas walki
* Poprawne działanie zmian w punktach życia bohaterów, zgodność sprite’a oznaczającego wartość zmiany z faktycznym stanem postaci (heal: tylko do max hp, damage: ogólna ilość zadanych obrażeń)
* W obecnej wersji, informacje w lewym górnym rogu pomagają sprawdzać deweloperowi czy umiejętności przypisane do panelu akcji mają odpowiednio dobraną klasę i nazwy postaci są wyświetlane poprawnie
* W obecnej wersji, informacje w lewym górnym rogu mogą być nieczytelne dla użytkownika, zaleca się rozwinięcie rozwiązania developer\_mode On/Off pozwalające na wspomniane wyżej narzędzie rozwoju oraz przystępność dla użytkownika
* WAŻNE! Zawsze należy upewnić się, że wyświetlane grafiki i teksty są odpowiednich rozmiarów i nie nachodzą na siebie. Grafiki powinny zgadzać się z plikami źródłowymi dla zachowania jakości obrazu (brak rozciągnięć) oraz zakładane jest użycie jednego wzoru czcionki w całej aplikacji
* (Subiektywne): Upewnić się, że odstęp czasowy między akcjami i animacjami nie przytłacza gracza ilością ruchu na ekranie i jest przejrzysty w zrozumieniu tego co się stało w grze
* Zgodność animacji z akcjami podejmowanymi przez gracza i efektami działającymi na postacie