



**WYŻSZA SZKOŁA BANKOWA**  
**w Poznaniu**

WYDZIAŁ FINANSÓW I BANKOWOŚCI

# **Implementacja gry platformowej**

PROJEKT DYPLOMOWY

Poznań 2020



**WYŻSZA SZKOŁA BANKOWA**  
**w Poznaniu**



## **DANE PARTNERÓW**

### **A1. Dane Opiekuna**

Imię i nazwisko	<b>Grzegorz Nowak</b>
Stopień / Tytuł naukowy	Doktor
Data i podpis	

### **A2. Dane członków Zespołu projektu**

Imię i nazwisko	<b>Karol Krasuski</b>
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarny
Data i podpis	

Imię i nazwisko	<b>Bartosz Kwiatkowski</b>
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarny
Podpis	

Imię i nazwisko	<b>Jakub Łodaj</b>
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarny
Data i podpis	

Imię i nazwisko	<b>Jakub Łysiak</b>
Kierunek studiów	Informatyka
Tryb studiów	Niestacjonarny
Data i podpis	



## **Spis treści**

<b>ZAŁOŻENIA PROJEKTU .....</b>	<b>5</b>
B1. Opis projektu .....	5
B2. Cele projektu.....	6
<b>REALIZACJA.....</b>	<b>6</b>
C1. Zadania w projekcie.....	6
C2. Użyteczność projektu .....	10
C3. Wykorzystane materiały związane z realizacją projektu.....	11
<b>ZAŁĄCZNIKI.....</b>	<b>13</b>
1.1. Problematyka projektu dyplomowego - Trudności w tworzeniu gier komputerowych.....	13
2.1. Opis narzędzia GDevelop i jego funkcjonalność .....	15
3.1. Wniosek z rozpoznania .....	22
3.2. Opis fabuły .....	24
4.1. Baza wiedzy GDevelop oraz witryny tematyczne - przekrój znalezionych informacji .....	27
5.1. Argumentacja i przedstawienie podstawowej szaty graficznej gry .....	34
6.1. Opis scen zawartych w grze .....	36
6.2. Przedstawienie zdarzeń zewnętrznych zawartych w grze.....	44
7.1. Proces testowania produktu projektowego.....	48
7.2. Raport błędów i opis sposobu ich rozwiązania .....	49
7.3. Wdrożenie poprawek, raport z pomyślnie przeprowadzonych testów.....	50
8.1. Opis procesu eksportu na platformy użytkowe .....	53
<b>SPIS RYSUNKÓW .....</b>	<b>55</b>
<b>BIBLIOGRAFIA.....</b>	<b>57</b>

## **ZAŁOŻENIA PROJEKTU**

### **B1. Opis projektu**

#### **1. Problem badawczy**

Problemem w projekcie jest stworzenie grywalnej wersji gry platformowej opartej na silniku wysokopoziomowego narzędzia programistycznego GDevelop. Wymagane jest stworzenie mechaniki samej rozgrywki, animowanej postaci bohatera głównego oraz postaci pobocznych, a także tła będącego jednocześnie grywalnym elementem gry. Dodatkowo problemem jest odpowiedni zarys fabularny przebiegu gry wpleciony w całość rozgrywki.

#### **2. Uzasadnienie wyboru tematu**

Gra jako produkt końcowy przede wszystkim ma służyć celom rozrywkowym grających, jednak w przypadku stosowania narzędzia GDevelop sam proces tworzenia gry może stanowić swoistą rozrywkę dla samych twórców. Wybierając to narzędzie optymalizujemy wykorzystanie czasu skupiając się przede wszystkim na grywalności i głównym wątku gry jednocześnie poświęcając mniej czasu na kwestie techniczne. Duża część technicznych aspektów jest już zaimplementowana, zawarta w silniku gry i gotowa do użycia w dostarczonym przez producenta narzędzia GDevelop. Oprócz tego niewątpliwie pożądaną cechą w dzisiejszym świecie jest wieloplatformowość aplikacji.

#### **3. Zakres podmiotowy, przedmiotowy, czasowy i przestrzenny**

- przedmiot: narzędzie programistyczne GDevelop,
- zakres badawczy: implementacja gry platformowej,
- zakres czasowy: projekt realizowany od października 2019 do stycznia 2020.

#### **4. Metodyka pracy (metody i techniki badawcze)**

- język programowania: C++ (natywny dla narzędzia GDevelop),
- środowisko programistyczne: GDevelop,
- darmowe oprogramowanie do edycji grafiki wektorowej: GIMP,
- platforma do przechowywania i wersjonowania poszczególnych etapów, projektu: GitHub,



- narzędzie do kontroli cyklu życia projektu: Trello,
- narzędzie do komunikacji grupowej: Facebook Messenger.

## **B2. Cele projektu**

Celem projektu dyplomowego jest zaprojektowanie grywalnej wersji gry platformowej przy użyciu programistycznego narzędzia wysokopoziomowego GDevelop. Aby osiągnąć tak postawiony cel projektu, określono następujące zadania i cele szczegółowe:

- zaprojektowanie oprawy graficznej i otoczenia dla poszczególnych etapów gry,
- opracowanie fabuły i wiodącego wątku tematycznego w grze,
- zaprojektowanie, animacja głównej i pobocznych postaci występujących w grze,
- implementacja zachowań (ang. events) wszystkich aktywnych elementów gry,
- eksport projektu na użytkowe platformy (Web, Mobile, Desktop).

# **REALIZACJA**

## **C1. Zadania w projekcie**

### **Zadanie 1**

Nazwa zadania	<b>Przedstawienie problematyki projektu dyplomowego</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Bartosz Kwiatkowski</b>
Krótki opis zadania	Zadanie polegało na opisie problematyki projektu dyplomowego. Wskazano problemy, ich możliwe przyczyny i sposoby rozwiązania.
Efekty realizacji zadania	<b>Załącznik 1.1. Problematyka projektu dyplomowego - Trudności w tworzeniu gier komputerowych</b>
Termin rozpoczęcia realizacji zadania	Grudzień 2019
Termin zakończenia realizacji zadania	Styczeń 2020



### **Zadanie 2**

Nazwa zadania	<b>Opis obszaru badawczego/podmiotu projektu dyplomowego</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Jakub Łodaj</b>
Krótki opis zadania	Opis narzędzia GDevelop. - Przekrój funkcjonalności
Efekty realizacji zadania	<b>Załącznik 2.1. Opis narzędzia GDevelop i jego funkcjonalność</b>
Termin rozpoczęcia realizacji zadania	Grudzień 2019
Termin zakończenia realizacji zadania	Styczeń 2020

### **Zadanie 3**

Nazwa zadania	<b>Rozpoznanie rynku gier platformowych oraz proponowanie zarysu fabularnego gry.</b>
Podmioty zaangażowane w realizację zadania	<b>Jakub Łysiak</b>
Krótki opis zadania	Badanie zapotrzebowania na rodzaj i charakter gry platformowej. Zredagowanie głównego wątku w grze
Efekty realizacji zadania	<b>Załącznik 3.1. Wniosek z rozpoznania</b> <b>Załącznik 3.2. Opis fabuły</b>
Termin rozpoczęcia realizacji zadania	Październik 2019
Termin zakończenia realizacji zadania	Listopad 2019

### **Zadanie 4**

Nazwa zadania	<b>Analiza literatury w celu znalezienia sposobów/metod/narzędzi do przygotowania rozwiązania projektu dyplomowego</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Bartosz Kwiatkowski, Jakub Łodaj, Jakub Łysiak</b>

Krótki opis zadania	Zadanie polegało na odnalezieniu sposobów rozwiązania problemów oraz metod programistycznych w celu stworzenia aplikacji. Wykorzystano publikacje dostępne w literaturze oraz Internecie.
Efekty realizacji zadania	<b>Załącznik 4.1. Baza wiedzy GDevelop oraz witryny tematyczne - przekrój znalezionych informacji</b>
Termin rozpoczęcia realizacji zadania	Październik 2019
Termin zakończenia realizacji zadania	Grudzień 2019

#### **Zadanie 5**

Nazwa zadania	<b>Zaprojektowanie szaty graficznej dla ostatecznej wersji gry</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Bartosz Kwiatkowski, Jakub Łodaj, Jakub Łysiak</b>
Krótki opis zadania	Zadanie zawiera utworzenie projektu ostatecznej wersji oprawy graficznej gry (w tym bohaterów i ich animacje)
Efekty realizacji zadania	<b>Załącznik 5.1. Argumentacja i przedstawienie podstawowej szaty graficznej gry</b>
Termin rozpoczęcia realizacji zadania	Październik 2019
Termin zakończenia realizacji zadania	Grudzień 2019

#### **Zadanie 6**

Nazwa zadania	<b>Dostrojenie silnika gry oraz implementacja pojedynczych zdarzeń pod potrzeby projektu</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Bartosz Kwiatkowski, Jakub Łodaj, Jakub Łysiak</b>
Krótki opis zadania	Zadanie polega na edycji silnika gry w taki sposób, aby odpowiednio reagował na konkretne zdarzenia podczas rozgrywki
Efekty realizacji zadania	<b>Załącznik 6.1. Opis scen zawartych w grze</b> <b>Załącznik 6.2. Przedstawienie zdarzeń zewnętrznych zawartych w grze</b>





Termin rozpoczęcia realizacji zadania	Październik 2019
Termin zakończenia realizacji zadania	Styczeń 2020

#### **Zadanie 7**

Nazwa zadania	<b>Testy grywalnej części gry, korekcja błędów</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Bartosz Kwiatkowski, Jakub Łodaj, Jakub Łysiak</b>
Krótki opis zadania	Testowanie gry pod kątem potencjalnych błędów, raportowanie znalezionych błędów, zaproponowanie i podjęcie próby naprawy napotkanych błędów. Ponowne testy w oparciu o zastosowane rozwiązania
Efekty realizacji zadania	<b>Załącznik 7.1. Proces testowania produktu projektowego</b> <b>Załącznik 7.2. Raport błędów i opis sposobu ich rozwiązania</b> <b>Załącznik 7.3. Wdrożenie poprawek, raport z pomyślnie przeprowadzonych testów</b>
Termin rozpoczęcia realizacji zadania	Październik 2019
Termin zakończenia realizacji zadania	Styczeń 2020

#### **Zadanie 8**

Nazwa zadania	<b>Eksport ostatecznego projektu na konkretne platformy użytkowe</b>
Podmioty zaangażowane w realizację zadania	<b>Bartosz Kwiatkowski, Jakub Łodaj</b>
Krótki opis zadania	Wyeksportowanie pliku projektowego na platformy Web, Desktop Mobile
Efekty realizacji zadania	<b>Załącznik 8.1. Opis procesu eksportu na platformy użytkowe</b> <b>Gotowy produkt projektowy w postaci wykonywalnych aplikacji lub aplikacji webowej</b>
Termin rozpoczęcia realizacji zadania	Styczeń 2020
Termin zakończenia realizacji zadania	Styczeń 2020

## Zadanie 9

Nazwa zadania	<b>Przygotowanie dokumentacji końcowej i prezentacji multimedialnej projektu dyplomowego</b>
Podmioty zaangażowane w realizację zadania	<b>Karol Krasuski, Bartosz Kwiatkowski, Jakub Łodaj, Jakub Łysiak</b>
Krótki opis zadania	Zadanie polegało na przedstawieniu grywalnej części gry, przykładowym jej działaniu oraz omówieniu rozgrywki.
Efekty realizacji zadania	<b>Prezentacja multimedialna i dokumentacja końcowa projektu dyplomowego</b>
Termin rozpoczęcia realizacji zadania	Styczeń 2020
Termin zakończenia realizacji zadania	Luty 2020

## C2. Użyteczność projektu

Użyteczność projektu przygotowanego przez grupę polega na przedstawieniu możliwości narzędzia programistycznego GDevelop na konkretnym przykładzie tzn.: funkcjonalnej i przystępnej użytkownikowi gry platformowej. Projekt wykazuje również, że bez poświęcania stosunkowo dużej ilości czasu można stworzyć mechanikę gry podobną do projektów, które były tworzone od początku przy użyciu niskopoziomowych narzędzi. Tym samym projekt udowadnia, że skupiając się na cechach kluczowych gry tj. fabule i grywalności, możemy uzyskać znacznie lepsze rezultaty końcowe.

Kwintesencją projektu dyplomowego jest wypuszczenie gotowego produktu tj. gry komputerowej dostępnej na najpopularniejsze platformy. Głównym aspektem użyteczności jest walor rozrywkowy, który jest niewątpliwie jednym z ważniejszych celów tworzenia gier komputerowych. Oprócz tego kod źródłowy projektu może służyć również celom edukacyjnym jako wzorzec projektowy.

Gotowy produkt będzie gotowy do publikacji na największych platformach growych np. (Steam, Google Play). Natomiast kod źródłowy został opublikowany na platformie GitHub jako projekt wzorcowy dla przyszłych produkcji i będzie dostępny dla wszystkich zainteresowanych pod adresem [https://github.com/kwiat323/Praca\\_Inzynierska](https://github.com/kwiat323/Praca_Inzynierska).



### C3. Wykorzystane materiały związane z realizacją projektu

- 1) *Player, arrow* - <https://www.gamedevmarket.net/asset/fantasy-heroes-character-sprite-sheet-10156/> [dostęp 14.12.2019].
- 2) *Slime* - <https://bevouliin.com/free-2d-game-asset-pink-monster-game-character/> [dostęp 17.01.2020].
- 3) *Spider* - <https://bevouliin.com/furry-monster-free-2d-game-asset/> [dostęp 16.01.2020].
- 4) *Slime2* - <https://opengameart.org/content/bevouliin-free-orange-bubble-land-monster-sprite-sheets> [dostęp 16.01.2020].
- 5) *Fly, hearth, coin* - <https://kenney.nl/assets/platformer-pack-redux> [dostęp 20.12.2019].
- 6) *AddHearth, AddCoin* - <https://craftpix.net/freebies/free-game-coins-sprite-sheets/> [dostęp 18.12.2019].
- 7) *Shuriken* - <https://opengameart.org/content/shuriken-pixel-art> [dostęp 17.01.2020].
- 8) *Boss1, Boss2, Boss3* - <https://craftpix.net/freebies/free-fallen-angel-chibi-2d-game-sprites/> [dostęp 24.12.2019].
- 9) *Spike* - <https://gamedeveloperstudio.itch.io/floor-spikes> [dostęp 19.12.2019]
- 10) *Ledge* - <https://www.gamedeveloperstudio.com/graphics/viewgraphic.php?item=1e4j3506912n753r9ket> [dostęp 20.12.2019].
- 11) *Tło 1 i 2* - <https://craftpix.net/product/cartoon-forest-2d-backgrounds/> [17.12.2019].
- 12) *Tło 3* - <https://craftpix.net/freebies/free-horizontal-2d-game-backgrounds/> [dostęp 19.01.2020].
- 13) *Tło Menu, Story, MenuInGame, GameOver* - <https://free-game-assets.itch.io/free-jungle-cartoon-gui> [dostęp 18.12.2019].
- 14) *Otoczenie 1 i 2* - <https://www.gameart2d.com/free-platformer-game-tileset.html> [dostęp 18.12.2019].
- 15) *Otoczenie 3* - <https://www.kenney.nl/assets/platformer-art-extended-tileset> [dostęp 19.12.2019].
- 16) *ArrowShoot* - <https://freesound.org/people/brendan89/sounds/321553/> [dostęp 19.12.2019].



- 17) *BackgroundMusicMenu* - <https://freesound.org/people/Greek555/sounds/497865/> [dostęp 19.12.2019].
- 18) *Confirm* - <https://freesound.org/people/JustInvoke/sounds/446114/> [dostęp 20.01.2020].
- 19) *Death* - <https://freesound.org/people/AlineAudio/sounds/416839/> [dostęp 21.12.2019].
- 20) *Enemy\_hit* - <https://freesound.org/people/D001447733/sounds/464623/> [dostęp 09.01.2020].
- 21) *Jump* - <https://freesound.org/people/BehanSean/sounds/422426/> [dostęp 19.12.2019].
- 22) *Jungle* - <https://freesound.org/people/aurelien.leveque/sounds/417635/> [dostęp 19.12.2019].
- 23) *Jungle2* - [https://freesound.org/people/jaime\\_enrique/sounds/424763/](https://freesound.org/people/jaime_enrique/sounds/424763/) [dostęp 20.01.2020].
- 24) *Menu\_click* - <https://freesound.org/people/NenadSimic/sounds/171697/> [dostęp 21.11.2019].
- 25) *Memorphile – Spooky Dungeon* - <https://freesound.org/people/LittleRobotSoundFactory/sounds/270402/> [dostęp 21.01.2020].
- 26) *PickCoin* - <https://freesound.org/people/ProjectsU012/sounds/341695/> [dostęp 19.12.2019].
- 27) *PickHearth* - [https://freesound.org/people/cabled\\_mess/sounds/350876/](https://freesound.org/people/cabled_mess/sounds/350876/) [dostęp 19.12.2019].
- 28) *Run* - <https://freesound.org/people/bevangoldswain/sounds/54778/> [dostęp 19.12.2019].
- 29) *ShurikenThrow* - <https://freesound.org/people/marchon11/sounds/493224/> [dostęp 20.01.2020].
- 30) *Spike* - <https://freesound.org/people/Deathscyp/sounds/443806/> [dostęp 19.12.2019].
- 31) *Youlose* - <https://freesound.org/people/mouse85224/sounds/371205/> [dostęp 21.01.2020].
- 32) *Youwin* - <https://freesound.org/people/LittleRobotSoundFactory/sounds/270402/> [dostęp 21.01.2020].

## ZAŁĄCZNIKI

### 1.1. Problematyka projektu dyplomowego - Trudności w tworzeniu gier komputerowych

Problemem badawczym w projekcie jest zgłębienie procesu implementacji gry platformowej. Zadanie to składa się z kilku etapów opisanych poniżej.

Pierwsza, a zarazem początkowa faza projektu powinna rozpocząć się od analizy dostępnych na rynku gier i potrzeb użytkowników. Na tym etapie należy również określić zakres i cel projektu, kierując się przede wszystkim użytecznością i zapotrzebowaniem tego typu produktów na rynku.

Następnym krokiem jest pochylenie się nad zagadnieniami teoretycznymi tj.: zdefiniowanie parametrów takich jak:

- zakres projektu,
- typ i kategoria gry,
- czas i miejsce gry,
- zaproponowanie ogólnego zarysu fabularnego.

Kolejnymi, a zarazem najtrudniejszym w realizacji elementami projektu jest faza projektowania i faza implementacji.

Pierwsza z nich: faza projektowania — polega w dużej mierze na przeszukaniu obecnego rynku developerskiego, dobraniu i przygotowaniu przydatnych w projekcie narzędzi programistycznych. Na tym etapie należałoby dopasować i wybrać najlepsze możliwe pomoce programistyczne a przede wszystkim:

- środowisko deweloperskie,
- narzędzie kontrolujące fazy życia projektu,
- platformę do przechowywania i wersjonowania poszczególnych etapów projektu,
- narzędzia do formalnej formy komunikacji (w przypadku projektów grupowych lub sponsorowanych).

Po skompletowaniu poszczególnych narzędzi projektu należy przystąpić do właściwego procesu tworzenia projektu gry.

Problemy, które można napotkać w trakcie tworzenia to w dużej mierze implementacja procesu związana z mechaniką gry na przykład:

- tworzenie i konfiguracja parametrów zdarzeń (ang. events) odpowiadających za czynności postaci w zależności od zaistniałej sytuacji,
- opracowanie postaci, jej animacji, a także zachowań w otoczeniu poszczególnych elementów gry.

Oprócz tego niezwykle ważnym dla efektu końcowego aspektem jest zaprojektowanie odpowiedniej szaty graficznej, wliczając w to wszystkie dostępne dla graczy wizualne elementy gry.

Dodatkowo w momencie samego procesu tworzenia, a także po jego ukończeniu należy przeprowadzić odpowiednie dla każdego etapu testy produkcyjne. Testy te powinny zakończyć się odpowiednim raportem opisującym ich wynik, a także w razie jakichkolwiek błędów możliwe propozycje ich rozwiązania.

Końcowym etapami są:

- Eksport pliku projektowego na konkretne platformy użytkowe,
- Przeprowadzenie po raz kolejny testów wykonanych już na docelowych platformach oraz ewentualna naprawa wykrytych błędów,
- Opracowanie dokumentacji projektowej.

Opracowanie dokumentacji projektowej powinno odbyć w taki sposób, aby osoby, nie związane z projektem w przyszłości miały możliwość rozbudowania i kontynuacji projektu.

## 2.1. Opis narzędzia GDevelop i jego funkcjonalność

### 1. Wstęp

GDevelop to oprogramowanie pozwalające na tworzenie wszelkiego rodzaju gier 2D bez użycia języka programowania. Program ten szczególnie zasługuje na uwagę, ponieważ jest przystępny dla początkujących deweloperów gier. Projektowanie gier odbywa się za sprawą intuicyjnych i modułowych elementów mechaniki gry tj.: warunków i akcji w postaci bloków nazywanych zdarzeniami. Program jest prawie całkowicie darmowy, a gotowe gry, które eksportujemy są w pełni kompatybilne z systemami: Windows, Linux, Android, iOS oraz platformy Internetowe (na silniku HTML5). Gotowe produkty które stworzyliśmy, w świetle prawa należą całkowicie do użytkownika, który to ma jednocześnie prawo do komercyjnego korzystania z niego bez opłat licencyjnych. Jedynym znaczącym ograniczeniem w darmowej wersji programu jest limitowana ilość kompilacji projektu które można wykonać na daną platformę w ciągu dnia. Silnik i kod IDE, które składają się na środowisko GDevelop, są wolnym oprogramowaniem na licencji typu Open Source, każdy, kto chce współtworzyć środowisko GDevelop, może to zrobić w dowolnym momencie. Kod źródłowy narzędzia jest dostępny w oficjalnym repozytorium na platformie GitHub pod adresem: <https://github.com/4ian/GDevelop>.

## 2. Opis interfejsu okna głównego

**Rysunek 1 Główne okno programu**



Źródło: Opracowanie własne

### 1) Pasek nawigacyjny składający się z pięciu rozwijanych list:

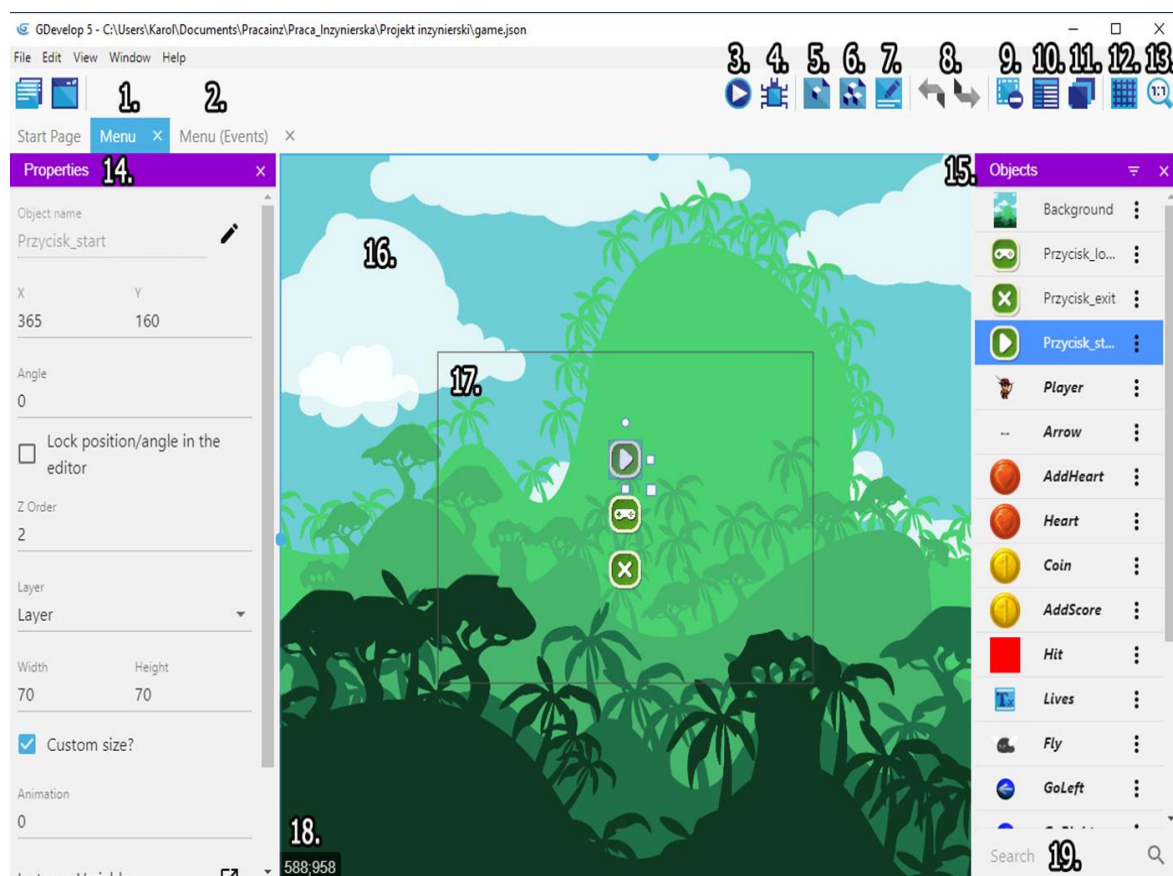
- File - w jego zakładkach znajdują się najważniejsze opcje programu np. takie jak tworzenie nowego projektu, otwarcie istniejącego lub export projektu na konkretne platformy,
- Edit - narzędzia pozwalające się cofnąć i przywrócić dokonane wcześniej zmiany oraz wklejać lub wycinać elementy projektu,
- View - zakładka z widokami uruchamiające poszczególne elementy interfejsu tj. menadżera projektu, menadżera scen, itp.,
- Window - pozwala zminimalizować okno programu,
- Help - zakładka zawierająca odnośniki do dokumentacji oraz witryn internetowych poświęconych środowisku Ikony skrótu.



- 2) Ikony skrótu - odnośniki do kluczowych elementów programu tj.: menadżera projektu, kompilatora i strony głównej programu.
- 3) Odnośniki do poradników i forum - odnośniki do witryn internetowych poświęconych narzędziu GDevelop jego dokumentacji (GDevelop Wiki) i forum dyskusyjnym poruszającym kwestie techniczne projektów.
- 4) Menadżer projektu - przejście do widoku szkieletowego projektu z podziałem na poszczególne sceny.
- 5) Odnośniki do mediów społecznościowych / Zmiany języka - łączy do mediów społecznościowych, gdzie znajdują się profile tematyczne poświęcone programowi GDevelop, oraz aktualnościami z nim związanymi. Dodatkowo obok znajdują się możliwość wyboru języka, chociaż zalecaną praktyką jest korzystanie z natywnej, anglojęzycznej wersji, gdyż cała dokumentacja jest właśnie w tym języku i nie wszystkie zwroty zostały przetłumaczone na inne języki.

### 3. Opis widoku sceny

**Rysunek 2 Widok sceny**



Źródło: Opracowanie własne

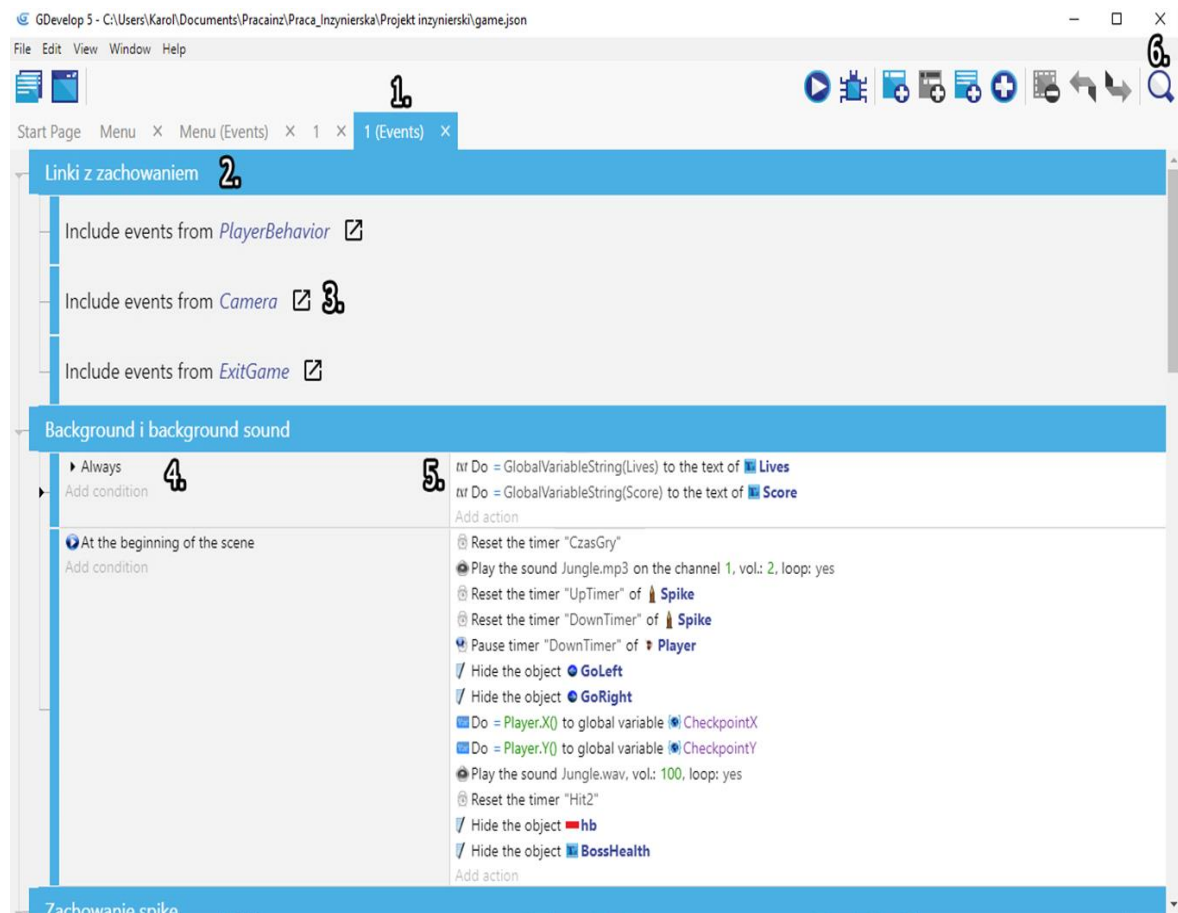


- 1) Okno główne sceny - przejście do widoku, który umożliwia edycje wizualną sceny, dodawanie obiektu itp. (Rysunek 2).
- 2) Okno zdarzeń sceny - przejście do widoku, który umożliwia zarządzanie zdarzeniami w ramach danej sceny (Rysunek 3).
- 3) Symulacja sceny - narzędzie umożliwiający skompilowanie bieżącej sceny w celu zasymulowania widoku sceny i mechaniki gry.
- 4) Debugger - narzędzie do dynamicznej analizy procedury zdarzeń i błędów w kodzie.
- 5) Edytor obiektów - skrót do narzędzia pozwalającego zarządzać obiektami globalnymi oraz tymi przypisanymi do sceny (15).
- 6) Edytor grup obiektów - skrót do menadżera grup obiektów, umożliwiającego łączenie obiektów w grupy.
- 7) Edytor właściwości projektu - skrót do edytora właściwości obiektu.
- 8) Cofnij/powtór - pozwala na cofnięcie zmian dokonanych w ramach danej sceny (o jeden krok w przód lub w tył).
- 9) Usuń instancje obiektu ze sceny (skrót klawiszowy: delete) - usuwa instancję obiektu po ówczesnym zaznaczeniu.
- 10) Lista instancji - wyświetla edytor instancji obiektów z listą obiektów w bieżącej scenie.
- 11) Edytor warstw - wyświetla edytor warstw pozwalający na dodanie, usunięcie i manipulację warstwami.
- 12) Edycja widoku / widok siatki - pozwala ukryć siatkę sceny oraz ramkę maski sceny.
- 13) Powiększenie - pozwala manipulować powiększeniem podglądu bieżącej sceny.
- 14) Zakładka właściwości - pozwala edytować parametry takie jak: wielkość położenie, warstwa czy zmienne instancji, istniejących już instancji obiektów.
- 15) Zakładka obiekty - pozwala dodawać i edytować obiekty, czyli główne elementy projektu.
- 16) Podgląd sceny - okno ilustrujące aktualny wygląd scenę i rozłożenie obiektów względem współrzędnych.
- 17) Podgląd maski - wykadrowana scena, która będzie widoczna w oknie gry.
- 18) Współrzędne kursora - Pozwalają zidentyfikować konkretny punkt na siatce sceny.

- 19) Wyszukiwarka obiektów - narzędzie przeszukujące istniejące w projekcie obiekty po nazwie (Szczególnie przydatne przy większej ilości obiektów).

#### 4. Opis widoku zdarzeń

**Rysunek 3 Okno widoku zdarzeń**



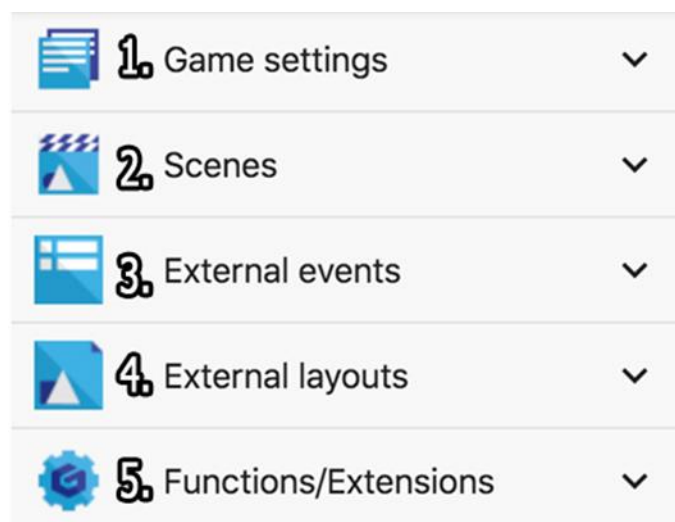
Źródło: Opracowanie własne

- 1) Okno zdarzeń sceny - przejście do widoku, który umożliwia zarządzanie zdarzeniami w ramach danej sceny (Rysunek 3).
- 2) Grupy zdarzeń - element hierarchii porządkujący zdarzenia w scenie. Może zawierać zdarzenia, które również posiadają już jakąś hierarchię. Można go przywoływać również w innych scenach korzystając np. odnośników (3).
- 3) Odnośniki do zdarzeń - rodzaj zdarzenia który może dziedziczyć warunki i akcje już z istniejących grup zdarzeń lub całych scen.
- 4) Warunki - elementarna jednostka organizacyjna w sekcji zdarzenia, gdy warunek jej porządkowy zostanie spełniony wykonuje zadaną w prawej sekcji akcję (5).

- 5) Akcje - konkretne polecenia wykonane po spełnieniu jednego lub więcej warunków. Mogą wpływać na wszystkie elementy w grze.
- 6) Wyszukiwarka sekcji zdarzeń - narzędzie wyszukujące po nazwę w ramach sekcji zdarzenia (szczególnie przydatne przy skomplikowanej hierarchii i większej ilości zdarzeń).

## 5. Menadżer projektu

### Rysunek 4 Menadżer projektu GDevelop



Źródło: Opracowanie własne

- 1) Ustawienia gry - zakładka, w której można dostosować globalne ustawienia gry. Na przykład: domyślny rozmiar okna gry lub jej nazwa. Dodatkowo w tym miejscu można ustawić ikony dla swojej gry. Wszystkie te ustawienia możesz zobaczyć w zakładce „Właściwości”.
- 2) Sceny - to różne widoki i poziomy w grze. W tym miejscu można wybrać bieżącą scenę, zarządzać istniejącymi lub tworzyć nowe sceny.
- 3) Wydarzenia zewnętrzne – w celu ułatwienia i kontrolowania rozwoju gry, można stworzyć wydarzenia w oddzielnych arkuszach wydarzeń, które nie są powiązane z określoną sceną. Nazywa się je zdarzeniami zewnętrznymi. Stworzone wydarzenia możemy połączyć z obiektami zdarzeń, kiedy ich potrzebujemy wykorzystując odnośniki zdarzeń (Rysunek 3). Jeśli musimy często korzystać z kolekcji wydarzeń w różnych miejscach, możemy ustawić je jako zewnętrzne i wywoływać je ze scen zamiast wielokrotnie dodawać te same zdarzenia pozbywając się



w ten sposób dwóch z tzw. *Code Smells*<sup>1</sup> (*Duplicated code*<sup>2</sup> oraz *Shotgun surgery*<sup>3</sup>).

- 4) Układy zewnętrzne - podobnie jak zdarzenia zewnętrzne, możesz tworzyć generyczne poziomy i ponownie wykorzystywać część poziomów w układach wewnętrznych, a następnie ładować je do swoich scen ponownie unikając Code Smells.
- 5) Funkcje / rozszerzenia - pozwalają tworzyć własne wydarzenia i łączyć je w funkcje dla jeszcze lepszej przejrzystości kodu czyniąc wydarzenia bardziej modułowymi.

---

<sup>1</sup> Zapach kodu (ang. Code Smells) – nawyki występujące przy wytwarzaniu oprogramowania uznawane za nieprawidłowe. Element badawczy w procesie tworzenia oprogramowania szczególnie na etapie code review – czyli analizy poprawności kodu. Bardzo często wykrywa i naprawie je się za pomocą automatycznych skanerów kodów czyniąc kod bardziej przejrzystym i zrozumiałym.

<sup>2</sup> Powielony kod (ang. Duplicated Code) – jeden z zapachów kodu. Występuje, gdy identyczny lub bardzo podobny kod istnieje w więcej niż jednej lokalizacji.

<sup>3</sup> Shotgun surgery - jeden z zapachów kodu. Występuje, gdy jedna zmiana musi być zastosowana do wielu klas projektu jednocześnie. Inaczej wywoła nieprawidłowości w działaniu kodu.

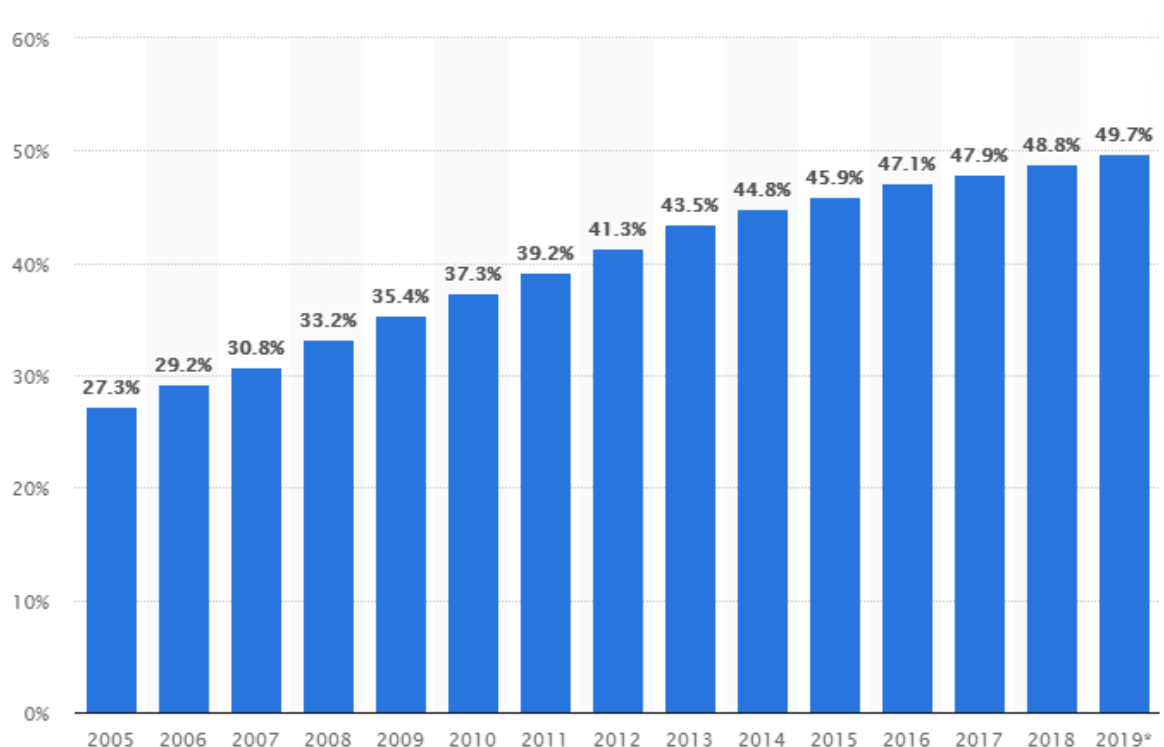
### 3.1. Wniosek z rozpoznania

#### 1. Analiza docelowej platformy dla gry

W 2018 r. prawie połowa prywatnych gospodarstw domowych na całym świecie miała komputer w domu (Rysunek 5). W krajach dopiero rozwijających się wskaźnik posiadania komputera jest niższy, a około jedna trzecia gospodarstw domowych ma komputer. Natomiast udział gospodarstw domowych z komputerem osobistym w krajach rozwiniętych przekracza 80 procent. Ogólnie udział gospodarstw domowych z komputerem stale rośnie na całym świecie, ponieważ korzystanie z komputera i dostęp do Internetu stają się coraz powszechniejsze na całym świecie. W 2019 roku z szacunków wynika, że już około 49,7% gospodarstw domowych posiadało komputery osobiste.

Zdecydowano, że tworzona gra zostanie dostosowana do komputerów osobistych oraz wyeksportowana jako plik wykonywalny do systemu Windows.

**Rysunek 5 Ilość gospodarstw domowych posiadających komputer osobisty w domu na całym świecie w latach 2005-2019**



Źródło: <https://www.statista.com/>



## 2. Analiza popularnych gier platformowych

Rayman Legends (Rysunek 6) jest jedną z najlepszych gier tego gatunku, rozgrywka toczy się w świecie fantasy. Głównym bohaterem jest tytułowy Rayman który wraz z przyjaciółmi budzi się z wieloletniego snu, aby ocalić krainę przed złymi stworami.

Głównym celem gry jest zbieranie Lumów, sennych, złotych kul, pojawiających się wszędzie wokół, a także uwalnianie Małaków. Rozgrywka polega na przechodzeniu kolejnych poziomów i zabijaniu różnych stworów. Przez znaczną część rozgrywki poruszamy się głównie w prawo.

### Rysunek 6 Zrzut ekranu z gry Rayman Legends



Źródło: <https://www.ubisoft.com/en-us/game/rayman-legends/>

Trine 4: The Nightmare Prince (Rysunek 7) kolejna odsłona gry, która w 2019 roku nadal przyciąga graczy. Akcja toczy się w świecie fantasy, obraca się wokół młodego księcia Seliusa którego nocami nękają koszmary, jednak dzięki jego mocy potwory pojawiające się w jego snach stają się prawdziwe i sieją spustoszenie. Niestety księżę przepadł bez wieści a za jego odnalezienie odpowiedzialni są doskonale znani z poprzednich części mag Amadeus, złodziejka Zoya oraz rycerz Pontius.

W grze przemierzamy zamkowe ruiny, nawiedzone grobowce, gęste lasy oraz inne baśniowe lokacje, które obfitują w liczne pułapki i przeszkody terenowe. Poza tym od czasu do czasu napotykamy przeciwników do wyeliminowania w tym przerośniętych bossów, a także rozwiązujemy zagadki środowiskowe. Mamy tu doskonale rozwinięty

system walki bohater Pontius świetnie radzi sobie w walce w zwarcu, Zoya atakuje przeciwników z łuku oraz korzysta z liny z hakiem, dzięki której może dostać się w trudno dostępne miejsca, z kolei Amadeus potrafi materializować i przenosić wybrane obiekty przy pomocy telekinezy.

### **Rysunek 7 Zrzut ekranu z gry Trine 4: The Nightmare Prince**



Źródło: <https://www.trine4.com/>

W tworzonej grze dostosowano się do panujących trendów gier platformowych. Gra podzielona zostanie na kilka etapów podczas której bohater będzie przemierzał krainę napotykając przeszkody terenowe oraz pułapki, zbierał punkty oraz toczył walki z przeciwnikami i większymi potężniejszymi bossami.

### **3.2. Opis fabuły**

Fabułę zbudowano na podstawie słów kluczowych (Tabela 1). Tworzenie fabuły, wokół której będzie toczyć się rozgrywka rozpoczęto od zbadania jakimi kluczowymi słowami są zainteresowani użytkownicy. Ciemnym kolorem niebieskim zaznaczono słowa kluczowe, które były brane za najważniejsze w pierwszej kolejności, jasnym kolorem niebieskim natomiast słowa kluczowe które były brane pod uwagę w drugiej kolejności jako mniej ważne przy tworzeniu fabuły. Słowa kluczowe na których bazowano w większości mają dużą liczbę wyświetleń w ciągu miesiąca w rejonie Polski. Zbudowanie fabuły wokół najbardziej pożądaných słów kluczowych pozwoli na pozyskanie większej potencjalnej grupy użytkowników zainteresowanych grą.



**Tabela 1 Słowa kluczowe**

Słowa kluczowe	Średnia miesięczna liczba wyszukiwań
archer	1 tys. – 10 tys.
arrow	10 tys. – 100 tys.
castle	1 tys. – 10 tys.
dragon	10 tys. – 100 tys.
fantasy	1 tys. – 10 tys.
forest	10 tys. – 100 tys.
gold	1 tys. – 10 tys.
hero	10 tys. – 100 tys.
island	1 tys. – 10 tys.
jungle	1 tys. – 10 tys.
king	10 tys. – 100 tys.
kingdom	1 tys. – 10 tys.
knight	1 tys. – 10 tys.
legend	1 tys. – 10 tys.
love	10 tys. – 100 tys.
magic	10 tys. – 100 tys.
monster	10 tys. – 100 tys.

Zródło: <https://ads.google.com/>

Po analizie słów kluczowych fabuła tworzonej gry będzie rozgrywać się w świecie fantasy, gdyż dla tego gatunku gry pasuje najwięcej słów kluczowych przez co jest większe prawdopodobieństwo zainteresowania taką rozgrywką. Poniżej przedstawiono zarys fabuły, według której będzie toczyć się cała rozgrywka:

*W Królestwie Caseden jednym z najbardziej malowniczych i pięknych krain na ziemi. Ludzie, zwierzęta różnorakie stwory żyją tu ze sobą w idealnej harmonii. Żył tam również jeden z najznamienitszych łuczników a nosił imię Armin. Wraz ze swoją rodziną żył w zdrowiu, spokoju i szczęściu. Żył tam również Gotan jeden z nadwornych kowali króla wraz ze swoją żoną Mariem. Pewnego razu król Caseden wyprawił wielkie przyjęcie na*

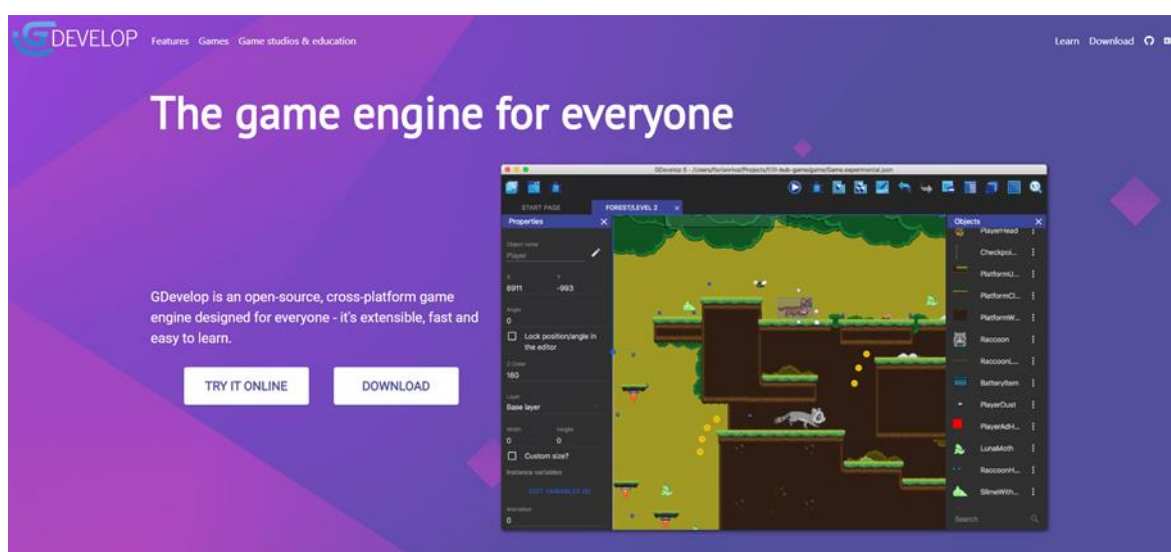
*część nowo narodzonego syna. Przybyli wszyscy w tym również Gotan i jego żona. Podczas wieczerzy doszło jednak do tragedii żona Gotana nagle źle się poczuła a po kilku dniach zmarła z niewiadomych przyczyn. Gotan pogrążony w gniewie i żalu obrzucił winą króla za śmierć ukochanej, próbując go zabić wpadł w ręce straży i został wygnany z królestwa. Jednak zapewnił, że pewnego razu powróci i zemści się za śmierć ukochanej. Mijały lata a w Caseden nic się nie zmieniło. Jednak pewnego poranka przybył Gotan wraz ze swoją zabójczą armią, którą gromadził przez lata, spustoszył całą krainę, wziął w niewolę ludność, niektórym jednak udało wyrwać i uniknąć niewoli, lecz musieli uciec z dala od armii Gotana aby mieć szansę na przeżycie, wśród nich był Armin. Nie mógł się pogodzić z tym, że zostawił rodzinę i swój dom. Gdy nabrał sił nie patrząc na innych postanowił wybrać się w samotną wędrówkę pełną niebezpieczeństw, aby powrócić i wyzwolić swoich bliskich oraz całe królestwo.*

Gracz wciela się w postać Armina, który przemierza królestwo Caseden w celu wyzwolenia swojego królestwa i rodziny z rąk groźnego Gotana. W związku z taką fabułą grze nadano tytuł *Armin Legends*.

#### 4.1. Baza wiedzy GDevelop oraz witryny tematyczne - przekrój znalezionych informacji

1. Oficjalna strona projektu w języku angielskim znajduje się pod adresem <https://gdevelop-app.com/>. Istnieje jednak kilka wersji językowych strony, ale również polska wersja serwisu dostępna pod adresem <https://gdevelop-app.com/pl/>. Wszystkie te witryny mają głównie funkcję prezentacyjną i znajdują się tutaj podstawowe informacje dotyczące gry a także odnośniki do kolejnych witryn poświęconych projektowi.

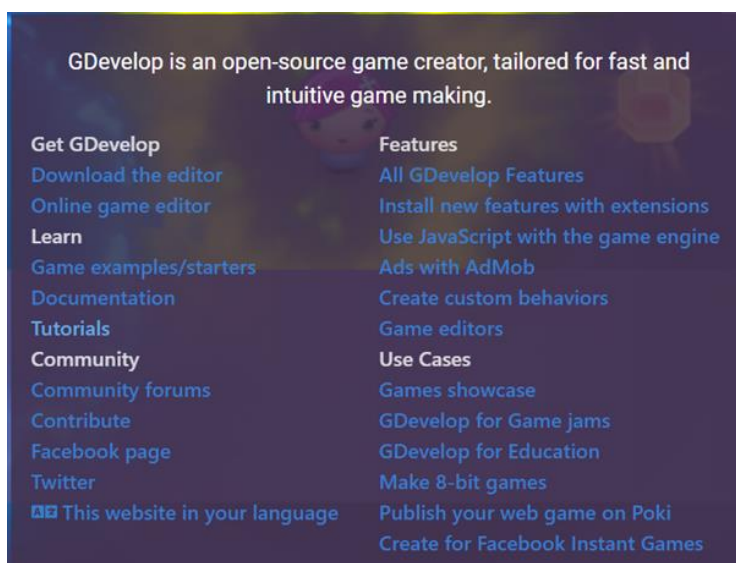
##### Rysunek 8 Oficjalna strona GDevelop



Źródło: Opracowanie własne

Sama strona zawiera przekrój funkcjonalności programu podany w przystępnej formie. Natomiast na końcu strony znajdują się spis najpotrzebniejszych odnośników dotyczących projektu GDevelop.

**Rysunek 9** Lista odnośników do witryn tematycznych znajdująca się na stronie głównej programu



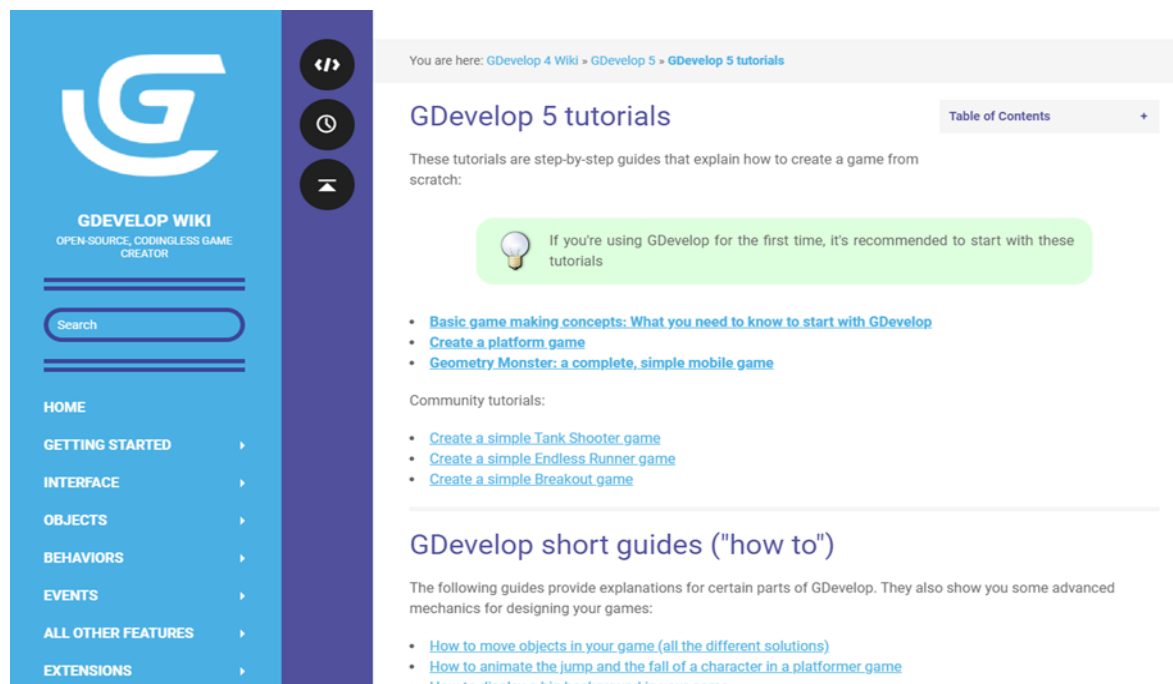
Źródło: Opracowanie własne

2. GDevelop Wiki - obok oficjalnej strony producenci platformy GDevelop udostępniają serwis informacyjny typu Wiki, gdzie możemy sami dodawać lub znaleźć szczegółowe informacje na temat samego projektu, funkcjonalności oraz sposobu ich obsługi w przystępny sposób w postaci np. poradników. Serwis Wiki poświęcony GDevelop istnieje w dwóch odsłonach:

- Najnowsze kompendium wiedzy znajduje się pod adresem <http://wiki.compilgames.net/doku.php/gdevelop5/start> - jest to zaktualizowana wersja serwisu dostępnego w starszej wersji pod adresem,
- <http://wiki.compilgames.net/doku.php/gdevelop/documentation> - tutaj znajdziemy najważniejsze informacje, jeżeli korzystamy ze starszej wersji programu (GDevelop 4).

Nowa, odświeżona wersja strony posiada menu nawigacyjne z lewej strony i jest podzielona na kilka sekcji tematycznych w których są odnośniki do opisów poszczególnych zagadnień związanych z programem.

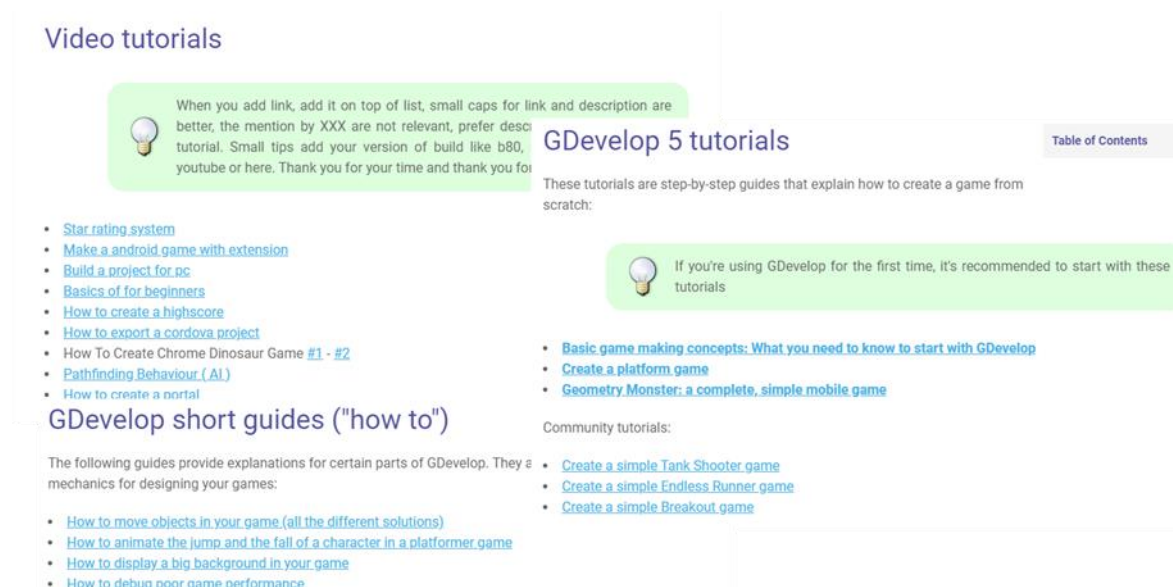
## Rysunek 10 GDevelop Wiki



Źródło: Opracowanie własne

Na szczególną uwagę zasługuje sekcja poradników która to w swojej treści zawiera gotowe przepisy na podstawowe typy gry (Tank shooter, Endless runner itp.) a oprócz tego porusza konkretne problemy mechaniki i w przystępny sposób wyjaśnia zaimplementowane rozwiązanie w platformie GDevelop. Poza tym na stronie znajduje się również sekcja z poradnikami w formie wideo, gdzie społeczność dzieli się swoją wiedzą i udostępnia nagrania instruktażowe w serwisie YouTube.

## Rysunek 11 Przykłady poradników w serwisie GDevelop Wiki

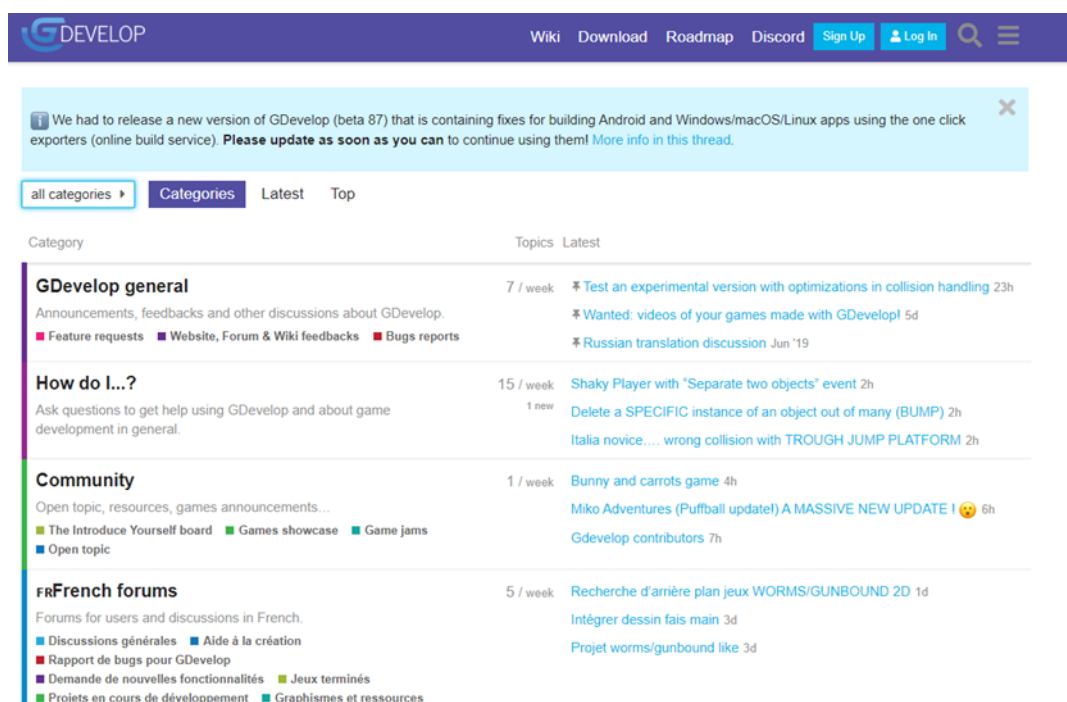


Źródło: Opracowanie własne

3. Forum zrzeszające społeczność użytkowników i twórców GDevelop - oficjalne forum znajdujące się pod adresem <https://forum.gdevelop-app.com/> zrzesza społeczność użytkowników GDevelop. Forum daje możliwość tworzenia i prowadzenia wątków dotyczących samego narzędzia a także sposobu jego wykorzystania w trakcie konkretnych problemów implementacyjnych w poszczególnych grach. Samo forum składa się z kilku modułów dzielących je na segmenty takie jak:

- sekcję ogólnotematyczną,
- sekcję z poradnikami,
- osobne działy dla poszczególnych języków,
- sekcję poświęconą społeczności, gdzie użytkownicy dzielą się swoimi zasobami, projektami czy publikują gotowe gry.

## Rysunek 12 Forum GDevelop

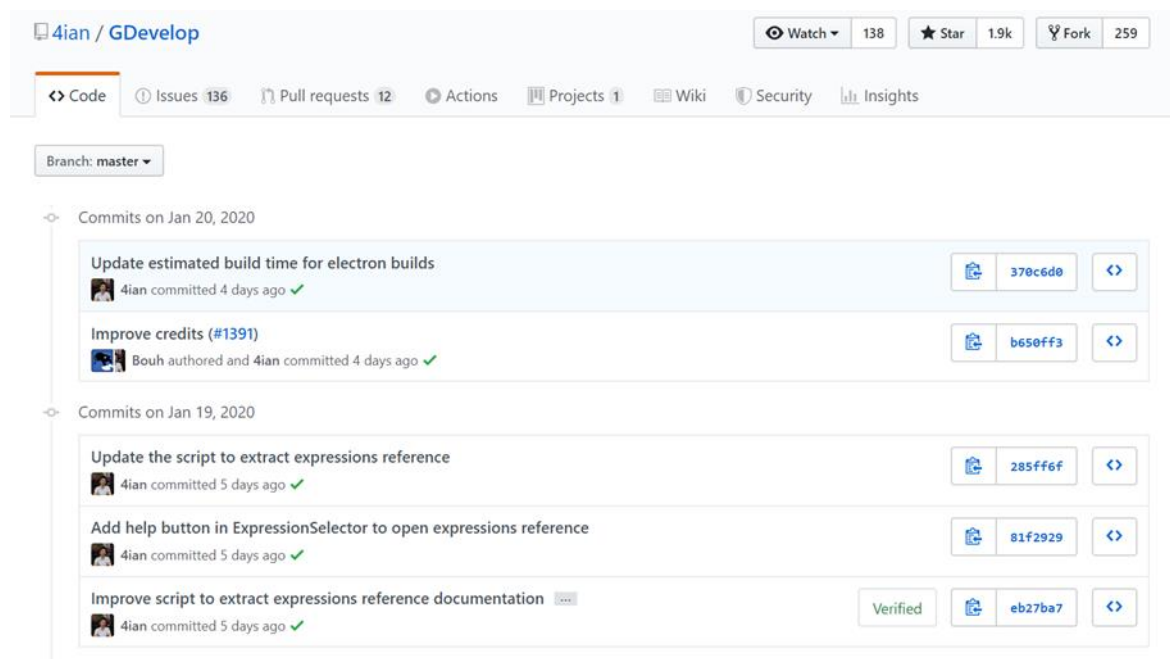


Źródło: Opracowanie własne

4. Repozytorium na platformie GitHub. Platforma GDevelop opiera się od początku swojego istnienia na formie projektu open-source co oznacza, że kod oprogramowania jest dostępny publicznie i może być edytowany przez każdego. Główny projekt znajduje się pod adresem <https://github.com/4ian/GDevelop>, a główni twórcy zachęcają do czynnego udziału w tworzeniu oprogramowania. Na stronie głównego repozytorium znajdują się wszelkie zmiany dotyczące platformy, a także nowe funkcjonalności, które będą

wprowadzone w kolejnych wersjach programu. Projekt podzielony jest na gałęzie (ang. branch) gdzie każda gałąź odpowiada za osobny moduł programu.

### Rysunek 13 GitHub GDevelop

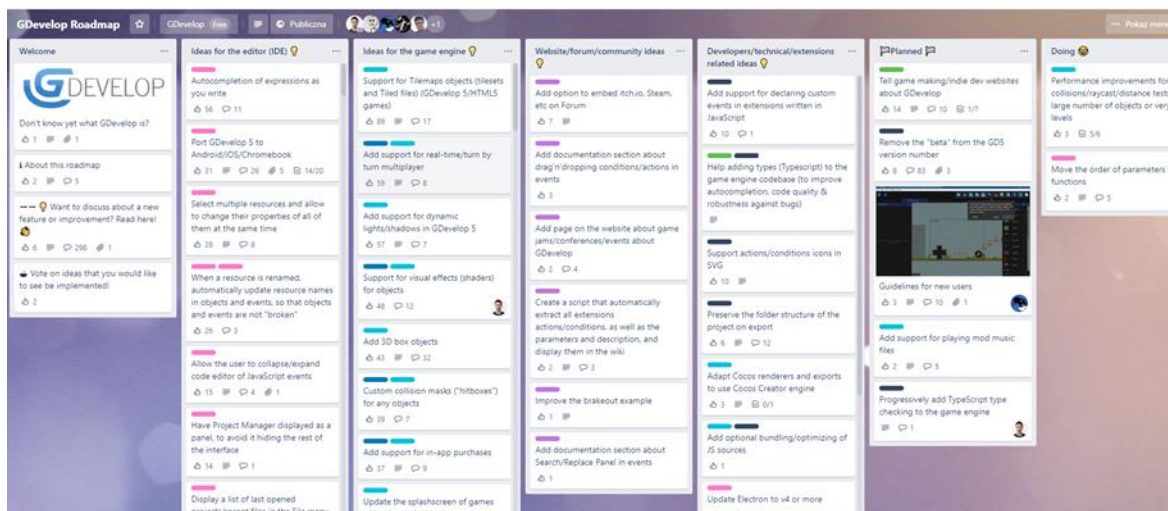


Źródło: Opracowanie własne

5. Kolejnym kanałem informacyjnym dostarczającym przez twórców jest tablica Trello pod adresem <https://trello.com/b/qf0lM7k8/gdevelop-roadmap>. Jest to miejsce gdzie twórcy dzielą się swoimi pomysłami i poczynaniami w zakresie tworzenia oprogramowania tym samym zachowując swoistą dokumentację procesu tworzenia konkretnych funkcjonalności. Dodatkowo możemy obserwować listę zmian wprowadzanych w kolejnych wersjach programu z podziałem na konkretne funkcjonalności. Większość z zadań posiada odnośniki do właściwych tematów na forum, platformie github i innych kanałach informacyjnych.



**Rysunek 14 Tablica Trello projektu GDevelop**

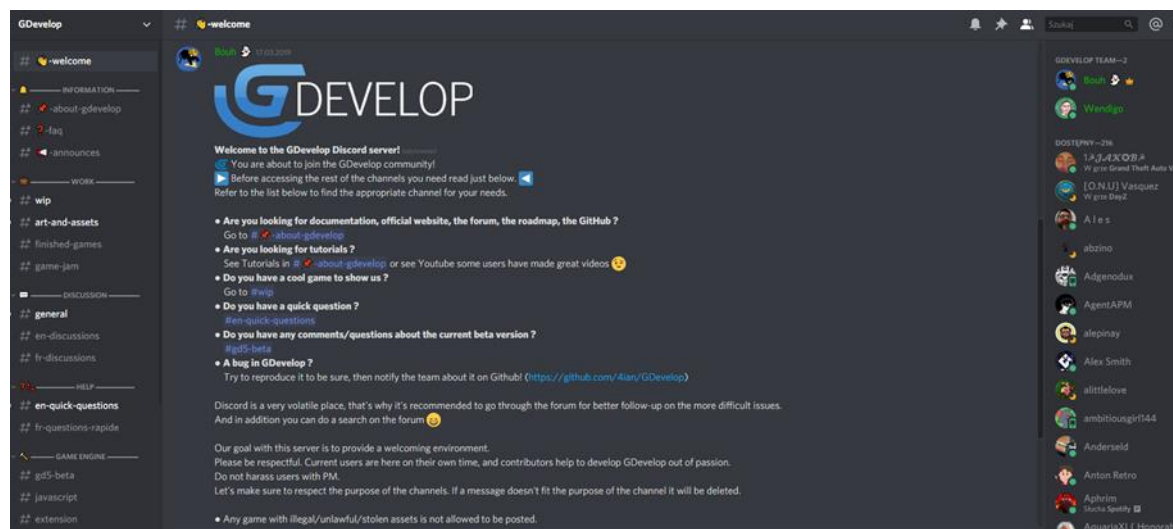


Źródło: Opracowanie własne

6. Discord, czat popularnego komunikatora Discord dostępny na wielu platformach również posiada dedykowany kanał stworzony przez twórców GDevelop. Kanał poświęcony GDevelop znajduje się pod adresem: <https://discordapp.com/channels/258623956158906368/555883550042488852>. Czat służy jako proste narzędzie do komunikacji na żywo pomiędzy użytkownikami narzędzia GDevelop. Użytkownicy czatu mogą rozmawiać na czacie tekstowym lub głosowym i tym samym dzielić się wiedzą, pomagać w zdobyciu potrzebnych informacji i rozwiązywać najbardziej irytujące problemy za pomocą burzy mózgów w jak najkrótszym czasie. Obecnie czat zrzesza kilkuset użytkowników w tym samych twórców, którzy też często udzielają się na poszczególnych kanałach. Sama budowa czatu jest prosta i przejrzysta. W ramach jednego kanału tworzy się drzewo mniejszych komórek, gdzie możemy przypinać najistotniejsze informacje i prowadzić rozmowy w ramach jednego tematu. Oprócz tego użytkownicy mogą zakładać własne konwersacje w zależności od potrzeb.



**Rysunek 15 Discord**



Źródło: Opracowanie własne

7. Treści tworzone przez użytkowników. Jedną z metod dzielenia się informacjami w Internecie są blogi tematyczne w przypadku narzędzia GDevelop jest identycznie. Istnieją blogi tematyczne tworzone przez pasjonatów, którzy chętnie dzielą się wiedzą związaną z tworzeniem gier za pomocą narzędzia GDevelop. Blogi i strony mają różne formy i w zależności od autora zawierają różne typy treści choć zdecydowana większość to poradniki w postaci pisemnej lub nagrania video poniżej znajdują się przykładowe witryny z poradnikami:

- <https://gametemplates.itch.io/gdevelop-examples>,
- <https://dev.end3r.com/2019/11/gdevelop-game-tutorial-for-season-of-docs-project-report/>,
- <https://www.c-sharpcorner.com/technologies/games-programming-csharp>,
- <https://www.gamefromscratch.com/post/2018/09/24/GDevelop-5.aspx>.

Oprócz tego w serwisie YouTube można znaleźć twórców, którzy udostępniają pełne poradniki, w których opisują postęp swoich projektów w postaci nagrań przykładowe kanały z poradnikami udostępnione w serwisie GDevelop Wiki:

- [https://www.youtube.com/channel/UCdGwB40bg\\_Tgg2lhx7FbOkw/](https://www.youtube.com/channel/UCdGwB40bg_Tgg2lhx7FbOkw/),
- <https://www.youtube.com/channel/UCsZ4Ue8c94YLJDbGRafCI5Q/>,
- [https://www.youtube.com/channel/UCQ9LqJ9ZOT8aYr-by\\_TJoKQ](https://www.youtube.com/channel/UCQ9LqJ9ZOT8aYr-by_TJoKQ),
- <https://www.youtube.com/channel/UCGgUpzgP379FS1tu12zm3CQ/playlists>.

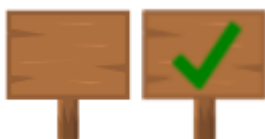
### 5.1. Argumentacja i przedstawienie podstawowej szaty graficznej gry

Podczas przygotowywania szaty graficznej bez której nie było możliwości rozpoczęcia procesu tworzenia gry w programie GDevelop, grupa projektowa kierowała się głównie trzema ustalonymi przez siebie aspektami:

- cena,
- licencja,
- styl wykonania.

Większość pozyskanych elementów szaty graficznej pobrane zostało z sieci Internet. Wszystkie użyte w projekcie grafiki są udostępniane w ramach licencji otwartych tzn. za darmo z możliwością ich edycji i wykorzystania w projektach komercyjnych. Część grafik została edytowana za pomocą programu GIMP w celu przystosowania wyglądu do spełnianych przez nie funkcji. Przykład dostosowanej do potrzeb grafiki to drewniany znak, który po modyfikacji został wykorzystany do funkcji *Checkpoint*.

#### Rysunek 16 Zmodyfikowana grafika



Źródło: Opracowanie własne

Zgodnie z przygotowanym wcześniej opisem fabuły gry, jako główną postać wybrana została grafika prezentująca łucznika.

#### Rysunek 17 Grafika przedstawiająca główną postać gry



Źródło: <https://www.gamedevmarket.net/asset/fantasy-heroes-character-sprite-sheet-10156/>

W pobranym pakiecie dostępne były animacje poruszania się postaci (bieg, skok, atak, śmierć itp.), które umożliwiły pełną obsługę zachowań bohatera. Kolejnymi grafikami do

wyselekcjonowania okazały się elementy otoczenia tj. platformy po których poruszać się będą postacie oraz tła do scen przedstawiające las lub dżunglę.

Do wyświetlania scenerii gry zaimplementowano prostą funkcję imitującą głębie otoczenia (obraz trójwymiarowy) za pomocą dwuwymiarowych obiektów (efekt paralaksy). Tło zostało podzielone na kilka warstw przedstawiających różne poziomy głębokości otoczenia. Podczas rozgrywki, gdy bohater gry porusza się, każda z warstw przesuwa się z odmienną prędkością. Taki zabieg powoduje odczucie u gracza efektu głębi tła. Kolejnym etapem kolekcjonowania potrzebnych grafik do wykonania gry było znalezienie obiektów przedstawiających przeciwników. Spośród wielu dostępnych możliwości wyselekcjonowano animacje dla trzech głównych antagonistów nazywanych w grze *Boss'ami*, oraz kilka grafik dla mniej znaczących wrogów.

#### **Rysunek 18 Przykładowa grafika przedstawiająca głównego przeciwnika**



Źródło: <https://craftpix.net/freebies/free-fallen-angel-chibi-2d-game-sprites/>

Podczas tworzenia projektu potrzebne okazało się także stworzenie własnych autorских grafik, które wykorzystane zostały m.in. w scenach *Menu* lub *Info*. W grze występuje element nazywany *GUI*<sup>4</sup>, na którym wyświetlane są dla gracza najważniejsze informacje tj. ilość pozostałych żyć oraz suma zdobytych punktów. Elementy te reprezentowane są poprzez ikonę serca oraz monety.

#### **Rysunek 19 Ikony umieszczone w GUI**



Źródło: <https://kenney.nl/assets/platformer-pack-redux>

Ostatnimi elementami występującymi w grze są przedmioty, które gracz może zbierać podczas rozgrywki. Przedmioty te mają wpływ na to co jest wyświetlane w *GUI*.

---

<sup>4</sup> Graficzny interfejs użytkownika (z ang. *Graphical User Interface*) - jest to jeden ze sposobów wykonywania interakcji z użytkownikiem w systemach informatycznych.

### 6.1. Opis scen zawartych w grze

Projekt składa się z trzech w pełni grywalnych poziomów oraz sześciu scen które uzupełniają rozgrywkę. Sceny są podzielone na opisane sekcje z zaimplementowaną logiką gry. Poziom grywalny jest to część w której gracz ma możliwość sterowania głównym bohaterem gry i wykorzystywania utworzonych mechanik rozgrywki. Grywalne poziomy wykorzystują zdarzenia zewnętrzne, które zostaną opisane w kolejnym załączniku. Sceny nie grywalne są to plansze w których m.in. gracz jest informowany o przebiegu rozgrywki, uzyskuje możliwość rozpoczęcia, wyjścia, zapisu oraz wczytania poprzednio zapisanego stanu rozgrywki. Projekt *Armin Legends* zawiera sceny:

1) Nie grywalne:

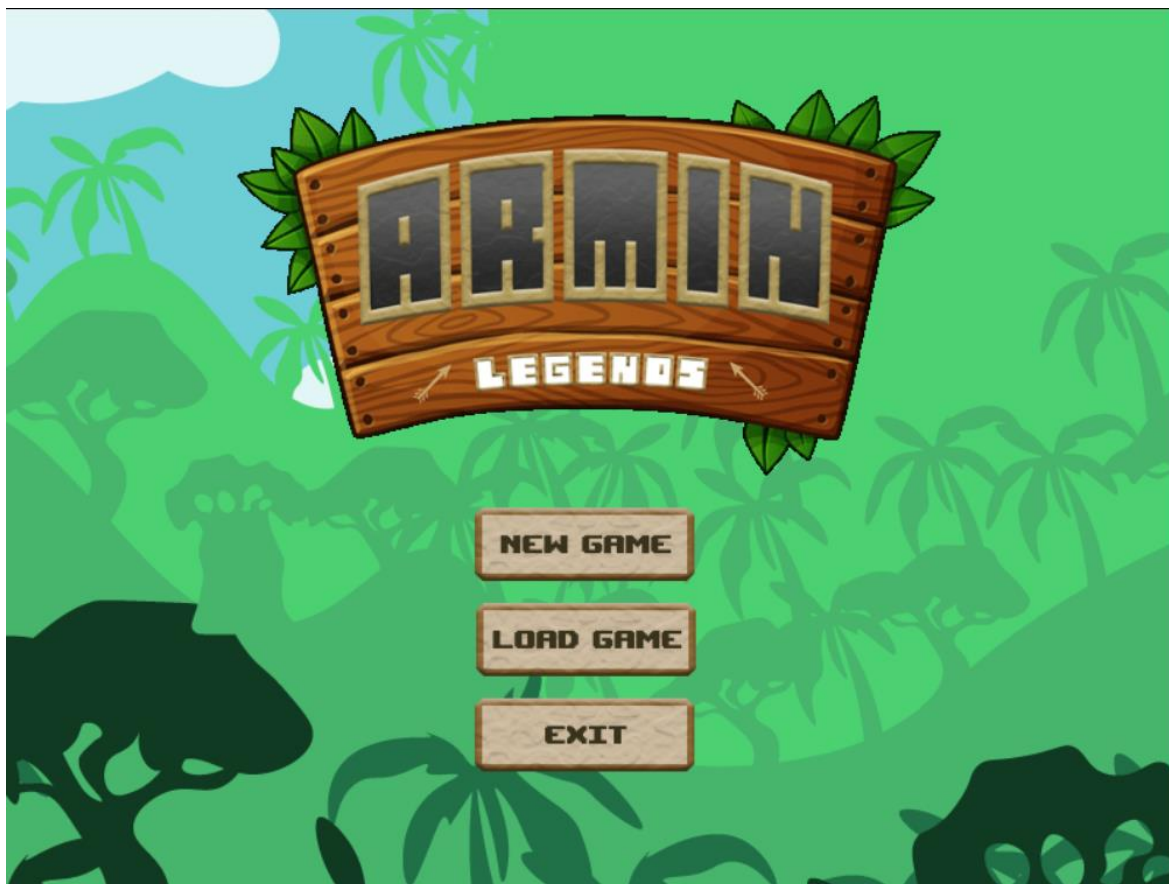
- a. *Menu*,
- b. *Story*,
- c. *Info*,
- d. *MenuInGame*,
- e. *GameOver*,
- f. *Win*.

2) Grywalne:

- a. *1* - pierwszy poziom rozgrywki,
- b. *2* - drugi poziom rozgrywki,
- c. *3* - ostatni poziom rozgrywki.

Po uruchomieniu gry jako pierwsza zostaje wyświetlona scena *Menu*. Pozwala na rozpoczęcie nowej rozgrywki, wczytanie ostatniego zapisanego stanu gry lub wyjście.

Rysunek 20 Menu w grze



Źródło: Opracowanie własne

Rysunek 21 Funkcje przycisków - Menu

Funkcje przycisków	
<p>🕒 The timer "Button_timer_start" is greater than 1 seconds Add condition</p>	<p>✗ Delete storage "progress"</p> <p>🕒 Delete timer "Button_timer_start" from memory</p> <p>🔧 Change the global variable (📄) Lives: set to 3</p> <p>🔧 Change the global variable (📄) Loaded: set to 0</p> <p>➡ Change to scene "Story"</p> <p>Add action</p>
<p>🕒 The timer "Button_timer_load" is greater than 1 seconds Add condition</p>	<p>🕒 Delete timer "Button_timer_load" from memory</p> <p>Add action</p>
<p>🕒 The timer "Button_timer_exit" is greater than 1 seconds Add condition</p>	<p>🕒 Delete timer "Button_timer_exit" from memory</p> <p>🗑 Close structured data "progress"</p> <p>🚪 Quit the game</p> <p>Add action</p>

Źródło: Opracowanie własne

Logika planszy składa się z trzech sekcji opisanych poniżej:

- *Ustawienia i resetowanie opóźnienia działania przycisków* - uruchamia ciągłą zmianę przezroczystości dla przycisków dostępnych na planszy. Po załadowaniu sceny zostają uruchomione czasomierze przypisane do przycisków. Odtworzona

zostaje ścieżka dźwiękowa oraz wczytuje się magazyn danych potrzebny do uzyskania informacji o stanie rozgrywki,

- *Działanie przycisków* - zaprogramowano reakcje przycisków na kolizję z kursorem i ich działanie po naciśnięciu lewego przycisku myszy,
- *Funkcje przycisków* - sekcja ta opisuje reakcje gry na naciśnięcie danego przycisku.

Następna scena zawarta w projekcie to *Story* na której zostaje przedstawiona graczowi fabuła gry. Plansza wyświetlana jest tylko w przypadku rozpoczęcia gry od nowa. Scena składa się z trzech sekcji.

- *Po załadowaniu sceny* - wyświetlona zostaje fabuła gry, mająca na celu zaciekawienie gracza i wprowadzenie do rozgrywki,
- *Efekt piszącego się tekstu* - zaimplementowano funkcję imitującą pisanie tekstu w czasie rzeczywistym,
- *Przejsie do następnej sceny* - po spełnieniu ustalonego warunku realizowane jest przejście do następnej sceny.

Kolejnym elementem gry jest *Info*. Scena opisuje graczowi sposób sterowania głównym bohaterem gry oraz przedstawia najważniejsze aspekty rozgrywki.

Rysunek 22 Przedstawienie sceny Info



Źródło: Opracowanie własne

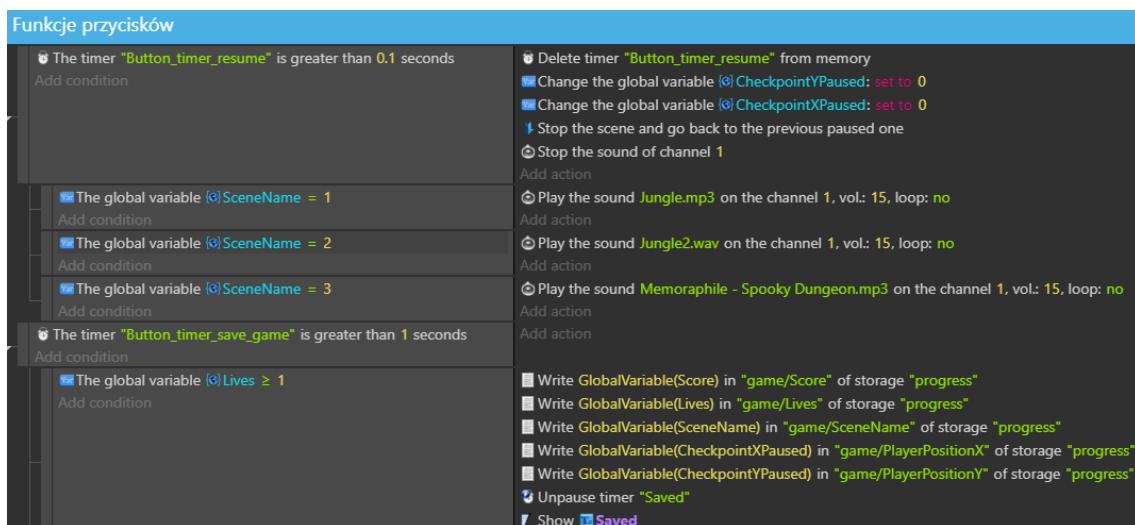
Scena składa się z trzech sekcji:

- *Odnośniki do zdarzeń zewnętrznych* - dołączone zostały zdarzenia zewnętrzne których opis znajduje się w załączniku 6.2.,
- *Po załadowaniu sceny* - wyzwolenie czasomierza odliczającego czas który musi upłynąć aby możliwe było przejście do kolejnej sceny,
- *Przejdźcie do następnej sceny* - po spełnieniu warunków gra przenosi rozgrywkę do sceny 1.

Plansza *MenuInGame* wyzwolana jest podczas trwania rozgrywki gdy gracz naciśnie przycisk *ESC* na klawiaturze. Pozwala ona na zapis stanu gry, przejście do planszy *Menu* lub całkowite zamknięcie gry.



## Rysunek 23 Funkcje przycisków - MenuInGame

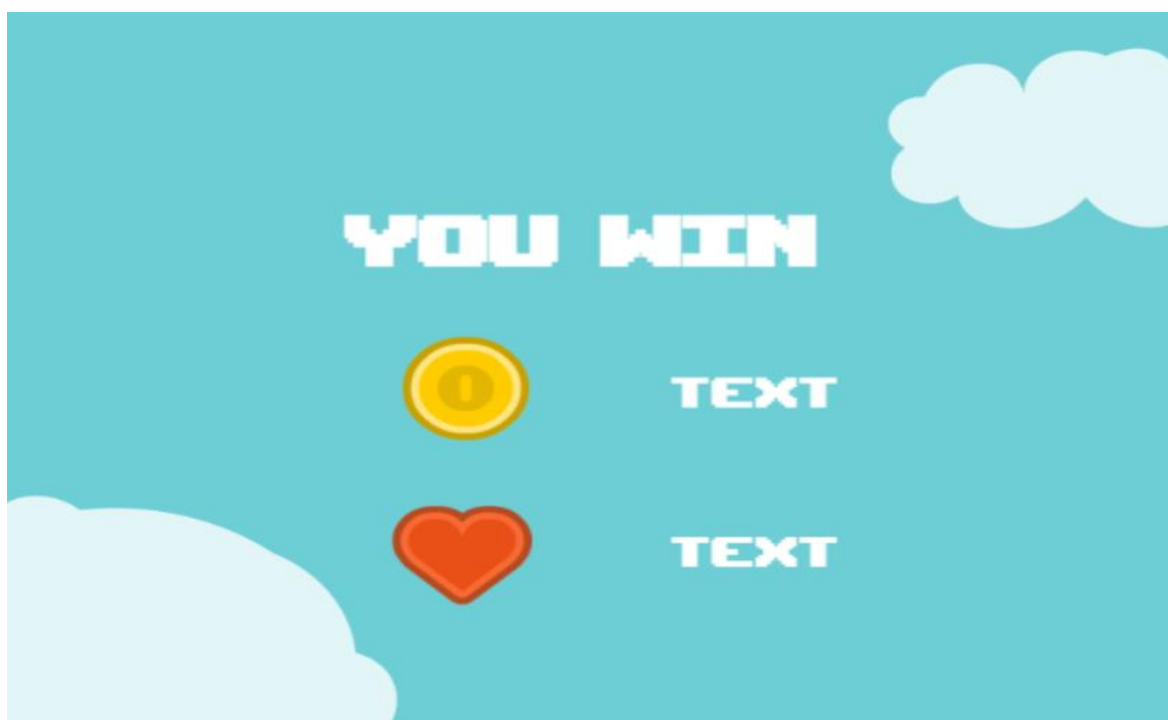


Źródło: Opracowanie własne

Sposób zaimplementowania logiki planszy opiera się na scenie *Menu*. Dodatkowymi elementami są zdarzenia odpowiedzialne za powrót do wstrzymanej rozgrywki umieszczone w sekcji *Powrót do poprzedniej - zatrzymanej sceny* oraz obsługa zapisu stanu gry umieszczone w sekcji *Funkcje przycisków*.

Plansze *Win* i *GameOver* wyświetlają się odpowiednio w sytuacji wygrania gry lub porażki.

## Rysunek 24 Plansza YouWin



Źródło: Opracowanie własne

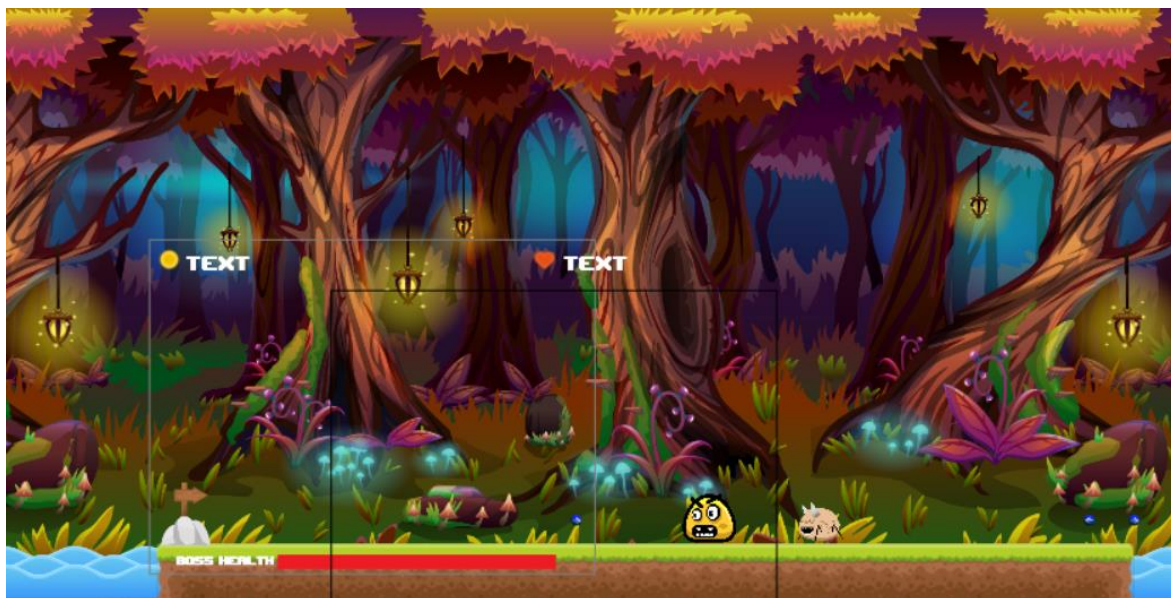


*Win* wyświetla uzyskany wynik, ilość pozostałych żyć, resetuje zmienne globalne do ustawień początkowych oraz przechodzi do *Menu*.

*GameOver* wyświetla uzyskany wynik w ciągu całej rozgrywki do momentu utracenia wszystkich żyć oraz przechodzi do *Menu*.

Wszystkie plansze scen grywalnych składają się z trzech podobnych do siebie sekcji. Różnice wynikają jedynie z nazewnictwa poszczególnych obiektów umieszczonych w zasobach. Scena 3 posiada dodatkową sekcję obsługującą specjalne wydarzenia związane z występującym jedynie w niej pojedynku bohatera z głównym antagonistą, którego następstwem jest koniec gry. Przykładowy wygląd planszy grywalnej (Rysunek 25).

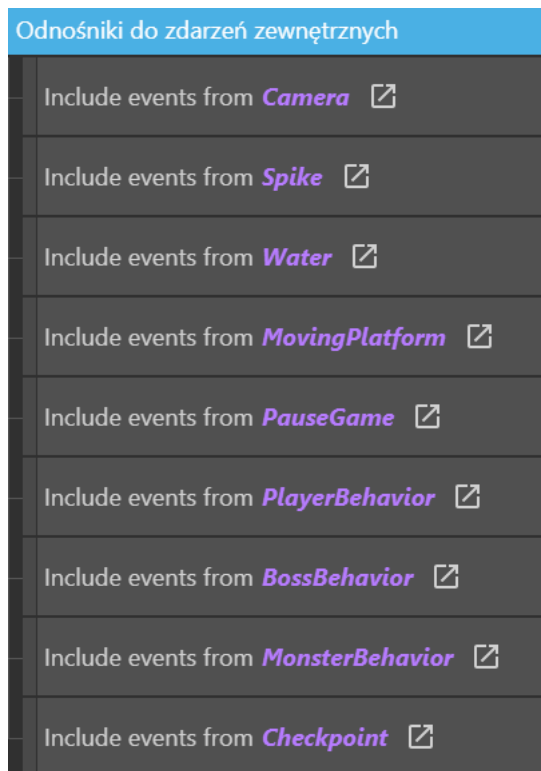
**Rysunek 25** Przedstawienie przykładowej planszy grywalnej



Źródło: Opracowanie własne

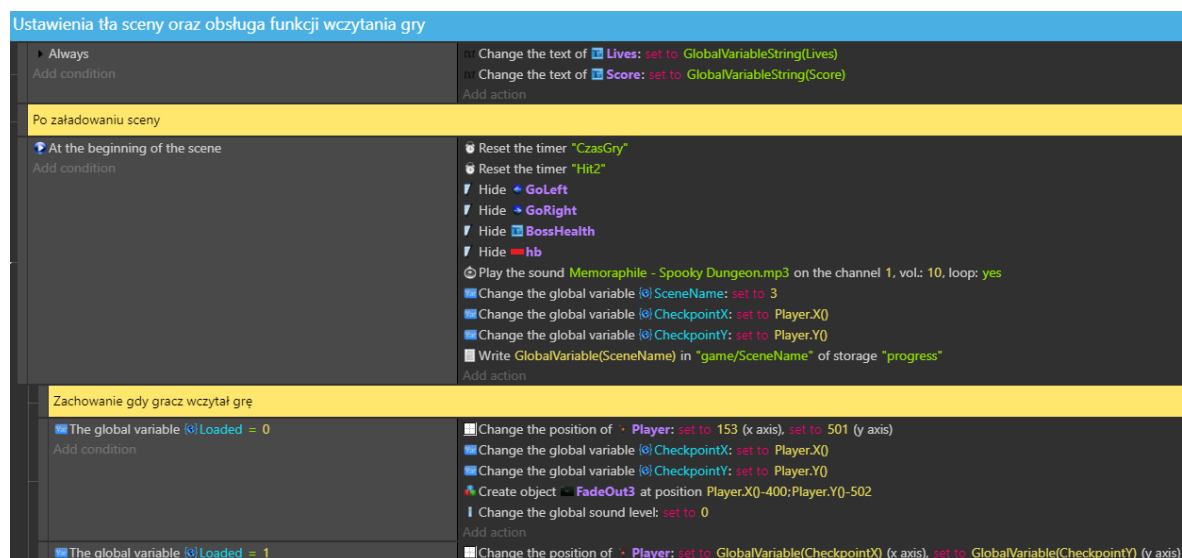
Logika plansz grywalnych opiera się na odnośnikach do zdarzeń zewnętrznych, akcji wykonywanych przez grę po załadowaniu sceny oraz funkcji zmiany sceny na kolejną po spełnieniu określonych warunków.

**Rysunek 26 Zdarzenia zewnętrzne**



Źródło: Opracowanie własne

**Rysunek 27 Ustawienia tła sceny oraz obsługa wczytania gry – Scena 3**



Źródło: Opracowanie własne

Po rozpoczęciu planszy ukrywane są elementy służące do sterowania poruszaniem się przeciwników. Następnie gra sprawdza czy użytkownik wykorzystał funkcję załadowania poprzednio zapisanego stanu gry. Rozpoczynane jest odtwarzanie podkładu muzycznego. Drugim elementem jest *FadeOut*, który służy do wyświetlenia nazwy sceny po jej

załadowaniu. W planszach grywalnych zastosowano tylko część mechaniki *FadeOut*, główna definicja znajduje się w zdarzeniach zewnętrznych.

**Rysunek 28 Wyświetlanie tytułu sceny - Plansza 3**

Wyświetlanie tytułu sceny	
<ul style="list-style-type: none"> <li>The number of <b>FadeOut3</b> objects = 1</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Activate behavior <b>PlatformerObject</b> of <b>Player: no</b></li> <li>Hide <b>Coin</b></li> <li>Hide <b>Heart</b></li> <li>Hide <b>Score</b></li> <li>Hide <b>Lives</b></li> <li>Change the z-order of <b>FadeOut3</b>: set to 500</li> <li>Add action</li> </ul>
<ul style="list-style-type: none"> <li>The timer <b>"CzasGry"</b> is greater than 3 seconds</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Change the opacity of <b>FadeOut3</b>: subtract 95*TimeDelta()</li> <li>Add action</li> </ul>
<ul style="list-style-type: none"> <li>The opacity of <b>FadeOut3</b> = 0</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Delete <b>FadeOut3</b></li> <li>Show <b>Heart</b></li> <li>Show <b>Coin</b></li> <li>Show <b>Lives</b></li> <li>Show <b>Score</b></li> <li>Activate behavior <b>PlatformerObject</b> of <b>Player: yes</b></li> <li>Change the global sound level: set to 100</li> </ul>

Źródło: Opracowanie własne

Trzecim elementem jest mechanika zmieniania aren na kolejne. Jest ona analogiczna do mechanik wykorzystywanych w scenach nie grywalnych.

W arenie 3 dodatkową sekcją jest *Mechanika walki z głównymi przeciwnikami*.

**Rysunek 29 Mechanika walki z głównymi przeciwnikami - Scena 3**

Mechanika walki z głównymi przeciwnikami	
<ul style="list-style-type: none"> <li>The number of <b>Boss1</b> objects = 0</li> <li>The scene variable <b>WhichBoss</b> = 1</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Change the scene variable <b>WhichBoss</b>: add 1</li> <li>Create object <b>Boss2</b> at position 666;380</li> <li>Change the variable <b>Health</b> of <b>Boss2</b>: set to 150</li> <li>Change the height of <b>Boss2</b>: set to 200</li> <li>Change the width of <b>Boss2</b>: set to 198</li> <li>Add action</li> </ul>
<ul style="list-style-type: none"> <li>The number of <b>Boss2</b> objects = 0</li> <li>The scene variable <b>WhichBoss</b> = 2</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Change the scene variable <b>WhichBoss</b>: add 1</li> <li>Create object <b>Boss3</b> at position 666;380</li> <li>Change the variable <b>Health</b> of <b>Boss3</b>: set to 150</li> <li>Change the height of <b>Boss3</b>: set to 200</li> <li>Change the width of <b>Boss3</b>: set to 198</li> <li>Add action</li> </ul>
<ul style="list-style-type: none"> <li>The number of <b>Boss3</b> objects = 0</li> <li>The scene variable <b>WhichBoss</b> = 3</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Change the scene variable <b>WhichBoss</b>: add 1</li> <li>Add action</li> </ul>
Pokonany ostatni Boss	
<ul style="list-style-type: none"> <li>The scene variable <b>WhichBoss</b> = 4</li> <li>Add condition</li> </ul>	<ul style="list-style-type: none"> <li>Change the global variable <b>Loaded</b>: set to 0</li> <li>Change to scene <b>"Win"</b></li> </ul>

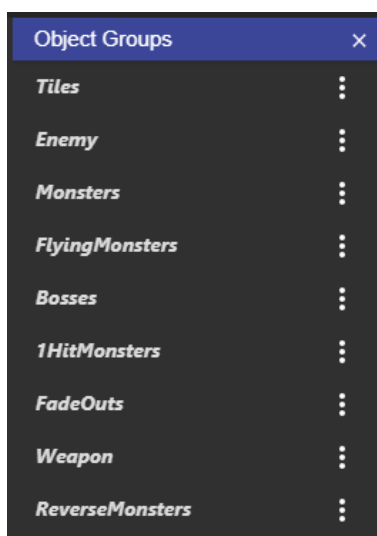
Źródło: Opracowanie własne

Podczas rozgrywania ostatniej planszy gracz ma za zadanie pokonać trzech przeciwników głównych. Antagoniści zostają przyzywani pojedynczo tj. gdy jeden zostaje pokonany, następny staje do dalszej walki.

## 6.2. Przedstawienie zdarzeń zewnętrznych zawartych w grze

Załącznik ten ukazuje efekty stworzenia mechaniki zdarzeń zewnętrznych wykorzystywanych w projekcie „Armin Legends”. W celu ujednolicenia kodu aplikacji, który wykorzystywany jest w różnych scenach utworzono zdarzenia zewnętrzne. Działanie to pozwoliło na ominięcie błędnego zachowywania się elementów umieszczonych w grze, które mogło być spowodowane powielającym się kodem w wielu miejscach. Taki zabieg zapewnił uzyskanie większej przejrzystości kodu, co ułatwiło pracę oraz zmniejszyło liczbę powiązanych ze sobą w skomplikowany sposób problemów. Dodatkowo utworzono specjalne grupy obiektów o tych samych właściwościach, aby w kodzie posługiwać się za pomocą nazw grup, a nie pojedynczych elementów.

**Rysunek 30 Grupy obiektów**



Źródło: Opracowanie własne

Zdarzenia zewnętrzne zaimplementowane w projekcie to:

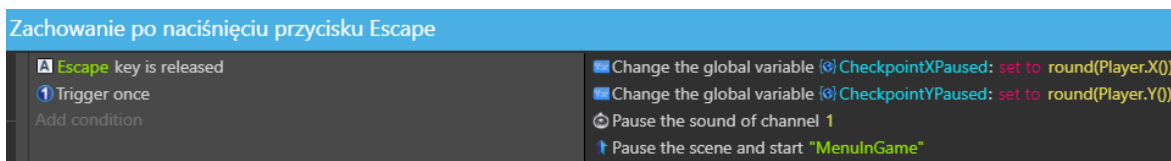
- *PauseGame*,
- *PlayerBehavior*,
- *Camera*,
- *BossBehavior*,
- *Checkpoint*,
- *MonsterBehavior*,
- *Water*,

- *Spike*,
- *MovingPlatform*.

Można je podzielić na zdarzenia powiązane z otoczeniem i na zdarzenia powiązane z postaciami występującymi w grze.

- 1) *PauseGame* służy do wstrzymywania aktualnie rozgrywanej sceny i przechodzenia do planszy *MenuInGame*.

### Rysunek 31 Zachowanie po naciśnięciu przycisku Escape



Źródło: Opracowanie własne

- 2) *PlayerBehavior* to rozbudowane zdarzenie zewnętrzne zawierające wszystkie mechaniki dotyczące zachowania postaci oraz jej interakcji z otoczeniem. Zdarzenie to zawiera dziewięć sekcji:
  - *Zachowanie po rozpoczęciu sceny* - wyzwalane są czasomierze odpowiedzialne za śmierć bohatera oraz czas jaki musi upłynąć, aby można było ponownie zaatakować,
  - *Orientacja grafiki player oraz implementacja poruszania się bohatera gry* - wyświetlanie wszystkich animacji zaimplementowanych dla obiektu bohatera. Dodatkowo umieszczono mechanikę ataku z łuku głównej postaci,

Rysunek 32 Mechanika ataku głównej postaci

Postać atakuje	
<p>Touch or Left mouse button is down</p> <p>Add condition</p> <p>Add condition</p>	<p>Add action</p> <p>Create object <b>Hit</b> at position <code>MouseX0-5;MouseY0-5</code></p> <p>Change the height of <b>Hit</b>: set to 10</p> <p>Change the width of <b>Hit</b>: set to 10</p> <p>Hide <b>Hit</b></p> <p>Change the maximum speed of <b>Player</b>: set to 150</p> <p>Pause the sound of channel 2</p> <p>Add action</p> <p>Change the variable <b>IsShooting</b> of <b>Player</b>: set to 1</p> <p>Unpause timer "ShootingTimer" of <b>Player</b></p> <p>Set animation of <b>Player</b> to "Attack"</p> <p>Create object <b>Arrow</b> at position <code>Player.PointX("Shoot");Player.PointY("Shoot")</code></p> <p>Change the scale of <b>Arrow</b>: multiply by 0.5</p> <p>Rotate <b>Arrow</b> towards <code>Hit.X0;Hit.Y0</code> at speed 0deg/second</p> <p>Move <b>Arrow</b> to <b>Hit</b> with a permanent force of 500 pixels</p> <p>Play the sound <b>ArrowShoot.wav</b> on the channel 4, vol.: 15, loop: no</p>
<p>The variable <b>Flipped</b> of <b>Player</b> = 0</p> <p>Add condition</p>	<p>Add action</p> <p>Change the variable <b>IsShooting</b> of <b>Player</b>: set to 1</p> <p>Unpause timer "ShootingTimer" of <b>Player</b></p> <p>Set animation of <b>Player</b> to "Attack"</p> <p>Create object <b>Arrow</b> at position <code>Player.PointX("Shoot");Player.PointY("Shoot")</code></p> <p>Change the scale of <b>Arrow</b>: multiply by 0.5</p> <p>Rotate <b>Arrow</b> towards <code>Hit.X0;Hit.Y0</code> at speed 0deg/second</p> <p>Move <b>Arrow</b> to <b>Hit</b> with a permanent force of 500 pixels</p> <p>Play the sound <b>ArrowShoot.wav</b> on the channel 4, vol.: 15, loop: no</p>
<p>The variable <b>Flipped</b> of <b>Player</b> = 1</p> <p>Add condition</p>	<p>Add action</p> <p>Change the variable <b>IsShooting</b> of <b>Player</b>: set to 1</p> <p>Unpause timer "ShootingTimer" of <b>Player</b></p> <p>Set animation of <b>Player</b> to "Attack"</p> <p>Create object <b>Arrow</b> at position <code>Player.PointX("Shoot");Player.PointY("Shoot")</code></p> <p>Change the scale of <b>Arrow</b>: multiply by 0.5</p> <p>Rotate <b>Arrow</b> towards <code>Hit.X0;Hit.Y0</code> at speed 0deg/second</p> <p>Move <b>Arrow</b> to <b>Hit</b> with a permanent force of 500 pixels</p> <p>Play the sound <b>ArrowShoot.wav</b> on the channel 4, vol.: 15, loop: no</p>

Źródło: Opracowanie własne

- *Śmierć bohatera* - obsługa animacji śmierci bohatera gry, po której następuje zmniejszenie ilości dostępnych żyć,
- *Brak obiektu player na mapie* - opisuje sytuację w której na mapie brakuje obiektu *Player*,

Rysunek 33 PlayerBehavior - zdarzenia zewnętrzne

Śmierć bohatera	
<p>If all of these conditions are true:</p> <p>The variable <b>IsDead</b> of <b>Player</b> = 1</p> <p>The global variable <b>Lives</b> &gt; 0</p> <p>Add a sub-condition</p> <p>Add condition</p> <p>The timer "IsDeadTimer" of <b>Player</b> is greater than 2 seconds</p> <p>Add condition</p> <p>Trigger once</p> <p>Add condition</p>	<p>Set animation of <b>Player</b> to "Die"</p> <p>Pause the sound of channel 2</p> <p>Activate behavior <b>PlatformerObject</b> of <b>Player</b>: no</p> <p>Unpause timer "IsDeadTimer" of <b>Player</b></p> <p>Add action</p> <p>Delete <b>Player</b></p> <p>Change the global variable <b>Lives</b>: subtract 1</p> <p>Add action</p> <p>Play the sound <b>Death.wav</b> on the channel 5, vol.: 15, loop: no</p> <p>Add action</p>
Brak obiektu Player na mapie	
<p>If all of these conditions are true:</p> <p>The number of <b>Player</b> objects = 0</p> <p>The global variable <b>Lives</b> &gt; 0</p> <p>Add a sub-condition</p> <p>Add condition</p>	<p>Delete <b>Weapon</b></p> <p>Create object <b>Player</b> at position <code>GlobalVariable(CheckpointX);GlobalVariable(CheckpointY)</code></p> <p>Change the height of <b>Player</b>: set to 150</p> <p>Change the width of <b>Player</b>: set to 150</p> <p>Change the z-order of <b>Player</b>: set to 10</p> <p>Change the variable <b>IsDead</b> of <b>Player</b>: set to 0</p>

Źródło: Opracowanie własne

- *Kolizja bohatera z sercem lub monetą* - podczas rozgrywki możliwe są do zdobycia dodatkowe punkty życia i punkty wyniku, które przedstawione są w postaci

serc i monet. Podczas kolizji obiektu *Player* z jednym z tych elementów do zmiennej globalnej dodawane są określone punkty,

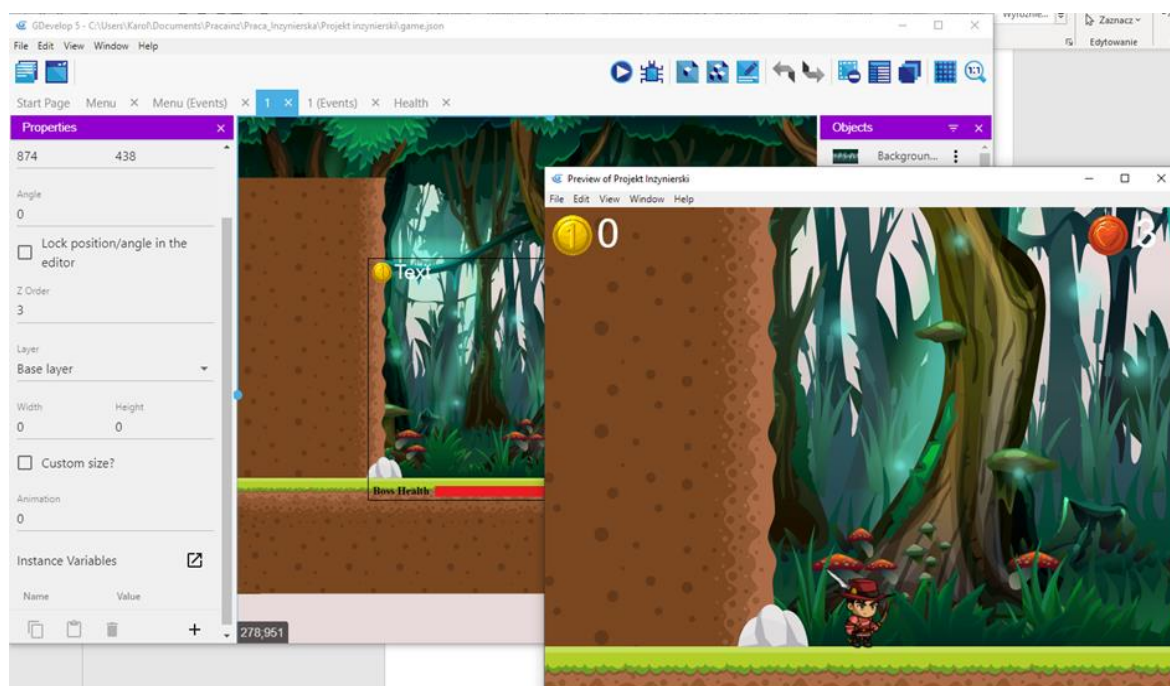
- *Kolizja bohatera z przeciwnikami* - przy kolizji z przeciwnikiem odejmy zostaje punkt życia, a postać aktywuje wydarzenia z sekcji *Śmierć bohatera*,
  - *Zmienna globalna Lives jest równa 0* - sprawdzanie czy aktualna wartość zmiennej globalnej *Lives* jest mniejsza niż zero. W przypadku potwierdzenia następuje zmiana sceny na *GameOver*,
  - *Zachowanie strzały po kontakcie z otoczeniem* - obsługa przedmiotu za pomocą którego atakuje postać główna. W przypadku kolizji z otoczeniem, obiekt zostaje usunięty,
  - *Zabijanie przeciwników poprzez upadanie* - implementacja przypadku w którym obiekt *Player* jest w trakcie opadania i w tym czasie nastąpi kolizja z przeciwnikiem. W tym przypadku przeciwnik zostanie uśmiercony.
- 3) Zdarzenie zewnętrzne *Camera* wprowadza do projektu funkcję sztucznego tła trójwymiarowego. Efekt ten został opisany w załączniku 5.1.
  - 4) *BossBehavior* sekcja opisująca zachowanie głównych przeciwników. Definiuje ona w jaki sposób postacie będą atakować głównego bohatera.
  - 5) *Checkpoint* służy do obsługi zapisu stanu gry do utworzonej komórki pamięci *progress*. Wywoływana jest w trakcie kolizji głównego bohatera z obiektem *Checkpoint*.
  - 6) *MonsterBehavior* opisuje sposób poruszania się wszystkich przeciwników występujących w grze, a także ich kolizje z obiektem *Arrow* i wiążącą się z tą kolizją śmierć przeciwnika.
  - 7) *Water* to zdarzenie opisujące sytuację w której dochodzi do zderzenia się dwóch obiektów *Player* oraz *Tile10*.
  - 8) *Spike* określa sposób zachowania wysuwającego się kolca tj. czas w którym jest wysunięte i czas w którym jest schowane.
  - 9) *MovingPlatform* definiuje zachowanie poruszających się drewnianych bali, służących do przemieszczania się postaci m.in. nad wodą lub żeby dostać się do wyżej, niedostępnych dla gracza miejsc.



### 7.1. Proces testowania produktu projektowego

Testy oprogramowania odbywały się na każdym etapie fazy implementacji gry. Każda funkcjonalność w grze po jej zaimplementowaniu została przetestowana przez wbudowane w środowisko GDevelop narzędzie symulatora. Symulator umożliwia skompilowanie bieżącej sceny do aplikacji wykonywalnej w której to jesteśmy w stanie sprawdzić funkcjonalność podczas rozgrywki w ramach danej sceny (Rysunek 34).

**Rysunek 34** Testowanie działania konkretnej sceny przy pomocy narzędzia do symulacji



Źródło: Opracowanie własne

Oprócz tego po ukończeniu większych faz implementacji gra była eksportowana na platformy docelowe, aby ponownie przeprowadzić testy funkcjonalne tym razem na „żywym organizmie” czyli docelowym środowisku gry (Rysunek 35).

**Rysunek 35 Test funkcjonalny konkretnej wersji gry na platformie Windows**



Źródło: Opracowanie własne

## 7.2. Raport błędów i opis sposobu ich rozwiązania

### 1. Założenia

Błędy które zostały wychwycone w odpowiednich fazach testów zostały opisane biorąc pod uwagę kryteria takie jak:

- opis błędu,
- przyczyna błędu,
- proponowane rozwiązanie,
- skala problemu,
- przewidywane trudności w rozwiązaniu.

Dodatkowo, skala problemu która wyznacza rangę problemu może przybrać następujące wartości:

- problem globalny - kolor czerwony - najpoważniejsze problemy, wpływające na całą rozgrywkę. Należy je rozwiązywać w pierwszej kolejności,

- problem o zakresie grupy obiektów lub typu obiektów - kolor żółty - problemy średniej skali których zakres działania obejmuje grupy obiektów lub dany typ, którego łączy wspólna cecha lub zmienna np. wszystkie elementy otoczenia,
- problem o zakresie pojedynczego obiektu - kolor zielony - często mało znaczący problem (w zależności od instancji danego obiektu) odnoszący się do konkretnego obiektu lub jego konkretnej instancji powinien zostać rozpatrzony po uprzednim uporaniu się z problemami o wyższych rangach.

## 2. Raport z przeprowadzonych testów i wstępna analiza

### Rysunek 36 Raport i Analiza znalezionych błędów

Opis błędu w funkcjonalności	Przyczyna	Proponowane rozwiązanie	Skala problemu	Przewidywana trudność w rozwiązaniu
Gracz strzelając z łuku zabija wszystkich przeciwników w linii strzału	Obiekt strzały nie znika, lecz robi się niewidzialny po zderzeniu z przeciwnikiem i zabija kolejnych na linii jego strzału	Utworzyć warunek który będzie wymuszał usunięcie obiektu po kolizji z przeciwnikiem	Globalna	Należy zagnieździć warunek znikania strzały w zdarzeniu przy kolizji, ponieważ obiekt strzały może zostać usunięty ale dopiero gdy zostaną zadane obrażenia
Animacja ranienia przeciwnika nie działa prawidłowo. Postać przeciwnika zatrzymuje się w miejscu wykonując animację zranienia	Animacja powiązana ze zdarzeniem kolizji ze strzałą nie jest podmieniana na inną	Stworzyć czasomierz który załącza się po trafieniu przeciwnika strzałą o czasie równym animacji. Po tym czasie postać wraca do normalnego zachowania	Dotyczy potworów które nie giną po jednym trafieniu	Należy zagnieździć warunek czasomierza w odpowiednim miejscu ponieważ warunek musi zostać spełniony dopiero po trafieniu strzałą
Po upadku na potwory których nie można zabić skacząc na nie (np. Boss) postać odbija się zamiast ginąć	Grupa potworów podatna na skok nie została ustalona a warunek jest globalny	Dodać grupę obiektów (potworów) odpornych na skok i zawrzeć ją w odpowiednich miejscach zdarzeń	Dotyczy potworów które nie giną po jednym trafieniu	Należy zachować ostrożność przy przypisywaniu grup obiektów do zdarzeń gdyż dany warunek będzie miał zastosowanie dla wszystkich z tej grupy. Być może należy rozważyć stworzenie jeszcze mniejszych grup
Postać bohatera głównego lewituje na krawędzi obiektu kamień	Niepoprawnie dobrany HitBox	Zmodyfikowanie hitboxa mając na uwadze kształt kamienia	Dotyczy konkretnego obiektu	Należy uważać aby nie wyciąć hitboxa poza ramkę kamienia gdyż postać wyląduje w "środku" obiektu
Animacja bossa wykonuje się bardzo szybko niewspółmiernie do prędkości poruszania się	Niepoprawny czas animacji	Zmodyfikować czas trwania animacji mając na uwadze optymalizację animacji	Dotyczy konkretnego obiektu	Należy dobrać odpowiedni parametr czasowy wykonywania aplikacji

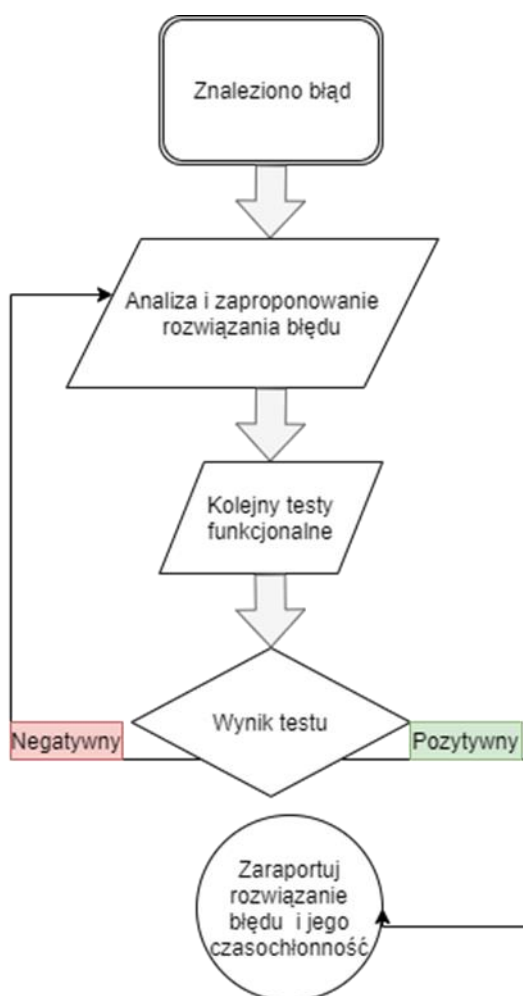
Źródło: Opracowanie własne

### 7.3. Wdrożenie poprawek, raport z pomyślnie przeprowadzonych testów

Po zdiagnozowaniu i analizie problemów w oprogramowaniu należy przystąpić do ich rozwiązywania. Po implementacji rozwiązania należy je przetestować, aby sprawdzić, czy rozwiązanie przyniosło oczekiwany rezultat. Dalsze kroki są zależne od wyniku przeprowadzonego testu tj.:

- jeżeli wynik testu jest pozytywny należy oznaczyć problem jako rozwiązany i oszacować czasochłonność rozwiązywania problemu,
- jeżeli wynik testu jest negatywny należy zaplanować i wdrożyć nowe rozwiązanie i ponownie przetestować. (krok powinien być powtarzany do momentu pozytywnego zakończenia testu).

**Rysunek 37 Procedura eliminowania błędów**



Źródło: Opracowanie własne

#### 4. Raport z rozwiązywania błędów

##### Rysunek 38 Raport z rozwiązywania błędów

Opis błędu w funkcjonalności	Przyczyna	Proponowane rozwiązanie	Skala problemu	Przewidywana trudność w rozwiązaniu	Czy problem został rozwiązany?	Wynik ponownego testu	Czasochłonność podana w godzinach
Gracz strzelając z łuku zabija wszystkich przeciwników w linii strzału	Obiekt strzały nie znika, lecz robi się niewidzialny po zderzeniu z przeciwnikiem i zabija kolejnych na linii jego strzału	Utworzyć warunek który będzie wymuszał usunięcie obiektu po kolizji z przeciwnikiem	Globalna	Należy zagnieździć warunek znikania strzały w zdarzeniu przy kolizji, ponieważ obiekt strzały może zostać usunięty ale dopiero gdy zostaną zadane obrażenia	tak	pozytywny	1
Animacja ranienia przeciwnika nie działa prawidłowo. Postać przeciwnika zatrzymuje się w miejscu wykonując animację zranienia	Animacja powiązana ze zdarzeniem kolizji ze strzałą nie jest podmieniana na inną	Stworzyć czasomierz który załącza się po trafieniu przeciwnika strzałą o czasie równym animacji. Po tym czasie postać wraca do normalnego zachowania	Dotyczy potworów które nie giną po jednym trafieniu	Należy zagnieździć warunek czasomierza w odpowiednim miejscu ponieważ warunek musi zostać spełniony dopiero po trafieniu strzałą	tak	pozytywny	1
Po upadku na potwory których nie można zabić skacząc na nie (np. Boss) postać odbija się zamiast ginać	Grupa potworów podatna na skok nie została ustalona a warunek jest globalny	Dodać grupę obiektów (potworów) odpornych na skok i zawrzeć ją w odpowiednich miejscach zdarzeń	Dotyczy potworów które nie giną po jednym trafieniu	Należy zachować ostrożność przy przypisywaniu grup obiektów do zdarzeń gdyż dany warunek będzie miał zastosowanie dla wszystkich z tej grupy. Być może należy rozważyć stworzenie jeszcze mniejszych grup	tak	pozytywny	0,5
Postać bohatera głównego lewituje na krawędzi obiektu kamień	Niepoprawnie dobrany HitBox	Zmodyfikowanie hitboxa mając na uwadze kształt kamienia	Dotyczy konkretnego obiektu	Należy uważać aby nie wyciąć hitboxa poza ramkę kamienia gdyż postać wylądowałaby w "środku" obiektu	tak	pozytywny	0,25
Animacja bossa wykonuje się bardzo szybko niewspółmiernie do prędkości poruszania się	Niepoprawny czas animacji	Zmodyfikować czas trwania animacji mając na uwadze optymalizację animacji	Dotyczy konkretnego obiektu	Należy dobrać odpowiedni parametr czasowy wykonywania aplikacji	tak	pozytywny	0,25

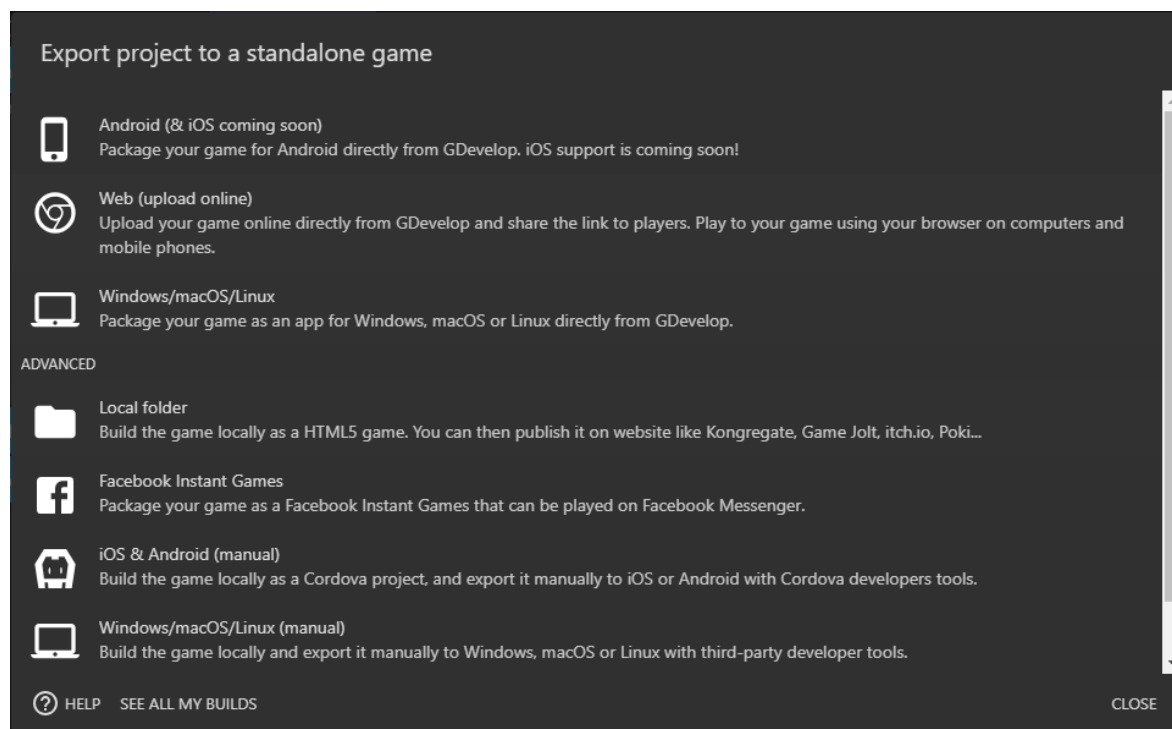
Źródło: Opracowanie własne



## 8.1. Opis procesu eksportu na platformy użytkowe

Narzędzie GDevelop umożliwia użytkownikowi na wyeksportowanie projektu na kilka różnych sposobów.

**Rysunek 39 Eksport projektu w programie GDevelop**



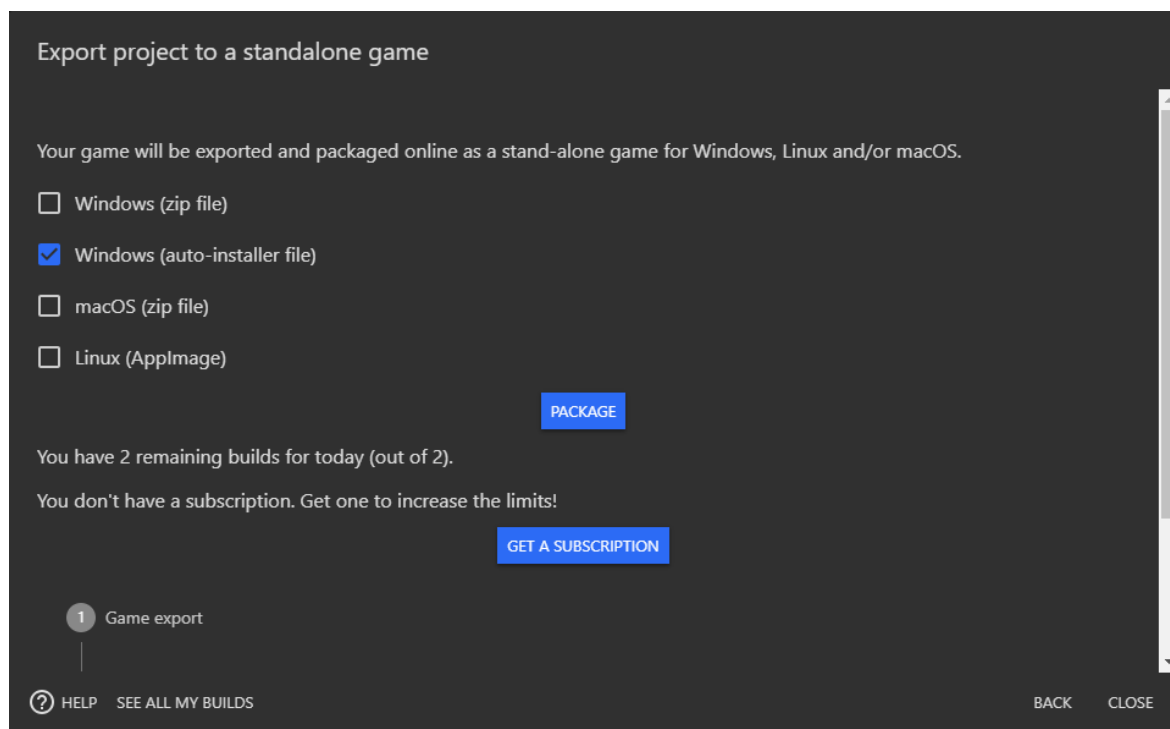
Źródło: Opracowanie własne

Otwierając z paska narzędzi zakładkę eksport ukazuje się widok (Rysunek 39). Z tego miejsca możliwe jest wyeksportowanie projektu na:

- plik .apk, który można uruchomić na urządzeniach z systemem Android,
- wysłać nasz projekt na serwer Web aplikacji GDevelop. Po kilku minutach w odpowiedzi otrzymujemy link do strony internetowej na której znajduje się grywalna wersja projektu,
- eksport gry na system Windows/macOS/Linux,
- zapis projektu do lokalnego folderu,
- eksport jako pliku Facebook Instant Games, który może zostać wykorzystany do uruchomienia gry w aplikacji firmy Facebook noszącej nazwę Messenger,
- utworzenie plików źródłowych na platformy iOS/Android/Windows/macOS/Linux, które w późniejszym czasie można skompilować za pomocą zewnętrznych programów.

Projekt zostanie zapisany przy pomocy opcji eksportu gry na system Windows/macOS/Linux. Po wybraniu wyżej opisanej opcji użytkownikowi ukazuje się następne okno wyboru (Rysunek 40).

#### Rysunek 40 Eksport projektu



Źródło: Opracowanie własne

W tym oknie program umożliwia eksport gry m.in. do systemu Windows, w którym można zapisać projekt w postaci skompresowanego pliku .zip lub pliku wykonywalnego .exe. Dla systemu macOS można wyeksportować grę podobnie jak do systemu Windows do pliku .zip, a dla systemu Linux jest możliwość zapisania projektu w postaci AppImage. Gra *Armin Legends* zostanie wyeksportowana do postaci pliku wykonywalnego .exe.

Wybierając opcję zaznaczoną na rysunku (Rysunek 40), a następnie naciskając przycisk *Package* program rozpoczyna proces eksportu gry. Proces ten składa się z trzech etapów:

- 1) Wysłania projektu na serwer Web aplikacji GDevelop.
- 2) Wyeksportowania przez serwer gry do pliku wykonywalnego.
- 3) Aplikacja udostępnia link do pobrania z sieci wyeksportowanego pliku .exe.

Po pobraniu pliku wykonywalnego i jego otwarciu, można rozpocząć rozgrywkę w grze *Armin Legends*.





## **SPIS RYSUNKÓW**

<b>Rysunek 1 Główne okno programu.....</b>	<b>16</b>
<b>Rysunek 2 Widok sceny.....</b>	<b>17</b>
<b>Rysunek 3 Okno widoku zdarzeń.....</b>	<b>19</b>
<b>Rysunek 4 Menadżer projektu GDevelop .....</b>	<b>20</b>
<b>Rysunek 5 Ilość gospodarstw domowych posiadających komputer osobisty w domu na całym świecie w latach 2005-2019 .....</b>	<b>22</b>
<b>Rysunek 6 Zrzut ekranu z gry Rayman Legends .....</b>	<b>23</b>
<b>Rysunek 7 Zrzut ekranu z gry Trine 4: The Nightmare Prince .....</b>	<b>24</b>
<b>Rysunek 8 Oficjalna strona GDevelop.....</b>	<b>27</b>
<b>Rysunek 9 Lista odnośników do witryn tematycznych znajdująca się na stronie głównej programu.....</b>	<b>28</b>
<b>Rysunek 10 GDevelop Wiki .....</b>	<b>29</b>
<b>Rysunek 11 Przykłady poradników w serwisie GDevelop Wiki.....</b>	<b>29</b>
<b>Rysunek 12 Forum GDevelop .....</b>	<b>30</b>
<b>Rysunek 13 GitHub GDevelop.....</b>	<b>31</b>
<b>Rysunek 14 Tablica Trello projektu GDevelop .....</b>	<b>32</b>
<b>Rysunek 15 Discord .....</b>	<b>33</b>
<b>Rysunek 16 Zmodyfikowana grafika .....</b>	<b>34</b>
<b>Rysunek 17 Grafika przedstawiająca główną postać gry .....</b>	<b>34</b>
<b>Rysunek 18 Przykładowa grafika przedstawiająca głównego przeciwnika .....</b>	<b>35</b>
<b>Rysunek 19 Ikony umieszczone w GUI.....</b>	<b>35</b>
<b>Rysunek 20 Menu w grze .....</b>	<b>37</b>
<b>Rysunek 21 Funkcje przycisków - Menu .....</b>	<b>37</b>
<b>Rysunek 22 Przedstawienie sceny Info .....</b>	<b>39</b>
<b>Rysunek 23 Funkcje przycisków - MenuInGame .....</b>	<b>40</b>
<b>Rysunek 24 Plansza YouWin .....</b>	<b>40</b>
<b>Rysunek 25 Przedstawienie przykładowej planszy grywalnej .....</b>	<b>41</b>
<b>Rysunek 26 Zdarzenia zewnętrzne.....</b>	<b>42</b>
<b>Rysunek 27 Ustawienia tła sceny oraz obsługa wczytania gry – Scena 3 .....</b>	<b>42</b>



<b>Rysunek 28 Wyświetlanie tytułu sceny - Plansza 3 .....</b>	<b>43</b>
<b>Rysunek 29 Mechanika walki z głównymi przeciwnikami - Scena 3 .....</b>	<b>43</b>
<b>Rysunek 30 Grupy obiektów .....</b>	<b>44</b>
<b>Rysunek 31 Zachowanie po naciśnięciu przycisku Escape.....</b>	<b>45</b>
<b>Rysunek 32 Mechanika ataku głównej postaci.....</b>	<b>46</b>
<b>Rysunek 33 PlayerBehavior - zdarzenia zewnętrzne .....</b>	<b>46</b>
<b>Rysunek 34 Testowanie działania konkretnej sceny przy pomocy narzędzia do symulacji .....</b>	<b>48</b>
<b>Rysunek 35 Test funkcjonalny konkretnej wersji gry na platformie Windows.....</b>	<b>49</b>
<b>Rysunek 36 Raport i Analiza znalezionych błędów .....</b>	<b>50</b>
<b>Rysunek 37 Procedura eliminowania błędów .....</b>	<b>51</b>
<b>Rysunek 38 Raport z rozwiązywania błędów .....</b>	<b>52</b>
<b>Rysunek 39 Eksport projektu w programie GDevelop.....</b>	<b>53</b>
<b>Rysunek 40 Eksport projektu .....</b>	<b>54</b>



## BIBLIOGRAFIA

1. Bomba Radosław, *Rola i oddziaływanie gier komputerowych na współczesną kulturę, społeczeństwo i rynek* [Online], Lublin, Uniwersytet Marii Curie-Skłodowskiej, 2009, [dostęp 25.01.2020]. Dostępny w Internecie: <https://rbomba.pl/archives/475>.
2. Kamiński Mateusz, *Tworzenie gier komputerowych. Kto tak naprawdę tworzy gry komputerowe?* [Online], 2019, [dostęp 25.01.2020]. Dostępny w Internecie: <http://www.sages.com.pl/blog/tworzenie-gier-komputerowych-kto-tak-na-prawde-tworzy-gry-komputerowe/>.
3. TheSpeed, *Czy wiesz, jak powstaje gra?* [Online], 2010, [dostęp 25.01.2020]. Dostępny w Internecie: [http://www.benchmark.pl/testy\\_i\\_recenzje/czy-wiesz-jak-powstaje-gra.html](http://www.benchmark.pl/testy_i_recenzje/czy-wiesz-jak-powstaje-gra.html).
4. Wilk Paweł, *System kontroli wersji Git, cz. 1 Podstawy i zasada działania*, [Online], 2019, [dostęp 25.01.2020]. Dostępny w Internecie: <https://randomseed.pl/pub/git-system-kontroli-wersji/01-wprowadzenie/>.
5. Dokumentacja programu GDevelop5 [Online]. Dostępny w Internecie: <http://wiki.compilgames.net/doku.php/gdevelop5/start>.
6. Dokumentacja programu GitHub [Online]. Dostępny w Internecie: <https://guides.github.com/>.
7. Dokumentacja programu GIMP [Online]. Dostępny w Internecie: <https://docs.gimp.org/2.4/pl/>.
8. Dokumentacja programu Trello [Online]. Dostępny w Internecie: <https://help.trello.com/>.
9. Przykładowe zastosowania GDevelop [Online]. Dostępny w Internecie: <https://gametemplates.itch.io/gdevelop-examples>.
10. Oficjalne forum programu GDevelop [Online]. Dostępny w Internecie: <https://forum.gdevelop-app.com/c/help-for-games-creation>.