Tytuł: 2D Game - Nidhogg

Autorzy: Róża Dwornik (RD), Przemysław Kurzak (PK)

Ostatnia modyfikacja:

Spis treści

[1. Repozytorium git projektu 1](#__RefHeading___Toc2562_3733460636)

[2. Wstęp 2](#__RefHeading___Toc375_832888478)

[3. Specyfikacja 2](#__RefHeading___Toc377_832888478)

[3.1. Opis ogólny algorytmu 2](#__RefHeading___Toc379_832888478)

[3.2. Tabela zdarzeń 2](#__RefHeading___Toc381_832888478)

[4. Architektura 2](#__RefHeading___Toc383_832888478)

[4.1. Moduł: top 2](#__RefHeading___Toc385_832888478)

[4.1.1. Schemat modułu głównego z IP integratora Vivado. 2](#__RefHeading___Toc387_832888478)

[5. Implementacja. Zaawansowanie na 14.03.2018 – 0% 2](#__RefHeading___Toc996_1436539759)

[6. Film. Zaawansowanie na 14.03.2018 – 0% 2](#__RefHeading___Toc1249_1436539759)

**Zanim zaczniesz projekt**

Przed rozpoczęciem projektu należy się przygotować:

* przejść tutorial firmy Digilent „Getting Started with the Vivado IP Integrator” ze strony <https://reference.digilentinc.com/vivado/getting-started-with-ipi/start>
* zapoznać się z protokołem AXI Light, źródło dowolne, np.:  
  <https://www.youtube.com/watch?v=cDc9B2zAPz4>
* zapoznać się z budową klientów AXI (AXI light slave), np.:  
  <https://www.youtube.com/watch?v=Vs0h0kue7p4>
* przejrzeć tutorial „AXI Interface Debug Using Vivado IP Integrator” ze strony  
  <https://www.xilinx.com/video/hardware/axi-interface-debug-using-vivado-ip-integrator.html>
* opcjonalnie: opracuj w oparciu o procesor microblaze projekt, który będzie zawierał klienta AXI odczytującego dane z klawiatury / myszy i w zależności od nich sterował diodami świecącymi
* opcjonalnie: opracuj w oparciu o procesor microblaze projekt, który będzie wyświetlał cokolwiek na ekranie VGA, w zależności od akcji użytkownika (klawiatura / mysz),
* zwróć uwagę na istnienie dokumentu „Integrated Logic Analyzer (ILA)”:   
  <https://www.xilinx.com/products/intellectual-property/ila.html#documentation>

# Repozytorium git projektu

Adres repozytorium GITa (jeżeli używane):

<https://github.com/NidhoggProject/UEC2_Nidhogg>

# Wstęp

[*https://github.com/NidhoggProject/UEC2\_Nidhogg/blob/master/GrupaH\_Nidhogg\_Dwornik\_Kurzak\_OpisProjektu.pdf*](https://github.com/NidhoggProject/UEC2_Nidhogg/blob/master/GrupaH_Nidhogg_Dwornik_Kurzak_OpisProjektu.pdf)

Pomysł został zainspirowany oryginalną grą Nidhogg ([*http://nidhogggame.com/*](http://nidhogggame.com/)). Naszym założeniem jest stworzenie gry, która będzie posiadać większość podstawowych funkcji oryginalnej gry (na potrzeby projektu niektóre funkcjonalności będą pominięte, bądź rozwiązane na podstawie przyjętych przez nas założeń).

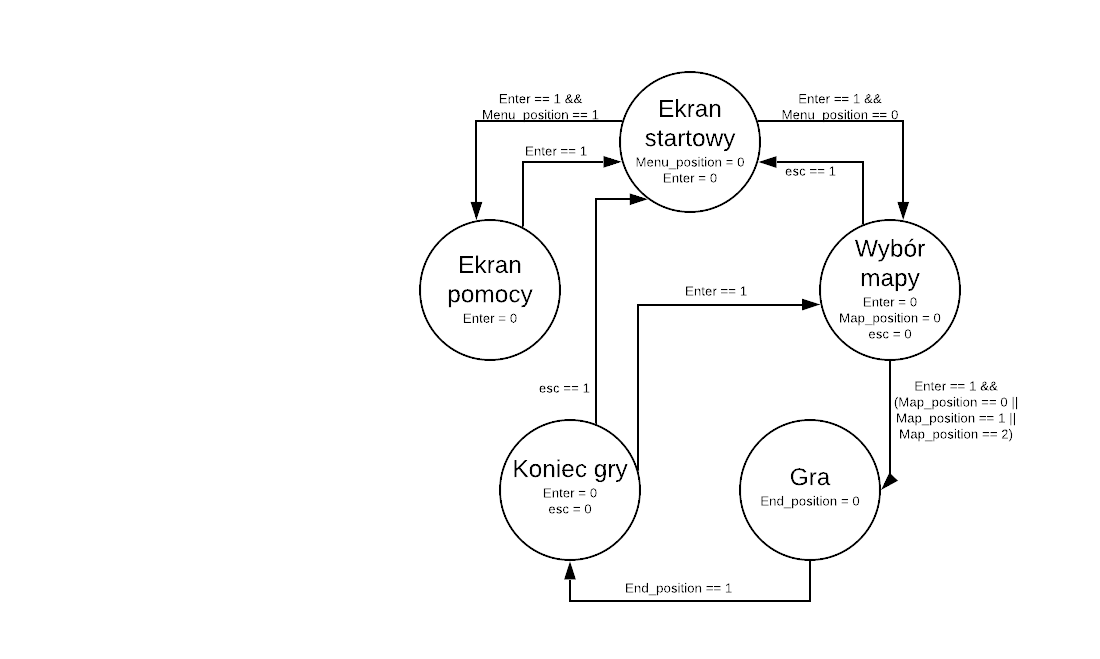
# Specyfikacja

## Opis ogólny algorytmu

*Po implementacji programu użytkownik widzi ekran startowy z trzema opcjami:*

*- Play - uruchamia wybór planszy, a następnie grę dla dwóch graczy,*

*- How to play - pokazuje ekran z zasadami gry i opisem sterowania.*

**

## Tabela zdarzeń

*Opis zdarzeń/akcji występujących podczas działania programu/urządzenia, zarówno zewnętrznych (interakcje z użytkownikiem), jak i wewnętrznych (specyficzne stany w algorytmie). Zdarzenia podzielone są na kategorie dotyczący różnych stanów działania programu. Kategorie powinny odpowiadać stanom ze schematu z pkt. 2.1.*

|  |  |  |
| --- | --- | --- |
| **Zdarzenie** | **Kategoria** | **Reakcja systemu** |
| Naciśnięcie strzałki w górę lub w dół | Ekran startowy | Ruch między dostępnymi opcjami wyboru w menu |
| Naciśnięcie klawisza Enter | Ekran startowy | Wybór opcji z menu |
| Wybór „Play” w menu | Ekran startowy | Przeniesienie użytkowników do pola wyboru mapy |
| Naciśnięcie esc w polu wyboru mapy | Wybór mapy | Powrót do ekranu głównego |
| Naciśnięcie strzałki (→, ←) | Wybór mapy | Ruch między obrazkami symbolizującymi dostępne plansze |
| Wciśnięcie klawisza Enter | Wybór mapy | Przeniesienie do głównego okna gry |
| Naciśnięcie strzałki ← | Gra | Przesunięcie PP w lewo |
| Naciśnięcie strzałki → | Gra | Przesunięcie PP w prawo |
| Naciśnięcie klawisza Enter | Gra | Skok PP |
| Naciśnięcie prawego Shifta | Gra | Rzut lub podniesienie miecza przez PP |
| Naciśnięcie strzałki w górę | Gra | Zmiana poziomu miecza (podniesienie w górę) przez PP |
| Naciśnięcie strzałki w dół | Gra | Obniżenie poziomu miecza przez PP |
| Naciśnięcie klawisza a | Gra | Przesunięcie LP w lewo |
| Naciśnięcie klawisza d | Gra | Przesunięcie LP w prawo |
| Naciśnięcie spacji | Gra | Skok LP |
| Naciśnięcie lewego Shifta | Gra | Rzut lub podniesienie miecza przez LP |
| Naciśnięcie klawisza w | Gra | Zmiana poziomu miecza (podniesienie w górę) przez LP |
| Naciśnięcie klawisza s | Gra | Obniżenie poziomu miecza przez LP |
| Naciśnięcie klawisza Enter | Koniec gry | Powrót do wyboru planszy |
| Naciśnięcie klawisza esc | Koniec gry | Przeniesienie do ekranu startowego |
| Wybór „How to play” w menu | Ekran startowy | Przeniesienie do ekranu pomocy |
| Naciśnięcie klawisza Enter | Ekran pomocy | Powrót do ekranu startowego |

# Architektura

Uwaga: dobrze zrobiony projekt zawiera tylko moduły strukturalne (zbudowane z innych modułów) i funkcjonalne (zawierające bloki proceduralne always @). Staramy się nie generować bloków mieszających te dwa typy, o ile to możliwe.

## Moduł: top

Osoba odpowiedzialna: JK

### Schemat modułu głównego z IP Integratora Vivado.

< tu umieść schemat >

# Film.

Link do ściągnięcia filmu:

https://