



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

Michał Majda

Proceduralne generowanie map
z wykorzystaniem w grze typu roguelike

Praca dyplomowa inżynierska

Opiekun pracy:

dr hab. inż. Maciej Kusy, prof. PRz

Rzeszów, 2022

Spis treści

1. Charakterystyka porównawcza gier roguelike	7
1.1. Rogue	7
1.2. Caves of Qud	9
1.3. The Binding of Isaac	13
2. Szczegółowy opis gry	16
2.1. Główne założenia	16
3. Technologie i implementacja	17
3.0.1. Listingi programów	17
3.0.2. Numerowanie i punktowanie	18
3.1. Wydruk pracy	18
4. Podsumowanie i wnioski końcowe	20
Załączniki	21
Literatura	22

Wstęp

Roguelike to gatunek gier komputerowych posiadający elementy gier fabularnych RPG [1], bazujący na dużej losowości rozgrywki [2]. Losowość w tych grach W głównej mierze opiera się na losowym rozmieszczeniu przeciwników i przedmiotów oraz losowo generowanych mapach. Inna ważną cechą gier Roguelike jest permadeath – śmierć gracza oznacza konieczność gry od początku. Klasyczne roguelike-i z racji trudnej rozgrywki nie były popularne, ale w ostatnich latach coraz popularniejsze są gry inspirowane tym gatunkiem biorąc z niego tylko wybrane elementy. Z powodu standardowej dla roguelike-ów słabej jakościowo grafiki, oraz map tworzonych proceduralnie to znaczy według algorytmów zamiast ręcznie, gry te są stosunkowo proste w produkcji i nawet współcześnie możliwe do zrealizowania przez jedną lub dwie osoby. Nowsze gry posiadające elementy roguelike najczęściej rezygnują z standardowego dla starszych tytułów systemu turowego oraz ruchu po kwadratowej siatce na rzecz gry w czasie rzeczywistym i swobodnego ruchu. Czyni to nowych przedstawicieli gatunku roguelike bardziej przystępnymi co przekłada się na wzrost popularności [3].

Gatunek ten zapoczątkowany został przez grę Rogue w 1980 roku [4], od tej gry wzięła się też nazwa gatunku roguelike - Rogue podobne. Rogue gracz eksploruje podziemia walcząc z przeciwnikami oraz zdobywając coraz lepszy ekwipunek, celem ukończenia gry jest zdobycie amuletu Yendoru znajdującego się w najniższym poziomie. Grafika oparka jest o znaki tekstowe w kodowaniu ASCII [5], rozgrywka dzieje się w systemie turowym na kwadratowej siatce.

Mimo iż gatunek roguelike rozwinął się znacznie to wciąż powstają gry wierne klasycznym założeniom gatunku. Przykładem takiej gry jest Caves of Qud z 2015 roku [6]. Gra ta posiada wszystkie cechy klasycznego roguelike-a - turowa rozgrywka, permadeath oraz ruch po kwadratowej siatce, jednak klasyczną grafikę tekstową zastąpiono prostą grafiką typu pixel art [7].

Jednym z najbardziej popularnych przykładów nowoczesnych roguelike-ów jest gra The Binding of Isaac z roku 2011 [8]. Z gatunku roguelike gra ta zaczerpnęła losowe generowanie map, przedmiotów i przeciwników oraz permadeath. W The Binding of Isaac rozgrywka dzieje się w czasie rzeczywistym a ruch gracza nie jest ograniczony do siatki, dzięki temu gra jest bardziej zręcznościowa i łatwiejsza.

Głównym celem pracy jest stworzenie gry z gatunku roguelike z proceduralnie generowanymi mapami, oraz omówienie użytych metod generacji map.

Struktura pracy jest następująca: W rozdziale pierwszym dokonano szczegółowej charakterystyki porównawczej innych gier z gatunku roguelike: Rogue, Caves of Qud, The Binding of Isaac. Rozdział drugi opisuje grę roguelike stworzoną na potrzeby niniejszej pracy, w rozdziale trzecim zaprezentowano proces tworzenia tej gry oraz omówienie metod proceduralnego generowania map.

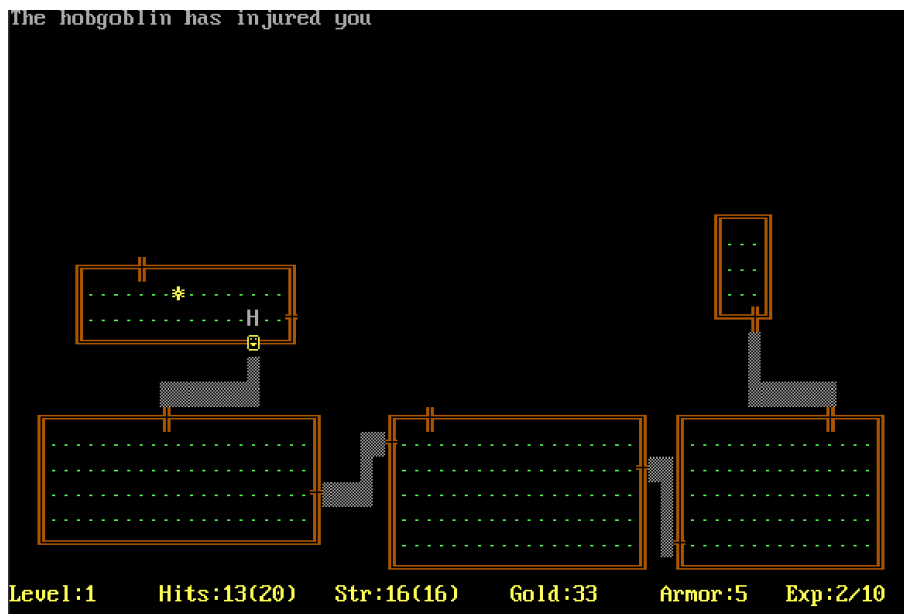
1. Charakterystyka porównawcza gier roguelike

W niniejszym rozdziale szczegółowo omówiono przykłowe gry z gatunku roguelike. Omówione gry to 1: Rogue, 2: Caves of Qud, 3: The Binding of Isaac. Gatunek roguelike od swoich początków nie jest mocno popularnym gatunkiem z powodu trudnej rozgrywki, skomplikowanego sterowania oraz prymitywnej oprawy graficznej. Powstały jednak popularne gry, które zaczerpują z gatunku tylko niektóre elementy, co w niektórych przypadkach oznacza płynną, prostszą rozgrywkę z elementami roguelike.

1.1. Rogue

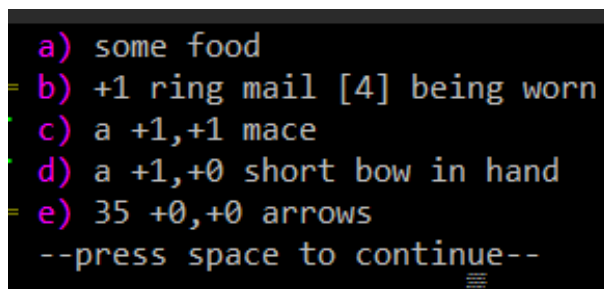
Rogue: Exploring the Dungeons of Doom to zaprogramowana w 1980 roku przez Michael-a Toy i Glenn-a Wichman w języku C gra, która zapoczątkowała gatunek Roguelike. Gracz wciela się w postać podróżującą w głąb podziemi osadzonych w fantastycznym, średniowiecznym świecie, w celu odnalezienia amuletu Yendoru. Rogue nie posiada wyboru ani konfiguracji postaci, dlatego każdy gracz rozpoczyna grę tą samą postacią. W trakcie rozgrywki napotkać można wielu przeciwników, a w walce z nimi pomagają znajdowane na poziomach przedmioty i ekwipunek. Podczas eksploracji głębszych poziomów podziemi gracz napotyka coraz trudniejszych przeciwników, lecz także przedmioty, które znajduje są coraz lepsze. Gra Rogue oparta jest o system turowy, przeciwnicy mogą wykonać ruch dopiero po ruchu gracza. W wyniku tego gra pozwala na dowolnie długie przemyślenie każdego ruchu i taktyczne podejście do walki. Każda mapa przedstawiona jest za pomocą siatki kwadratów, w wyniku czego ruch gracza ograniczony jest do 8 kierunków: góra, dół, lewo, prawo i ukosy.

Z racji ograniczeń sprzętowych w czasach wydania gry Rogue za reprezentację graficzną odpowiadają litery i znaki ASCII w terminalu. Na rysunku 1.1 przedstawiono fragment rozgrywki w grze Rogue. Pomarańczowymi liniami oznaczone są ściany pokoi, zielonymi kropkami podłogi, na szaro oznaczono korytarze. Żółta twarz reprezentuje postać gracza przeciwnicy są oznaczani literami jak hobogoblin oznaczony białą literą 'H' na rysunku.



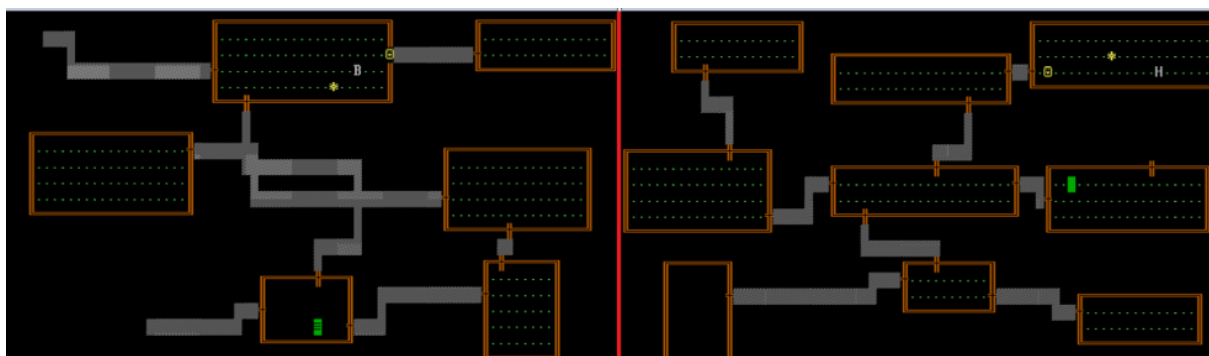
Rysunek 1.1: Widok główny gry Rogue

Do sterowania w Rogue używana jest tylko klawiatura, do poruszania przeznaczone są klawisze klawiatury numerycznej a do wykonywania akcji litery – na przykład klawisz 'T' służy do wyświetlenia ekwipunku. Aby zaatakować gracz musi poruszyć się w stronę przeciwnika będąc tuż przy nim. Wyświetlanie ekwipunku w grze Rogue pokazano na rysunku 1.2, manipulowanie przedmiotami w plecaku nie odbywa się w tym oknie, ale w głównym widoku gry za pomocą odpowiednich klawiszy i podania litery odpowiadającej danemu przedmiotowi w ekwipunku. Na przykład aby wyrzucić przedmiot 'a +1, +1 mace' znajdujący się w ekwipunku na rysunku 1.2 trzeba wcisnąć klawisz 'd' odpowiadający za wyrzucanie przedmiotów i literę 'c' przypisaną obecnie do tego przedmiotu, a w celu zjedzenia 'some food' klawisz 'e' odpowiadający za akcję jedzenia i literę 'a'.



Rysunek 1.2: Ekran ekwipunku w Rogue

Rogue zawiera proste proceduralnie generowane mapy, przykładowe dwie mapy pokazano na rysunku: 1.3. Każda mapa zawiera pokoje w układzie siatki 3x3, co szczególnie widać na mapie po prawej na rysunku 1.3, maksymalna liczba pokoi to dziewięć, ale może być ich mniej. Pokoje są połączone ze sobą korytarzami a niektóre przejścia mogą być ukryte przed graczem, co wymusza częste korzystanie z komendy przeszukiwania dostępnej pod klawiszem 'S'. Pokoje i korytarze mogą zawierać ukryte pułapki, które można wykryć używając komendy szukania. Jediną zmianą w głębszych poziomach są małe labirynty umieszczone czasami w miejsce niektórych pokoi.



Rysunek 1.3: Przykładowe mapy w grze Rogue

Z racji sporego wieku gry Rogue ciężko jest określić jej popularność, jednak z pewnością można stwierdzić, że jest to kluczowa dla gatunku roguelike gra, która stworzyła podwaliny jego głównych cech. Świadczy o tym między innymi sama nazwa gatunku roguelike, która oznacza "podobne do Rogue".

1.2. Caves of Qud

Caves of Qud to współczesny przykład rozwoju klasycznej formuły roguelike-ów. Gra tworzona jest w silniku Unity przez studio Freehold Games, wydana została w 2015 roku, lecz rozwijana jest do dzisiaj (stan na luty 2022 roku). Gra posiada wszystkie cechy klasycznego roguelike-a: działa w systemie turowym, posiada permadeath, mapa gry oparta jest o siatkę kwadratów, lecz dodaje też sporo nowoczesnych rozwiązań i znaczne rozwinięcie wielu mechanik gry. Caves of Qud osadzone jest w post apokaliptycznym fantastycznym świecie w którym zaawansowana technologia miesza się z fantastycznymi rasami oraz magią.

Jednym z głównych wyróżników Caves of Qud na tle innych gier tego typu

jest bardzo mocno rozwinięty system proceduralnie generowanych map. Gra posiada otwarty świat o statycznie umiejscowionych obszarach takich jak góry, dżungle i pustynie a także miastach, na rysunku 1.4 przedstawiono przykładowe mapy wygenerowane w tej grze. Pomimo statycznego ustawienia konkretnych obszarów w grze sam wygląd danych obszarów i miejsc jest proceduralnie generowany, a także na wielu mapach mogą zostać dodane obiekty takie jak ruiny, obozowiska i tym podobne. Główną i najbardziej interesującą metodą używaną w tej grze jest generowanie map oparte o Wave Function Collapse (Załamane funkcji falowej) [9]. Metoda ta pozwala na generowanie dowolnych map opartych o mały wzorec.



Rysunek 1.4: Przykładowe mapy w grze Caves of Qud. Po lewej wioska, po prawej dżungla z ruinami

Podobnie jak w grze Rogue tutaj również do poruszania się używana jest klawiatura numeryczne i klawisze liter do dostępu do ekwipunku, listy zadań i tym podobne, lecz Caves of Qud umożliwia też granie wyłącznie za pomocą myszki. Gra oferuje bardziej przejrzystą grafikę typu pixel art w formacie 16x24 piksele na jeden kwadrat na mapie, co przedstawiono na rysunku 1.5. Mylącym może być iż gra jest reprezentowana graficznie przez siatkę pionowych prostokątów, lecz w rzeczywistości gra traktuje je jako kwadraty, a ruch w każdym z ośmiu dostępnych kierunków zajmuje w czasie gry tyle samo.



Rysunek 1.5: Zrzut ekranu z gry Caves of Qud

Caves of Qud znacznie rozwinęło też interfejs użytkownika, co znacząco wpływa na ułatwienie rozgrywki. Interfejs jest tu przejrzysty i prosty, dzięki czemu zmiana ekwipunku oraz używanie przedmiotów jest dużo łatwiejsze. Na rysunku 1.6 przedstawiono ekran ekwipunku.



Rysunek 1.6: Ekran ekwipunku w Caves of Qud

Interesującym rozwiązaniem w Caves of Qud jest system handlowy. W grze istnieje system głodu i pragnienia, gracz więc w celu uniknięcia śmierci z głodu lub odwodnienia musi nosić ze sobą zapasy wody i pożywienia, które mają swoją wagę, więc można ich posiadać ograniczoną ilość. Walutą w tej grze jest woda pitna, która jest bardzo rzadko spotykana w postaci źródeł, zdecydowana większość wody w zbior-

nikach naturalnych jest niezdatna do picia. Z tego powodu gracz zbiera wodę nie tylko dla zaspokojenia pragnienia, ale również w celu handlu z spotykanymi między innymi w wioskach handlarzami. Postać gracza ma ograniczony udźwig, więc często handel sprowadza się do handlu wymiennego, ponieważ trudno jest nosić ze sobą duże kwoty w postaci wody.

Gra Caves of Qud znacznie rozwija aspekt RPG – odgrywania postaci. W trakcie wyboru postaci najpierw wybiera się typ postaci - mutant mający dostęp do wielu fizycznych lub mentalnych mutacji, które zapewniają dodatkowe zdolności, bonusy lub nawet dodatkowe kończyny lub True Kin będący niezmutowanym człowiekiem, który ma dostęp do cybernetycznych implantów dających różnorodne bonusy i umiejętności. Poza typem wybiera się też pochodzenie postaci, które określa początkowy ekwipunek oraz bonusy do atrybutów. W trakcie gry gracz może dowolnie rozwijać – zwiększając wartości atrybutów takich jak siła, zręczność, siła woli i tym podobne oraz przez uczenie się nowych umiejętności jak specjizacja w posługiwaniu się konkretnym typem broni lub lepszemu unikaniu ataków. Dzięki dostępowi do dużej ilości różnorodnych mutacji lub implantów, zależnie od typu postaci, gracz dodatkowo może kreować swoją postać na wiele sposobów.

W Caves of Qud proceduralne generowanie używane jest nie tylko do generowania map. W grze można znaleźć duże ilości proceduralnie tworzonych książek, z których część dotyczy także samej fabuły gry [10]. Oznacza to, że poza niektórymi stałymi we wszystkich rozgrywkach aspektami historii również część fabuły będzie inna w każdej nowej rozgrywce.

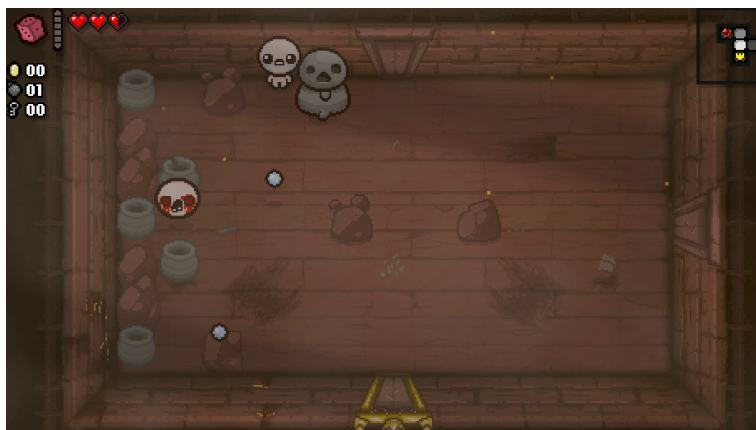
W odróżnieniu od większości gier tego typu Caves of Qud posiada mocno rozwiniętą historię i rozbudowane główne zadania fabularne. Poza główną linią zadań gra oferuje też sporo losowych zadań, które jednak są dosyć proste i najczęściej sprowadzają się do zdobycia konkretnego przedmiotu lub znalezienia jakiegoś miejsca. Główne zadania są różnorodne, często polegają na odwiedzaniu fabularnych, rozbudowanych lokacji, obronie pewnego miasta przed atakiem lub nawet rozwiązywaniu zagadek.

Biorąc pod uwagę fakt, że klasyczne gry typu roquelike posiadające dość wysoki poziom trudności rozgrywki, wciąż uznawane są za dość niszowy gatunek. Poszczególne gry zyskują jednak relatywnie dużą popularność, czego przykładem jest Caves of Qud, które w serwisie Steam posiada 4356 z czego 95% jest pozytywnych (stan na luty 2022) [11].

1.3. The Binding of Isaac

Gra The Binding of Isaac została stworzona w roku 2011 przez Edmund-a McMillen i Florian-a Himsl, a następnie jako remake The Binding of Isaac: Rebirth w roku 2014. W niniejszej pracy skupiono się na omówieniu The Binding of Isaac: Rebirth, gdyż jest to nowsza i bardziej popularna odsłona. Gra ta jest przykładem wzorowania się na roguelike-ach, ale odejściu od części klasycznych założeń. Z gatunku zaczerpnięto losowo generowane mapy, losowo rozmieszczane przedmioty i przeciwników, oraz konieczność grania od początku w przypadku śmierci głównej postaci. Rozgrywka w The Binding of Isaac dzieje się w czasie rzeczywistym, a ruch postaci nie jest ograniczony do siatki kwadratów. Z tego powodu gra jest bardzo dynamiczna, a walka zręcznościowa.

The Binding of Isaac odeszło od podziału mapy na kwadratową siatkę, ale wciąż korzysta z względnie prostej grafiki typu pixel art, co przedstawiono na rysunku: 1.7.



Rysunek 1.7: Przykład rozgrywki w The Binding of Isaac: Rebirth

Gracz wciela się początkowo w postać Isaac, ale w trakcie rozgrywki może odblokować wiele nowych postaci, które różnią się początkowymi statystykami takimi jak zdrowie i siła ataku oraz posiadanyimi przedmiotami. Gra nie posiada typowego ekwipunku takiego jak zbroje i broń, zamiast tego w grze zbiera się nieograniczoną ilość przeróżnych przedmiotów, które zapewniają pasywne bonusy. Bonusy te wzmacniają postać na wiele różnych sposobów, od podstawowych zwiększających zdrowie lub atak po takie, które zmieniają łyzy, którymi strzela gracz na zupełnie inną postać. Jednym z głównych aututów The Binding of Isaac są możliwe kombinacje przedmiotów, przykładowo jeśli gracz zdobędzie przedmiot, dzięki któremu pociski postaci będą się same

nakierowywały na przeciwników, a następnie przedmiot, który zamienia pociski na laser, to ten laser będzie zaginany tak, aby zawsze trafiać przeciwników bez konieczności celowania, co pokazano na rysunku 1.8.



Rysunek 1.8: Przykład kombinacji przedmiotów w The Binding of Isaac: Rebirth

W grze The Binding of Isaac za sterowanie postacią odpowiadają klawisze 'W', 'A', 'S', 'D', a za strzelanie klawisze strzałek, do używania przedmiotów jest też używana spacja i klawisz 'Q'. Proste sterowanie jest kolejnym powodem przystępności tej gry. W porównaniu do innych gier roguelike losowo generowane mapy w The Binding of Isaac są znacznie prostsze. Gra składa się z wielu poziomów podzielonych na pokoje, każdy z poziomów zakończony jest walką z silnym przeciwnikiem – bossem. Pokoje są wybierane losowo z puli pokoi przygotowanych przez twórców, a ich rozkład i połączenia są losowo generowane. Na rysunku 1.9 przedstawiono przykładowy układ poziomów. Z każdym kolejnym poziomem ilość pokoi jest coraz większa, ale jakość napotykanego przedmiotów nie wzrasta. Napotymane przedmioty są kompletnie losowe, dlatego czasem już nawet po pierwszym poziomie postać gracza może posiadać kilka najsilniejszych przedmiotów.



Rysunek 1.9: Przykład układ poziomów w The Binding of Isaac: Rebirth

Jedną z cech gier gatunku roguelike, które implementuje The Binding of Isaac jest permadeath. W porównaniu jednak do innych do tytułów, na których ukończenie trzeba często poświęcić parę godzin, The Binding of Isaac sprawnemu graczowi zajmie około godzinę na ukończenie, w wyniku czego rozpoczynanie rozgrywki od początku nie jest tak dotkliwe, i zachęca do częstego powtórnego przechodzenia gry.

The Binding of Isaac: Rebirth jest jedną z najbardziej popularnych gier z gatunku roguelike, na serwisie Steam posiada 166244 opini, w tym 97% pozytywnych (stan na luty 2022) [12]. Dowodzi to, że przeciętny gracz woli dynamiczną, rozgrywaną w czasie rzeczywistym rozgrywkę od wolnej, taktycznej rozgrywki turowej.

2. Szczegółowy opis gry

W niniejszym rozdziale przedstawiono wszystkie elementy i możliwości gry z gatunku roguelike stworzonej na potrzeby niniejszej pracy.

2.1. Główne założenia

3. Technologie i implementacja

- a) założenia – dane,
- b) opis zastosowanej metody rozwiązania lub analizy,
- c) opis proponowanego rozwiązania, wyniki analizy teoretycznej, obliczenia, projekt konstrukcyjny, procesowy, technologiczny,
- d) wyniki badań analitycznych, symulacyjnych lub eksperymentalnych itp.

3.0.1. Listingi programów

W pracy dyplomowej możesz umieszczać fragmenty programów. Pamiętaj, aby umieszczać krótkie, tylko najważniejsze fragmenty kodów źródłowych. Zawsze je komentuj w treści pracy dyplomowej. Typowo w L^AT_EX kody źródłowe umieszczane są w środowisku verbatim (`\begin{verbatim}...\end{verbatim}`). Obecnie istnieje jednak bardziej nowoczesne i bardziej funkcjonalne środowisko `lstlisting` (wymaga zainstalowanego w systemie pakietu `listings`). Zwróć uwagę, że możesz kolorować składnię automatycznie za pomocą parametru `language`. W niniejszym dokumencie przedstawiono dwa przykłady listingów, Listing 1 to przykład kodu źródłowego Matlab, a poniżej Listing 2 dla Perl'a.

```
1 pub fn new(x: usize, y: usize, width: usize, height: usize) ->
  GuiMapGenTestingManager {
2   GuiMapGenTestingManager {
3     x,
4     y,
5     width,
6     height,
7     selected: 0,
8     bg: rltk::RGB::named(rltk::BLACK),
9     title: TextCol::new(vec!["Map generators testing"].to_string
10    (), rltk::RGB::named(rltk::WHITE))),
11     options: vec![],
12     options_sprites_indexes: vec![],
13     show_steps: false,
14     map_gen: Box::new(TestMap::new(width - 4, height - 4)),
15     current_history_index: 0,
16  }
```

Listing 1: Listing programu Matlab

```
1 my $url = 'http://pei.prz.edu.pl';
2 use LWP::Simple;
```

```

3 my $content = get $url;
4 die "Couldn't get $url" unless defined $content;
5 print $content;
6 print "\n";
7 print "Length " + length($content)

```

Listing 2: Listing programu Perl

Z pewnością przeglądając źródło tego dokumentu zobaczysz, że kody źródłowe powinny mieć zdefiniowane parametry `label`, aby łatwo w tekście do nich się odwoływać. Numeracja linii jest w stylu domyślnie włączona (to przydatne, bo w treści pracy łatwo odwołać się dzięki temu do konkretnego wiersza w kodzie źródłowym), możesz je wyłączyć podając jako parametr `numbers=none`. Więcej szczegółów możesz odnaleźć w sekcji `\lstset` pliku arkusza stylu.

3.0.2. Numerowanie i punktowanie

- 1) Pierwszy poziom (stosuje się numerowanie lub punktowanie). Formatowanie: akapit wyjustowany, wcięcie od lewej 0,75 cm, wysunięcie co 0,5 cm.
- 2) Znakiem numerowania jest liczba (z kropką lub nawiasem).
 - drugi poziom (stosuje się wyłącznie punktowanie). Formatowanie: akapit wyjustowany, wcięcie od lewej 1,25 cm, wysunięcie co 0,5 cm,
 - znakiem punktowania jest łącznik lub mała litera alfabetu (z nawiasem). Nie zaleca się stosowania kropek, strzałek itp.,
 - punktowane akapity rozpoczyna się minuskulą (małą literą), na końcu akapitu stawia się przecinek, ostatni punktowany akapit kończy się kropką.
- 3) Numerowane akapity rozpoczyna się majuskulą (wielką literą) i kończy kropką.
- 4) Należy zwrócić uwagę, aby nie rozdzielać numerowania/punktowania pomiędzy kolejnymi stronami tekstu.

3.1. Wydruk pracy

Przed wydrukiem należy usunąć ewentualne błędy literowe i sprawdzić prawidłową interpunkcję. Przykładowo, łącznik zapisuje się za pomocą krótkiego minusa (np. badawczo-rozwojowy) natomiast myślnik – stosowany w zdaniach wtrąconych – zapisuje się za pomocą długiej pauzy. Dzielenie wyrazów według uznania Autora

(można podzielić długie wyrazy, powodujące duże „rozstrzelenie” tekstu w poprzedzającym wierszu. Zaleca się usunięcie pojedynczych znaków na końcu wiersza oraz podwójnych spacji w tekście. Dla przedrostka „mikro” należy unikać stosowania litery „u” zamiast „μ”. Znak „μ” można otrzymać przytrzymując lewy Alt i wpisując na klawiaturze numerycznej 0181 (podobnie „stopień”: Alt-0176). W celu uniknięcia „rozstrzelenia” liczb i ich jednostek zaleca się używanie „twardej” spacji pomiędzy liczbą i jednostką. Należy sprawdzić, czy tytuły podrozdziałów/zakresów nie zostały jako pojedyncze wiersze na poprzedniej stronie oraz czy rysunki/tabele i ich tytuły nie zostały rozdzielone pomiędzy kolejnymi stronami.

Pracę drukuje się dwustronnie. Zaleca się wydruk w kolorze. Przed wydrukiem należy ponumerować strony (czcionka 10 pkt., dół strony, akapit wyśrodkowany). Strony tytułowej oraz strony z podziękowaniem nie numeruje się. Spis treści rozpoczyna się od strony numer 3 (lub 5, jeżeli zamieszczono podziękowania).

4. Podsumowanie i wnioski końcowe

Załączniki

Według potrzeb zawarte i uporządkowane uzupełnienie pracy o dowolny materiał źródłowy (wydruk programu komputerowego, dokumentacja konstrukcyjno-technologiczna, konstrukcja modelu – makiety – urządzenia, instrukcja obsługi urządzenia lub stanowiska laboratoryjnego, zestawienie wyników pomiarów i obliczeń, informacyjne materiały katalogowe itp.).

Literatura

- [1] Felipe Pepe, The CRPG Book: A Guide to Computer Role-Playing Games, Bitmap Books, 2020.
- [2] Harris J.: Exploring Roguelike Games. CRC Press, 2020.
- [3] Daniel Doan, Why Are Modern Roguelike Games So Popular?, <https://gamedevlibrary.com/gamedev-thoughts-why-are-modern-roguelike-games-so-popular-53c1f5a05485>.
- [4] Michael Toy, Glenn Wichman, Ken Arnold, Rogue, 1980, <https://www.mobygames.com/game/rogue>
- [5] American Standards Association, American Standard Code for Information Interchange ASCII, 1963
- [6] Freehold Games, Caves of Qud, 2015, <https://www.cavesofqud.com>
- [7] Angie Kordic, What Exactly is Pixel Art and How Did It Come Back to Life ?, <https://www.widewalls.ch/magazine/pixel-art>
- [8] Edmund McMillen, Nicalis, The Binding of Isaac: Rebirth, 2014, <https://bindingofisaac.com>
- [9] Brian Bucklew (Freehold Games), Math for Game Developers: Tile-Based Map Generation using Wave Function Collapse in 'Caves of Qud', Game Developers Conference, <https://www.gdcvault.com/play/1026263/Math-for-Game-Developers-Tile> .
- [10] Jason Grinblat (Freehold Games), Procedurally Generating History in 'Caves of Qud', Game Developers Conference, <https://www.gdcvault.com/play/1024990/Procedurally-Generating-History-in-Caves> .
- [11] https://store.steampowered.com/app/333640/Caves_of_Qud/
- [12] https://store.steampowered.com/app/250900/The_Binding_of_Isaac_Rebirth/

STRESZCZENIE PRACY DYPLOMOWEJ INŻYNIERSKIEJ

**PROCEDURALNE GENEROWANIE MAP
Z WYKORZYSTANIEM W GRZE TYPU ROGUELIKE**

Autor: Michał Majda, nr albumu: EF-157100

Opiekun: dr hab. inż. Maciej Kusy, prof. PRz

Słowa kluczowe: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po polsku

BSC THESIS ABSTRACT

TEMAT PRACY PO ANGIELSKU

Author: Michał Majda, nr albumu: EF-157100

Supervisor: (academic degree) Imię i nazwisko opiekuna

Key words: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po angielsku