

---

# **Specyfikacja oprogramowania**

**„Life is Hard: Student Edition”**

**Jakub Persjanow  
Politechnika Gdańska**

---

## Spis Treści

Spis treści	2,3
Zmiany specyfikacji oprogramowania	3
Streszczenie	4
1. Wstęp	
1.1 Cel	5
1.2 Zakres	5
1.3 Skróty, Akronimy i Definicje	5
1.4 Referencje	5
1.5 Krótki Przegląd	5
2. Ogólny Opis	
2.1 Walory użytkowe i przydatność projektowanego systemu	6
2.2 Ogólne możliwości projektowanego systemu	6
2.3 Ogólne ograniczenia	6
2.4 Środowisko operacyjne	6
2.5 Charakterystyka użytkowników	6
3. Specyficzne Wymagania	
3.1 Wymagania dotyczące funkcji systemu	6
3.1.1 Interfejs	6
3.1.1.a Menu	6,7
3.1.1.b LifeBoard	8
3.1.1.c Interfejs Rozgrywki	8
3.1.1.d Interfejs Końcowy	8,9
3.1.2 Funkcje	
3.1.2.a Poruszanie się	9
3.1.2.b Interakcja z obiektami, przeciwnikami i NPC	9
3.1.2.c Interakcja z Interfejsem Rozgrywki	9
3.1.2.d Funkcje Rozgrywki	9
3.2 Wymagania dotyczące wydajności systemu	9
3.3 Wymagania dotyczące zewnętrznych interfejsów	9
3.4 Wymagania dotyczące zasobów	9
3.5 Wymagania dotyczące sposobów weryfikacji	9
3.6 Wymagania dotyczące sposobów testowania	9
3.7 Wymagania dotyczące dokumentacji	9
3.8 Wymagania dotyczące ochrony	9

3.9 Wymagania dotyczące przenośności	9
3.10 Wymagania dotyczące jakości	9
3.11 Wymagania dotyczące niezawodności	10
3.12 Wymagania dotyczące pielęgnacyjności	10
3.13 Wymagania dotyczące bezpieczeństwa	10
4. Inne	
4.1 Harmonogram prac nad projektem	10

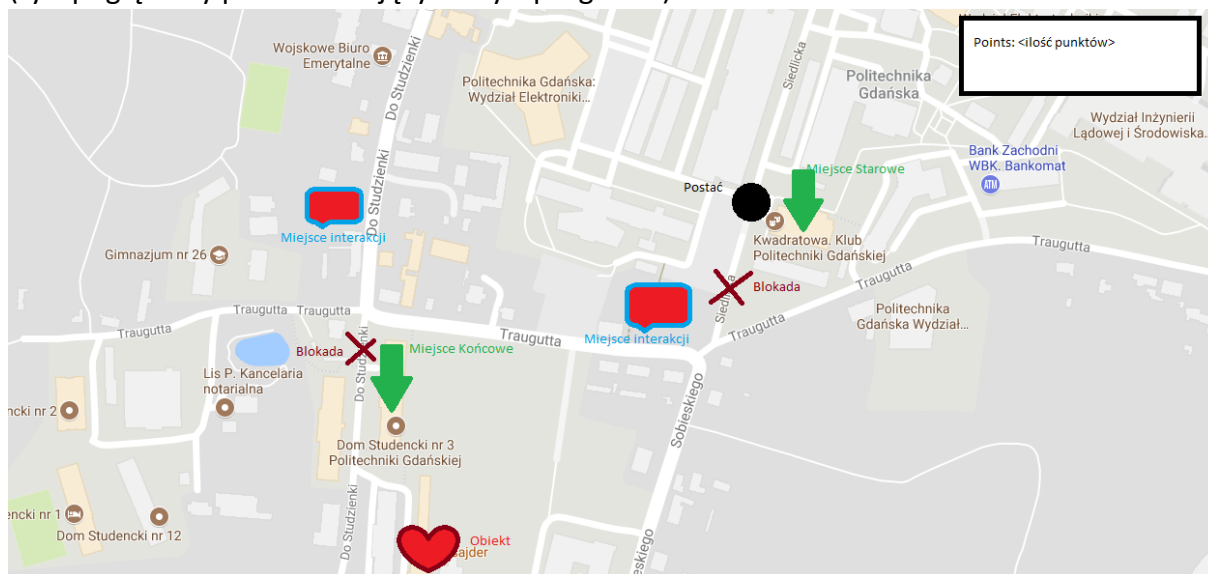
---

Imię i Nazwisko	Data	Zmiany w stosunku do wersji poprzedniej
Jakub Persjanow	17.11.17	n/d

## Streszczenie

„Life is Hard: Student Edition” (zwane dalej grą lub programem) to gra dwuwymiarowa widziana z perspektywy lotu ptaka w której użytkownik (zwany dalej graczem) sterujący postacią, studentem, musi przedostać się z klubu „Square” do domu. Postać porusza się po mapie, na której zaznaczony jest punkt początkowy i końcowy oraz różne pojawiające się obiekty pomocnicze z którymi gracz może wejść w interakcje. W dotarciu do celu będą przeszkadzać graczowi niedogodności w postaci przeciwników, miejsca/obszary zmian położenia postaci (zwane blackout). Aby ukończyć grę postać musi przejść do punktu końcowego, po drodze może (ale nie musi) zbierać dodatkowe punkty poprzez wchodzenie w interakcję z postaciami, pokonywanie przeciwników, zbieranie przedmiotów. Poprzez wykonywanie wyżej wymienionych czynności licznik punktów gracza zwiększa się. Na koniec gry punkty są sumowane i zapisywane na tablicy wyników na której użytkownik może zapisać swoje imię.

(rys. poglądowy przedstawiający zamysł programu)



# 1.Wstęp

## 1.1 Cel

Bazowym celem oprogramowania jest cel rozrywkowy oraz rozwijanie umiejętności sensorycznych, głównie koordynacji ruchowo-wzrokowej użytkownika.

## 1.2 Zakres

## 1.3 Definicje, akronimy i skróty

Wy tłumaczenie zawartych w specyfikacji skrótów, akronimów i definicji:

- Mapa – plansza po której porusza się postać (patrz postać)
- Ulica – miejsce na mapie po których postać ma możliwość poruszania się w kierunku zgodnym z osiami OX i OY
- Postać/Student – obiekt sterowany przez użytkownika
- Blackout – obszary niewidoczne dla użytkownika na mapie (patrz mapa) które zmieniają położenie postaci
- Teleport – zmiana położenia postaci, inicjowane przez blackout
- Friend/NPC – obiekt z którym postać ma możliwość wejścia w interakcję, wyświetla na ekranie interfejs rozmowy, w którym gracz może przeczytać dodatkowe informacje na temat historii gry lub/i uzyskać pomocne informacje, zwiększa sumę punktów
- Bottle – obiekt z którym postać ma możliwość wejścia w interakcję, podwyższa sumę punktów, zwiększa możliwość wystąpienia balackout'u (patrz blackout)
- Sebix/Przeciwnik – obiekt sterowany przez komputer, przeszkadzający postaci w dotarciu do celu
- Street closed – obiekt lub obszar uniemożliwiający postaci przejście po ulicy, niemożliwy do usunięcia
- LifeBoard – tablica wyników na której zapisywane jest imię podane przez użytkownika oraz ilość punktów zebrane podczas rozgrywki
- Menu – Interfejs Podstawowy (patrz punkt 3)
- Changelog – plik tekstowy w którym opisywane są zmiany w kodzie oraz występujące błędy

## 1.4 Referencje

[1] IEEE Std 830-1998 SRS outline

## 1.5 Krótki przegląd

Dokument przedstawia ogólny opis oprogramowania, główne założenia, wymagania oraz dokładny opis funkcji występujących w oprogramowaniu. Dodatkowo przedstawiony został harmonogram pracy nad oprogramowaniem.

## 2. Opis ogólny

### 2.1 Walory użytkowe i przydatność projektowanego systemu

Główną zaletą oprogramowania jest rozwijanie koordynacji wzrokowo-ruchowej u użytkownika poprzez stymulacje bodźcami pojawiającymi się podczas rozgrywki. Dodatkową zaletą jest rozrywka i możliwość poprzez opowiedzianą historię oraz prostotę rozgrywki.

### 2.2 Ogólne możliwości projektowanego systemu

Użytkownik ma możliwość interakcji z oprogramowaniem poprzez interfejs (opisany szczegółowo w punkcie 3). Gracz kontroluje postać klawiaturą, dokładniej strzałkami kierunkowymi, przyciskiem 'e' umożliwiającym interakcje z postaciami oraz przeciwnikami (opisane szczegółowo w punkcie 3) oraz przyciskiem 'esc' umożliwiającym przywołanie menu (szczegółowy opis w punkcie 3).

### 2.3 Ogólne ograniczenia

Ograniczenie do stosowania tylko i wyłącznie obiektów dwuwymiarowych oraz plików audio nie objętych prawami autorskimi bądź opartych na licencji open media.

### 2.4 Środowisko operacyjne

Oprogramowanie zostanie projektowane w języku C wykorzystując środowisko Microsoft Visual Studio 2015 Enterprise oraz bibliotekę Allegro5.

### 2.5 Charakterystyka użytkowników

Oprogramowanie zaprojektowane zostanie z myślą o użytkownikach powyżej 16 roku życia chcących poprawić koordynację ręka-oko oraz chcących korzystać z oprogramowania w celach rozrywkowych. Poziom umiejętności technologicznych ze względu na prostotę rozgrywki i użytkowania jest nieznaczący.

## 3. Specyficzne Wymagania

### 3.1 Wymagania dotyczące funkcji systemu

#### 3.1.1 Interfejs

Interfejs zostanie podzielony pod względem funkcjonalności

##### 3.1.1.a

##### Menu

Podstawowy interfejs zainicjowany od razu po uruchomieniu programu. Tło powinno być czarne, styl całego menu nawiązuje do gier 8-bitowych. Znajdują się w nim:

- Tytuł – wypisany dużą czcionką znajdujący się na środku ekranu zawierający nazwę gry
- New Game – przycisk funkcjonalny znajdujący się w ramce poniżej tytułu, po wyborze przez użytkownika przycisk podświetla się a akceptacja, która następuje po wciśnięciu przycisku 'enter' rozpoczyna nową grę na pierwszej planszy.
- LifeBoard – przycisk funkcjonalny znajdujący się w ramce, poniżej przycisku New Game, po wyborze przez użytkownika podświetla się a akceptacja, która następuje po wciśnięciu przycisku 'enter' przenosi do interfejsu na którym znajdują się zapisane wyniki poprzednich rozgrywek.
- Exit - przycisk funkcjonalny znajdujący się w ramce, poniżej przycisku New Game, po wyborze przez użytkownika podświetla się a akceptacja, która następuje po wciśnięciu przycisku 'enter' zamyka program.

#### 3.1.1.b

##### LifeBoard

Interfejs pokazujący zapisane wyniki (sumy punktów uzyskane przez użytkownika) poprzednich rozgrywek. Tło powinno być czarne a styl samego interfejsu nawiązuje do gier 8-bitowych. Same wyniki przedstawione są w tabelce, uszeregowane od największego do najmniejszego wraz z nazwami użytkowników

#### 3.1.1.c

##### Interfejs Rozgrywki

Główny interfejs programu. Na mapie która reprezentuje ulice widoczny jest Square, Dormitory, Student, Bottle, Sebix, Friend, Street Blocked. Niewidoczne są BlackOut. W prawym górnym rogu widoczny jest licznik punktów zmieniający się podczas rozgrywki. Styl całego interfejsu, przeciwników, postaci oraz samej mapy nawiązuje do gier 8-bitowych.

#### 3.1.1.c'

##### Okienko Rozmowy

Po zainicjowaniu interakcji z Friend zostaje wyświetlone okienko rozmowy w którym pojawia się tekst mówiący o historii gry lub podaje dodatkowe informacje na temat sterowania.

#### 3.1.1.d

##### Interfejs Końcowy

Interfejs wyświetlany po zakończonej rozgrywce (gdy użytkownik dojdzie z Square do Dormitory). Pokazuje sumę punktów zebranych przez gracza oraz opcje zapisania wyniku, możliwości powtórzenia rozgrywki (daje możliwość rozpoczęcia gry od nowa) oraz możliwość wyjścia z gry (zamyka program).

(przykładowy wygląd Menu i LifeBoard)



### 3.1.2 Funkcje

#### 3.1.2.a

Poruszanie się

Użytkownik porusza postać na mapie przyciskami klawiatury.

- Strzałka w górę – odpowiada za przemieszczenia postaci w górę na osi OX
- Strzałka w dół – odpowiada za przemieszczenie postaci w dół na osi OX
- Strzałka w lewo – odpowiada za przemieszczenie postaci na osi OY
- Strzałka w prawo – odpowiada za przemieszczenie postaci na osi OY w stronę przeciwną niż strzałka w lewo

#### 3.1.2.b

Interakcja z obiektami, przeciwnikami i NPC.

- Przycisk 'e' – wciśnięcie podczas ustawienia postaci obok Friend inicjalizuje pokazania okienka rozmowy (opisane w punkcie 3.1.1.c').
- Przycisk 'e' – wciśnięcie podczas ustawienia postaci obok Bottle dodaje dodatkowe punkty do licznika i wyniku końcowego, pokazuje to przez zmianę koloru postaci.
- Przycisk 'e' – wciśnięcie podczas ustawienia postaci obok Sebix niczy przeciwnika dodając dodatkowe punkty do licznika i wyniku końcowego, pokazuje to przez zmianę koloru postaci.

#### 3.1.2.c

Interakcja z Interfejsem rozgrywki

- Przycisk 'esc' – wciśnięcie w dowolnym momencie rozgrywki powoduje wyjście z programu

#### 3.1.2.d

Funkcje rozgrywki

- Możliwość poruszania się (patrz punkt 3.1.2.a)
- Możliwość interakcji (patrz punkty 3.1.2.b i .c)
- Funkcja Blackout  
Obszar niewidoczny na mapie. Po wejściu w niego postacią następuje Teleport, zmiana pozycji gracza na planszy.
- Funkcja Ataku Przeciwników



Przeciwnicy mają możliwość ataku postaci, gdy poruszając się wejdą w obszar postaci a gracz nie zareaguje przyciśnięciem przycisku 'e' ilość punktów na liczniku punktów i w wyniku końcowym zmniejsza się.

### 3.2 Wymagania dotyczące wydajności systemu

Brak szczególnych wymagań

### 3.3 Wymagania dotyczące zewnętrznych interfejsów

Nie dotyczy

### 3.4 Wymagania dotyczące zasobów

#### 3.4.1 Wymagania zalecane

Ze względu na brak testów, wymagania podane zgodnie z maszyną na której tworzone jest oprogramowanie:

Intel® Core™ i5-3210M CPU @ 2,50 GHZ 2,50 GHZ (lub odpowiadające)

RAM: 8 GB

Karta Graficzna: Nvidia GTX 660 M (lub odpowiadająca)

### 3.5 Wymagania dotyczące sposobów weryfikacji

Weryfikacja przeprowadzona zostanie w systemie cotygodniowych raportów o zmianach w oprogramowaniu

### 3.6 Wymagania dotyczące sposobów testowania

Oprogramowanie zostanie przetestowane na trzech jednostkach w celach określenia optymalizacji. W fazie końcowej zostaną przeprowadzone testy przez osoby w różnym wieku i z różnymi umiejętnościami technicznymi w celu sprawdzenia łatwości obsługi oraz poziomu trudności rozgrywki.

### 3.7 Wymagania dotyczące dokumentacji

Zmiany w kodzie oprogramowania zostaną zapisane w changelogu łącznie z datami zmian i opisem zmian.

### 3.8 Wymagania dotyczące ochrony

Nie dotyczy

### 3.9 Wymagania dotyczące przenośności

Oprogramowanie zostanie opracowane w taki sposób aby samo przenoszenie było łatwe celem testów na innych jednostkach.

### 3.10 Wymagania dotyczące jakości

Oprogramowanie powinno być zoptymalizowane.

### 3.11 Wymagania dotyczące niezawodności

Oprogramowanie powinno być stabilne.

### 3.12 Wymagania dotyczące pielęgnacyjności

Oprogramowanie powinno być sprawdzane oraz testowane podczas każdej zmiany kodu. Zachowania niezgodne z założeniami lub niespodziewane podczas działania programu powinny być eliminowane oraz zapisywane w changelogu.

### 3.13 Wymagania dotyczące bezpieczeństwa

Nie dotyczy

---

## 4. Dodatki

### 4.1 Harmonogram prac nad projektem

Data	Cel
24.11.17	Stworzenie postaci, zainicjowanie kontroli nad postacią
01.12.17	Stworzenie obiektów będących przeciwnikami
08.12.17	Stworzenie planszy, stworzenie menu
15.12.17	Stworzenie obiektów będącymi NPC oraz obszarów oddziałujących na gracza
22.12.17	Zaprojektowanie modeli oraz wyglądu obiektów, przeciwników, NPC oraz postaci
05.01.18	Kontynuacja projektowania, ulepszanie funkcji rozgrywki
12.01.18	Okres Testowy, poprawa błędów
19.01.18	Usprawnienie oprogramowania, testy końcowe, finalna poprawa błędów