

# Dokument wymagań projektowych

Nazwa projektu: Hack`n`Slash

Autorzy:

- Paweł Stelmach
- Jakub Skwiercz
- Jakub Góralczyk

Data: 15.11.2020r

## 1. Elementy składowe projektu (produkty projektu)

Elementy programistyczne:

- Aplikacja działająca na systemie windows oraz linux.
- Aplikacja napisana w języku C# w silniku unity,

Elementy nie programistyczne:

- specyfikacja funkcjonalna
- raporty z wykonanych testów użyteczności,

## 2. Granice projektu

- Aplikacja będzie działać tylko na 64-bitowy Ubuntu w wersji 12.04 lub nowszej ponieważ unity wspiera tylko takie wersje linuxa
- Aplikacja będzie wspierać windows 10 ponieważ tylko ten system operacyjny posiadamy. Poprzednie wersje systemu operacyjnego windows są nie tylko nie wspierane ale również nie posiadamy do nich dostępu.
- Gra nie będzie umieszczona na platformie steam ponieważ by to zrobić potrzeba mieć zarejestrowaną firmę oraz uiścić opłatę.
- Aplikacja nie będzie dostępna na urządzenia mobilne ponieważ jej mechanika jest zbyt skomplikowana, żeby dało się ją zaimplementować w mobilnym odpowiedniku.

## 3. Lista wymagań funkcjonalnych

Postać:

- Gra powinna pozwalać graczowi na sterowanie postacią po planszy
- Gra powinna pozwalać graczowi na celowanie
- Gra powinna pozwalać graczowi na oddanie strzału
- Postać powinna posiadać ustaloną przez system ilość życia, ataku, obrony oraz prędkość poruszania.
- Gra powinna obejmować ustaloną ilość życia w momencie gdy gracz otrzymuje obrażenia
- Gra powinna zabić postać gdy jej poziom życia wyniesie 0

Menu:

- Gra powinna pozwalać graczowi wybrać opcję wyjścia z gry

- Gra powinna pozwalać graczowi wybrać opcję startu rozgrywki
- Gra powinna pozwalać graczowi na zmianę ustawień sterowania
- Gra powinna pozwalać graczowi na zmianę ustawień dźwięku
- Gra powinna pozwalać graczowi na zmianę ustawień graficznych
- Gra powinna pozwalać graczowi na załadowanie ekranu z listą autorów

#### Pętla gry:

- Gra powinna pozwolić graczowi na pauzowanie rozgrywki
- Gra powinna pozwolić graczowi na wyjście z gry z poziomu menu pauzy
- Gra powinna pozwolić graczowi na powrót do gry z poziomu menu pauzy
- Gra powinna pozwolić graczowi na wyjście do menu głównego z poziomu menu pauzy
- Gra powinna generować w sposób proceduralny poziomy gry
- Gra powinna rozmieszczać przeciwników oraz przeszkody na planszy
- Gra powinna posiadać bossa na końcu każdego poziomu
- Gra powinna generować jednego NPC na mapę
- Gra powinna pozwalać na interakcje z NPC
- Gra powinna pozwalać na wymianę przedmiotów z NPC
- Po pokonaniu bossa gra powinna przechodzić na kolejny poziom

#### Przeciwnicy:

- Przeciwnik powinien posiadać ustaloną przez system ilość życia, ataku, obrony oraz prędkość poruszania.
- Gra powinna obejmować ustaloną ilość życia w momencie gdy przeciwnik otrzymuje obrażenia
- Gra powinna zneutralizować przeciwnika gdy jego poziom życia wyniesie 0
- Przeciwnik powinien posiadać ustalone przez system pole widzenia
- Przeciwnik powinien szukać gracza w obrębie swojego pola widzenia
- Przeciwnik powinien zaatakować gracza po zlokalizowaniu jego pozycji

#### Interfejs użytkownika:

- Gra powinna pozwolić graczowi na podnoszenie przedmiotów
- Gra powinna pozwolić graczowi na używanie umiejętności
- Gra powinna pozwolić na zakładanie przedmiotów w panelu ekwipunku
- Gra powinna pozwalać na zarządzanie ekwipunkiem
- Gra powinna wyświetlać aktualną ilość życia postaci
- Gra powinna wyświetlać aktualną ilość życia przeciwników
- Gra powinna odblokowywać nowe umiejętności po pokonaniu bossa
- Gra powinna pozwalać na wybór umiejętności po rozpoczęciu rozgrywki
- Gra powinna generować losowo skrzynie z przedmiotami
- Gra powinna wyświetlać ilość zadawanych i otrzymywanych obrażeń

#### 4. Lista wymagań нефunkcjonalnych

- Minimalna liczba klatek powinna wynosić: 30
- Gra powinna działać z kartą graficzną NVIDIA GeForce GTX 760 jej odpowiednikiem lub nowszą
- Gra powinna działać z procesorem i3-3220 jej odpowiednikiem lub nowszym
- Gra powinna działać zajmować nie więcej niż 6Gb pamięci na dysku twardym
- Gra powinna działać z 4GB pamięci ram
- Gra powinna działać na systemie Windows 10 oraz 64-bitowy Ubuntu w wersji 12.04 lub nowszej

- Gra powinna wspierać następujące rozdzielczości ekranu:

Długość ekranu w px	Wysokość ekranu w px
1280	720
1366	768
1600	900
1920	1080
2048	1152
2560	1440

#### 5. Mierzalne wskaźniki wdrożeniowe

- Grę będzie można pobrać za darmo z serwisu Itch.io.
- System zostanie poddany testom gwarantującym, poprawne działanie na wskazanych wcześniej konfiguracjach systemowych.
- Na koniec pierwszego semestru system zostanie dostarczony do klienta w wersji testowej alpha (rozumianej zgodnie z metodyką Agile). Na koniec drugiego semestru udostępniony zostanie system w wersji finalnej.

#### 6. Kryteria akceptacji projektu dla I semestru prac

- Wymagane - Dostarczenie prototypu, pozytywnie zakończone testy użyteczności, wdrożenie odpowiednich funkcjonalności potrzebnych do zakończenia prac na tym etapie
- Oczekiwane - Stworzenie Grywalnego prototypu z menu głównym, pętlą gry kilkoma prostymi przeciwnikami oraz podstawowym sterowaniem gracza
- Planowane – Brak zaimplementowania NPC

#### 7. Kryteria akceptacji projektu dla II semestru prac

- Wymagane - Dostarczenie finalnej wersji projektu, pozytywnie zakończone testy użyteczności, wdrożenie odpowiednich funkcjonalności potrzebnych do zakończenia prac na tym etapie
- Oczekiwane - Stworzenie finalnego produktu zawierającego wszystkie funkcjonalności potrzebne do wydania finalnej wersji gry oraz umieszczenie jej na platformie itch.io
- planowane – brak

#### 8. Organizacja pracy zespołu

- Zakres prac poszczególnych członków zespołu:
  - Paweł Stelmach
    - Sterowania postacią
    - Kamera podąża za graczem
    - Przemieszczanie się pomiędzy pokojami
    - Minimapa
    - Sztuczna inteligencja przeciwników
    - Generowanie poziomów
  - Jakub Góralczyk
    - System umiejętności postaci
    - Menu główne
    - Interfejs użytkownika
    - Modele i animacje postaci
    - Efekty cząsteczkowe
  - Jakub Skwiercz
    - Atakowanie przeciwników
    - Używanie przedmiotów
    - Statystyki gracza
    - Zarządzanie ekwipunkiem
- Role projektowe i deweloperskie poszczególnych członków zespołu:
  - Paweł Stelmach - lider zespołu, programista
  - Jakub Góralczyk - scrum master, programista, grafik
  - Jakub Skwiercz - programista
- Jako metodykę pracy zespół przyjął metodyki zwinne Scrum. Metodyki te zostały wybrane jako najlepsze dla procesu tworzenia gier wideo gdyż działamy na przyrostach dzięki czemu w dowolnym momencie wiemy gdzie jesteśmy i mamy działającą wersję produktu oraz wiemy w jakim kierunku rozwija się projekt i jakie funkcje muszą być dodane w następnej iteracji, a jakie funkcje które były nominalnie planowane muszą zostać usunięte. Dzięki temu zespół wie na czym musi skupić uwagę w danej iteracji gry co pozwala na szybszy i bardziej optymalny development gry.
- Jira - Zespół wykorzystuje narzędzie to tworzenia sprintów, przydzielania zadań poszczególnym członkom zespołu, sprawdzania na jakim etapie rozwoju produkt się znajduje oraz zarządzania przebiegiem projektu.
- Unity Collaborate - Wbudowane w ekosystem unity narzędzie pozwalające nam na zarządzanie kodami źródłowymi oraz kolejnymi wersjami projektu
- Cykl życia zadań w projekcie:
  - Lista zadań do wykonania zostaje umieszczona w backlogu
  - Poszczególne zadania trafiają do backlogu sprintu
  - Zadania zostają przydzielone do poszczególnych osób na dany sprint
  - Weekly Scrum - pozwala na sprawdzenie w jakim kierunku rozwija się projekt
  - Sprint Review - po zakończeniu sprintu analiza obecnego wyglądu projektu oraz informacja zwrotna o potrzebnych zmianach

## 9. Ryzyka projektowe:

- Możemy źle oszacować minimalne wymagania systemowe co może spowodować, w tym przypadku należy przetestować finalny projekt i uaktualnić wymagania systemowe.

- Ilość użytkowników na platformie Itch.io jest o wiele mniejsza z konkurencyjną platformą steam co może prowadzić do małego zainteresowania grą. W tym przypadku należy poszukać innych portali typu itch.io oraz tam też zamieścić grę.

### **10. Kamienie milowe**

koniec I semestru - Stworzenie Grywalnego prototypu z menu głównym, pętlą gry kilkoma prostymi przeciwnikami oraz podstawowym sterowaniem gracza

koniec II semestru - Umieszczenie na platformie itch.io gotowej grywalnej gry z 5 losowo generowanymi poziomami na końcu każdego znajduje się boss - przeciwnik z umiejętnościami.