

**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
2.2. Opis menu głównego gry	17
2.3. Sterowanie w grze	17
2.4. Opis poziomów	19
2.4.1. Poziom 1	20
2.4.2. Poziom 2	20
2.4.3. Poziom 3	21
2.4.4. Poziom 4	22
2.4.5. Poziom 5	23
2.4.6. Poziom 6	23
2.5. Okno gry	23
2.6. Przedmioty oraz ich używanie	25
2.6.1. Okna ekwipunku	25
2.6.2. Zbroje i bronie	27
2.6.3. Mikstury	27
2.6.4. Zwoje	27
2.7. Tester proceduralnie generowanych poziomów	29
3. Technologie i implementacja	32
3.1. Technologie	32
3.2. Implementacja	33
3.2.1. Struktura projektu	33
3.2.2. Architektura Entity Component System	35
4. Podsumowanie i wnioski końcowe	38
Załączniki	39
Literatura	40

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 inspirowające się tym gatunkiem biorąc z niego tylko wybrane elementy. Z powodu standar-dowej 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 przystępymi 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 graczu 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 rozwinał 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 proceduralnej 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 przypadku oznacza płynną, prostszą rozgrywkę z elementami roguelike.

1.1. Rogue

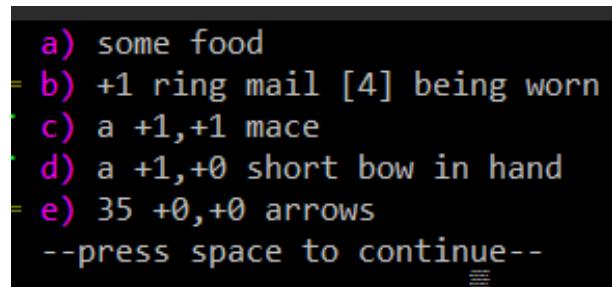
Rogue: Exploring the Dungeons of Doom to zaprogramowana w 1980 roku przez Michael-a Toy i Glenn-a Wichman w języku C gra, która zapoczątkowała gatunek Roguelike. Gracz wciela się w postać podróżującą w głąb podziemi osadzonych w fantastycznym, średniowiecznym świecie, w celu odnalezienia amuletu Yendoru. 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 bardziej trudnych 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 pokojów, zielonymi kropkami podłogi, na szaro oznaczono korytarze. Żółta twarz reprezentuje postać gracza przeciwnicy są oznaczani literami jak hobogoblin oznaczony białą literą 'H' na rysunku.



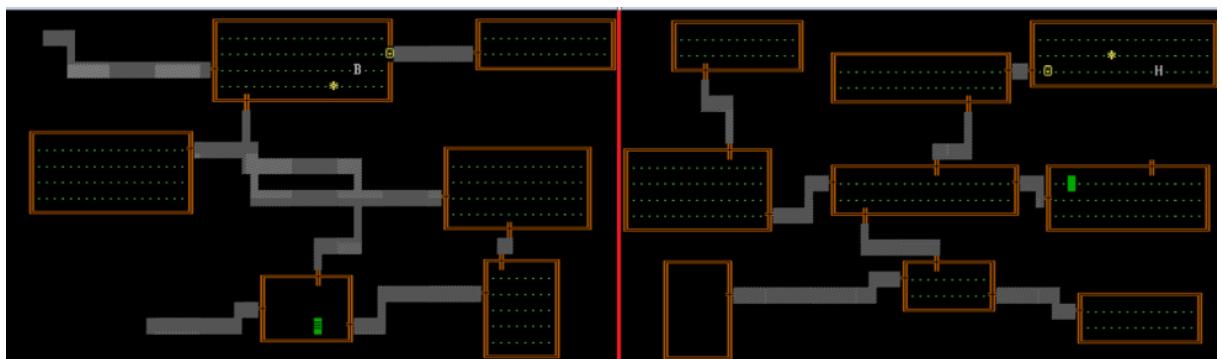
Rysunek 1.1: Widok główny gry Rogue

Do sterowania w Rogue używana jest tylko klawiatura, do poruszania przeznaczone są klawisze klawiatury numerycznej a do wykonywania akcji litery – na przykład klawisz 'I' 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 pokojów 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. Jedyną zmianą w głębszych poziomach są małe labirynty umiejsczone czasami w miejsce niektórych pokojów.



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 (Załamanie funkcji falowej) [9]. Metoda ta pozwala na generowanie dowolnych map opartych o mały wzorzec.



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 spotykanyimi 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 specjalizacja w posługiwaniu się konkretnym typem broni lub lepszego unikania 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 ksiązek, 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 rozwijywania zagadek.

Biorąc pod uwagę fakt, że klasyczne gry typu roquellike posiadające dość wysoki poziom trudności rozgrywki, wciąż uznawane są za dość niszowy gatunek. Poszczególne gry zyskują jednak relatywnie dużą popularność, czego przykładem jest Caves 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 Himsł, 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 założen. 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 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 od blokować wiele nowych postaci, które różnią się początkowymi statystykami takimi jak zdrowie i siła ataku oraz posiadanymi przedmiotami. Gra nie posiada typowego ekwi-punktu takiego jak zbroje i bronie, 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 łzy, którymi strzela gracz na zupełnie inną postać. Jednym z głównym 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 zginany 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 pokojów przygotowanych przez twórców, a ich rozkład i połączenia są losowo generowane. Na rysunku 1.9 przedstawiono przykładowy układ poziomów. Z każdym kolejnym poziomem ilość pokojów jest coraz większa, ale jakość napotykanych przedmiotów nie wzrasta. Napotykane przedmioty są kompletnie losowe, dlatego czasem już nawet po pierwszym poziomie postać gracza może posiadać kilka najsilniejszych przedmiotów.



Rysunek 1.9: Przykładowe układy 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 godzi, The Binding of Isaac sprawnemu graczowi zajmie około godzinę na ukończenie, w wyniku czego rozpoczęwanie 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 opinii, 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. Kolejnym dowodem popularności tej gry jest wydana w 2018 roku gra planszowa The Binding of Isaac: Four Souls, która w tydzień zebrała ponad milion dolarów na crowdfundingowym serwisie Kickstarter [13].

2. Szczegółowy opis gry

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

2.1. Główne założenia

Gra stworzona na potrzeby nieniejszej pracy spełnia podstawowe cechy klasycznych gier z gatunku roguelike:

- oparta jest o system turowy,
- zawiera plansze generowane proceduralnie,
- rozmieszczenie przeciwników i przedmiotów jest losowe,
- mapy są w postaci siatki kwadratów, ruch postaci ograniczony jest do poruszania się po nich,
- postać gracza posiada ekwipunek, który może założyć – zbroje i bronie,
- gra posiada przedmioty, które można użyć – mikstury i magiczne zwoje,
- permadeath - śmierć kończy rozgrywkę, brak możliwości zapisu i wczytania stanu rozgrywki.

Projekt gry posiada też kilka bardziej nowoczesnych aspektów:

- grafika typu pixel art,
- rozwinięty, prostszy w użytkowaniu interfejs,
- system prezentowania z krokami proceduralnego generowania map.

Gra składa się z sześciu różnorodnych, proceduralnie generowanych poziomów, gracz napotyka coraz trudniejszych przeciwników w kolejnych poziomach, a także znajduje coraz lepszy ekwipunek i przedmioty. Celem gry jest pokonanie ostatniego przeciwnika "Mighty Blop", znajdującego się w ostatnim poziomie. Często takiego ostatniego, najsilniejszego przeciwnika określa się w grach jako "Boss". Aby przejść grę gracz powinien eksplorować kolejne poziomy w celu odnalezienia coraz silniejszych przedmiotów i ekwipunku, które niezbędne są przy walce z trudniejszymi przeciwnikami. Gra

osadzona jest w fantastycznym, średniowiecznym świecie, w którym spotkać można rycerzy, goblinów i orków, a także magiczne mikstury i zwoje.

2.2. Opis menu głównego gry

Po uruchomieniu gry pierwsze co zobaczy gracz to główne menu przedstawione na rysunku 2.10. Menu składa się z żółtego tytułu gry, oraz czterech opcji. Poruszać się po menu można za pomocą klawiszy strzałek, aktualnie wybrana opcja jest zaznaczona na zielono, a wybór opcji przeprowadzany jest klawiszem enter.



Rysunek 2.10: Menu główne gry

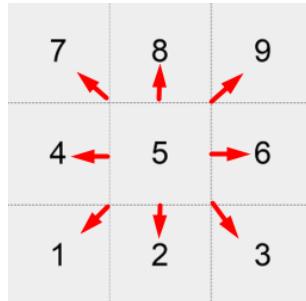
Menu składa się z następujących opcji:

- 'New Game' – rozpoczęcie nowej gry,
- 'Test Map Generators' – otwiera menu testera proceduralnego generowania map,
- 'Controls' – otwiera listę komend dostępnych w grze,
- 'Quit' – zamknięcie programu.

2.3. Sterowanie w grze

Główną metodą poruszania się postacią jest klawiatura numeryczna, co używane jest od dawna w gatunku roguelike. Jest to także najwygodniejsza kombinacja klawiszy przy możliwości ruchu w ośmiu kierunkach – dół, góra, prawo, lewo oraz skosy, schemat poruszania się przy użyciu klawiatury numerycznej przedstawiono na rysunku 2.11. Nie wszystkie klawiatury, a szczególnie te w laptopach, posiadają klawiaturę numeryczną, dlatego gra posiada też alternatywne sterowanie oparte o klawisze strzałek i klawisze

liter. Klaiwsz '5' w przypadku klawiatury numerycznej używany jest do pomijania tury, co jest istotną możliwością w grze turowej. Jednokrotne naciśnięcie klaiwsza odpowiadającego za ruch w danym kierunku porusza postać gracza o jedno pole w danym kierunku, jeżeli nie jest ono zajmowane przez ścianę. Jeśli pole na które porusza się gracz zajmowane jest przez przeciwnika, to zamiast ruchu postać gracza wykona atak przeciwko temu przeciwnikowi, jest to standardowa metoda ataku w tego typu grach.



Rysunek 2.11: Schemat sterowania na klawiaturze numerycznej

Objaśnienia sterowania i lista dostępnych akcji w grze z menu głównego za pomocą opcji 'Controls' na rysunku 2.12 przedstawiono widok dostępny po wybraniu tej opcji.



Rysunek 2.12: Objaśnienia sterowania w grze

Poniżej wyjaśnienie wszystkich komend dostępnych w grze w kolejności z ry-

sunku 2.12, Oznaczenie NumpadX oznacza klawisz cyfry X z klawiatury numerycznej.

Oznaczenie 'X' oznacza klawisz litery lub znaku X:

- 1) Numpad8 / klawisz strzałki w góre – poruszenie się w góre,
- 2) Numpad2 / klawisz strzałki w dół – poruszenie się w dół,
- 3) Numpad4 / klawisz strzałki w lewo – poruszenie się w lewo,
- 4) Numpad6 / klawisz strzałki w prawo – poruszenie się w prawo,
- 5) Numpad7 / 'O' – poruszenie się na ukos – góra, lewo,
- 6) Numpad9 / 'P' – poruszenie się na ukos – góra, prawo,
- 7) Numpad1 / 'K' – poruszenie się na ukos – dół, lewo,
- 8) Numpad3 / 'L' – poruszenie się na ukos – dół, prawo,
- 9) Numpad5 / 'W' – pominięcie jednej tury,
- 10) ',' (klawisz przecinka) – zejście w dół po schodach,
- 11) '.' (klawisz kropki) – wejście w góre po schodach,
- 12) 'G' – podniesienie przedmiotu,
- 13) 'I' – otwarcie menu ekwipunku,
- 14) 'E' – pokazuje obecnie wyekwipowane przedmioty,
- 15) klawisz Enter – wybór opcji w menu,
- 16) klawisz Esc – powrót do menu głównego, kończy to obecną rozgrywkę,
- 17) klawiszem Enter powrócić można z menu 'Controls' do menu głównego.

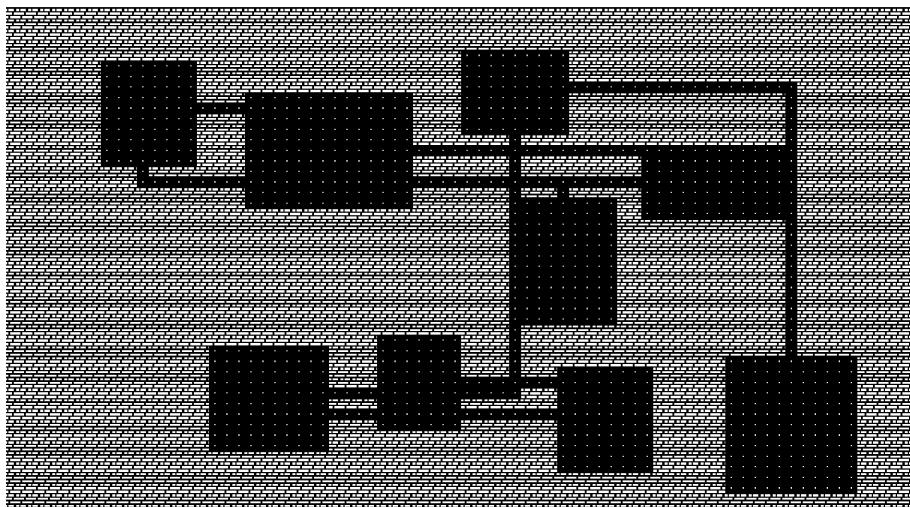
2.4. Opis poziomów

Gra składa się z sześciu poziomów oferujących różnych przeciwników i przedmioty. W tej sekcji w celu prezentacji wyglądu poszczególnych poziomów posłużono się testerem proceduralnie generowanych map. Poziomy zaprezentowano w kolejności w jakiej występują one w trakcie rozgrywki. Mapy na poziomach 1-5 są proceduralnie

generowane, więc przy każdej nowej rozgrywce ich wygląd będzie podobny, ale odmienny. Również rozmieszczenie przeciwników i przedmiotów na poziomach jest za każdym razem losowe. Akcja gry ma miejsce w podziemnych lochach połączonych z jaskiniami, kolejne poziomy są coraz głębiej w ziemię, dlatego są połączone ze sobą schodami. Gracz ma możliwość swobodnego wracania do poprzednich poziomów.

2.4.1. Poziom 1

Poziom pierwszy to kilka prostych pokojów połączonych ze sobą korytarzami, przykładowy układ tej mapy przedstawiono na rysunku 2.13. Na poziomie tym napotkać można najprostszych przeciwników: 'blip' oraz 'goblin'. Istnieje też mała szansa, że w pokoju z golbinami pojawi się też bardziej trudny przeciwnik – 'orc'. Na poziomie tym znaleźć można podstawowy pancerz i miecz, a także mikstury uzdrawiające i zwoje uśpienia, które są szczególnie przydatne w walkach z bardziej trudnymi przeciwnikami w późniejszych poziomach.

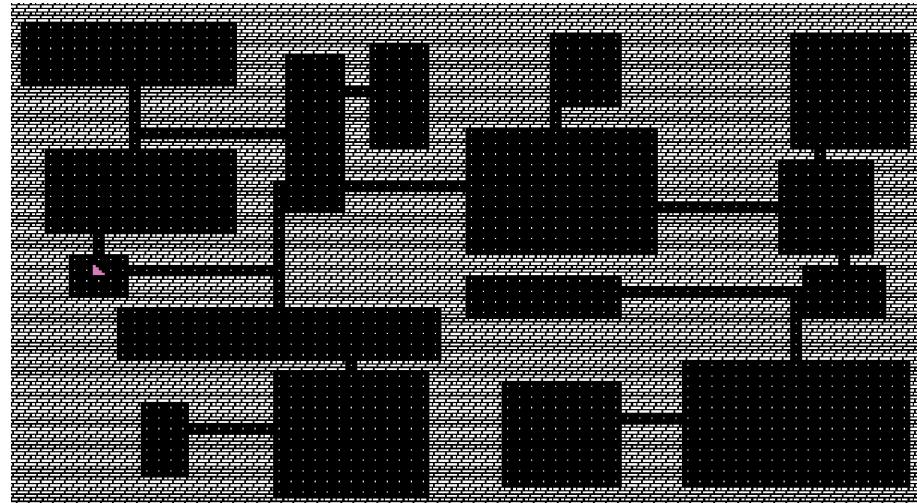


Rysunek 2.13: Przykładowy układ poziomu pierwszego

2.4.2. Poziom 2

Poziom drugi to bardziej rozwinięta wersja poziomu pierwszego z większą liczbą pokojów i inną metodą generacji. Przykładowy poziom drugi przedstawiono na rysunku 2.14. Na drugim poziomie spotkać można gołbliny z poprzedniego poziomu a także bardziej trudnych przeciwników 'orc' – ork oraz 'knight' – rycerz. Szczególnie trudnym przeciwnikiem na tym poziomie jest rycerz, który wyekwipowany jest w mocną zbroję i miecz, które można zdobyć po jego pokonaniu. W pokoju z rycerzem jest też

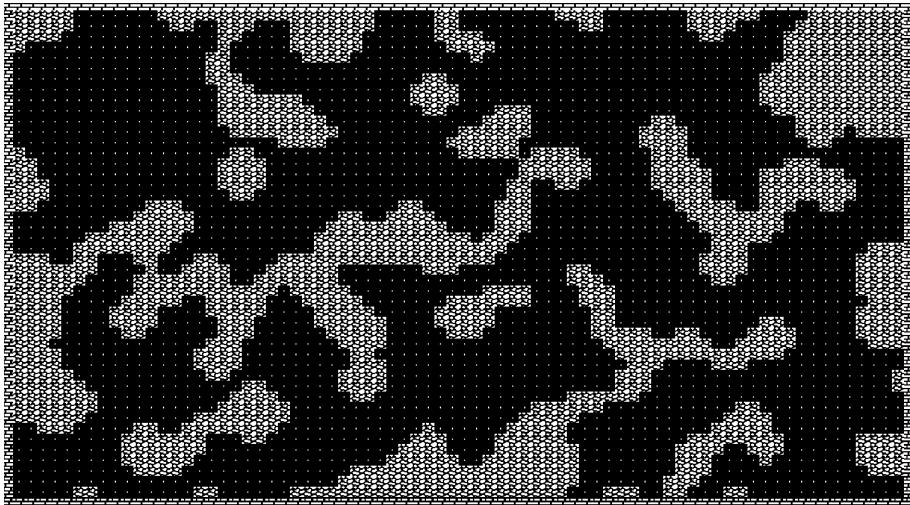
szansa na znalezienie poteżniejszych magicznych zwojów - zwoju kuli ognia, oraz zwoju teleportacji. Na poziomie tym można też znaleźć małe ilości mocniejszych mikstur uzdrowienia, oraz zwoje magicznych pocisków.



Rysunek 2.14: Przykładowy układ poziomu drugiego

2.4.3. Poziom 3

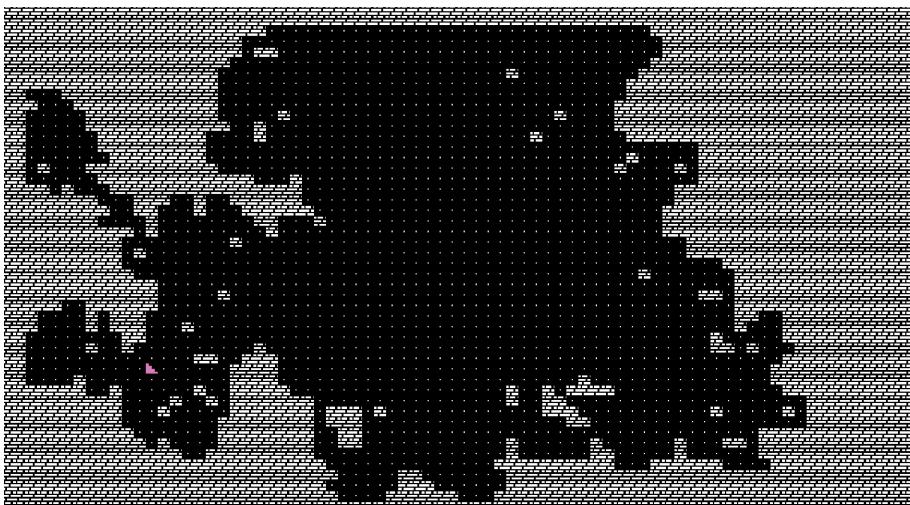
Poziom trzeci to duża jaskinia wypełniona masą goblinów i orków. Przykładowy poziom trzeci przedstawiono na rysunku 2.15. Mimo, że nie spotyka tu się trudniejszych przeciwników niż orkowie, to trudnością tego etapu jest duża liczba przeciwników, oraz duży otwarty teren. Taki układ mapy w przeciwieństwie do ciasnych korytarzy poprzednich poziomów może łatwo doprowadzić do otoczenia gracza przez wielu przeciwników.



Rysunek 2.15: Przykładowy układ poziomu trzeciego

2.4.4. Poziom 4

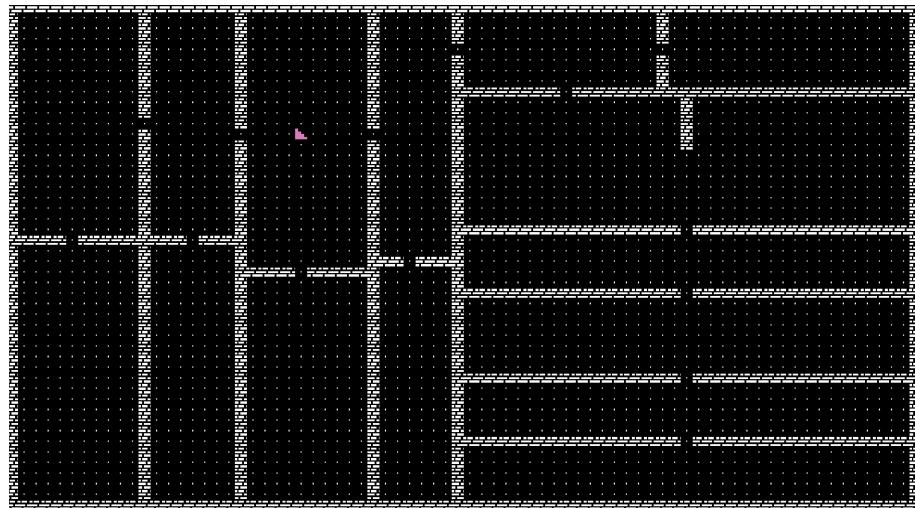
Poziom czwarty to kolejna jaskinia, tym razem z dużo większym obszarem w centrum, oraz okazjonalnymi ciasnymi odnogami. Przykładowy poziom czwarty przedstawiono na rysunku 2.16. Na poziomie tym można spotkać duże ilości prostych 'blipów' z pierwszego poziomu, a także sporo ich większych odmian – 'blop'-ów. 'Blop' poza byciem silniejszą odmianą blipa posiada też specjalną zdolność – po śmierci rozpadnie się na kilka blipów. Na tym poziomie można też znaleźć mocniejszą zbroję i miecz, a także duże mikstury uzdrowienia i zwoje obszarowego uśpienia.



Rysunek 2.16: Przykładowy układ poziomu czwartego

2.4.5. Poziom 5

Poziom piąty to duży obszar podzielony na pokoje bezpośrednio połączone ze sobą, bez korytarzy. Przykładowy poziom piąty przedstawiono na rysunku 2.17. Na tym poziomie spotkać można najmocniejszych przeciwników: rycerzy ('knight') oraz łotrzyków ('rogue'). W trakcie eksploracji poziomu piątego odnaleźć można namocniejszą w grze zbroję oraz miecz, co będzie dla gracza szczególnie przydatne na poziomie szóstym.



Rysunek 2.17: Przykładowy układ poziomu piątego

2.4.6. Poziom 6

Poziom szósty to ostatni poziom, cały będący otwartą przestrzenią. W tym poziomie gracz spotka tylko jednego przeciwnika – 'Mighty Blop' (potężny blop), którego pokonanie jest celem gry. Potężny blop jest silniejszą wariacją blopa, na które rozpada się po śmierci, one z kolei po śmierci rozpadną się na blipy. Aby ukończyć grę należy go pokonać oraz zejść schodami w dół.

2.5. Okno gry

Po rozpoczęciu nowej gry gracz zobaczy okno z wycinkiem mapy wycentrowanym na postać gracza, a także okno dziennika zdarzeń i pasek punktów życia, co pokazano na rysunku 2.18. Widok mapy jest powiększonym wycinkiem poziomu, przesuwającym się razem z graczem. Na mapie poza ścianami zaznaczone są też pola podłogi za pomocą białej kropki. Jest to przydatne w grach opierających się o ruch na siatce

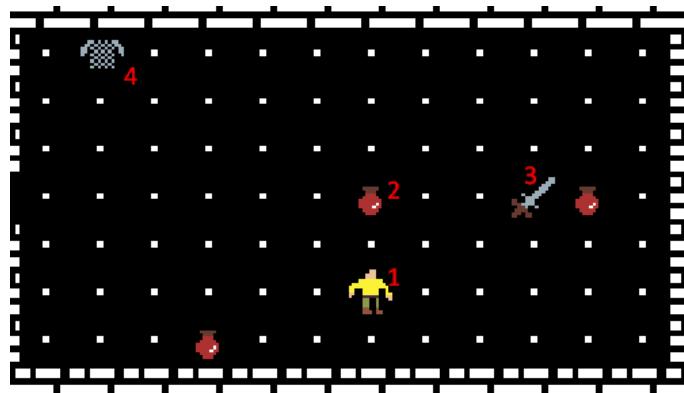
kwadratów do przeliczania odległości. W dzienniku zdarzeń zapisywane są wydarzenia z gry takie jak podnoszenie przedmiotu i ataki gracza lub przeciwników. Pasek życia posiada liczbową prezentację punktów życia w postaci [aktualne punkty zdrowia / maksymalne punkty zdrowia], oraz graficzną reprezentację – czerwony pasek zmniejszający się wraz ze spadkiem zdrowia. Gracz posiada pole widzenia o określonym zasięgu, który działa w sposób realistyczny to znaczy będąc blisko wejścia do pokoju jest on w stanie zobaczyć tylko jego skrawek, a nie cały pokój, co widać na rysunku 2.18. Poza aktualnie widzianymi polami postać gracza pamięta też obszary, które wcześniej zobaczyła, są one zaznaczone na szaro. Na obszarach, które gracz pamięta, ale nie widzi ich obecnie zaobserwować można tylko układ podłóg i ścian, przeciwników i przedmioty zauważyci można tylko w bezpośrednim polu widzenia.



Rysunek 2.18: Widok główny gry. 1 - aktualny wycinek mapy, 2 - dziennik zdarzeń, 3 - aktualny poziom, 4 - pasek zdrowia. Żółta postać to gracz, a zielone postacie to gobliny – jeden z typów przeciwników

2.6. Przedmioty oraz ich używanie

W grze gracz przez eksplorację może znaleźć wiele różnorodnych przedmiotów, których używanie jest konieczne w celu ukończenia gry. Przedmioty dzielą się na zbroje, bronie, mikstury oraz magiczne zwoje. Aby podnieść przedmiot gracz musi przejść na zajmowane przez przedmiot pole a następnie wcisnąć klawisz 'G'. Na rysunku 2.19 przedstawiono pokój wypełniony przedmiotami.



Rysunek 2.19: Pokój z przedmiotami. 1 - postać gracza, 2 - mikstura zdrowia, 3 - broń, 4 - zbroja

2.6.1. Okna ekwipunku

Gracz w każdej chwili może sprawdzić swój ekwipunek klawiszem 'I', ekran ten zaprezentowano na rysunku 2.20. Wyboru przedmiotu dokonuje się za pomocą strzałek i klawisza Enter, litera po lewej stronie przy aktualnie wybranym przedmiocie jest zaznaczona na zielono. Innym sposobem wybrania przedmiotu jest wcisnięcie przypisanej mu po lewej stronie litery w nawiasach. Dopus '<EQUIPPED>' po prawej stronie danego przedmiotu oznacza, że jest on aktualnie założony przez postać gracza. Zielone liczby oznaczają ilość danego przedmiotu, na rysunku 2.20 zaobserwować można, że gracz posiada cztery mikstury uzdrawiającą 'Health potion'. Klawiszem Esc można wyjść z menu ekwipunku i powrócić do gry.



Rysunek 2.20: Okno ekwipunku

Po wybraniu przedmiotu wyświetla się menu wyboru akcji dostępnych dla każdego przedmiotu, przykładowe dostępne akcje dla 'Health potion' przedstawiono na rysunku ???. Każdy przedmiot można wyrzucić za pomocą akcji 'drop', spowoduje to wyrzucenie przedmiotu na pole zajmowane przez gracza. W przypadku nie założonych zbroi i broni dostępna jest opcja 'Equip' powodująca wyekwipowanie danego przedmiotu. Konsekwentnie założone przedmioty posiadają akcję 'UnEquip', która pozwala na ściągnięcie danego przedmiotu. Mikstury i zwoje posiadają akcję 'Use' pozwalającą na użycie danego przedmiotu. Niektóre przedmioty są używane od razu na postaci gracza, a niektóre pozwalają na wybór celu.



Rysunek 2.21: Okno akcji przedmiotu z ekwipunku

2.6.2. Zbroje i bronie

Każda postać w grze posiada wartość obrony, która obniża wartość otrzymywanych obrażeń. Obronę zwiększyć można za pomocą zbroi zakładanych na poszczególne części ciała: głowę, tors, ręce oraz nogi. W grze w kolejnych poziomach spotkać można coraz lepsze zbroje, dzięki czemu postać gracza jest coraz wytrzymalsza.

Podobnie jak obronę, postać gracza oraz przeciwnicy posiadają wartość ataku, która określa ilość zadawanych w trakcie ataku obrażeń. Wartość tę można zwiększyć ekwipując bronie rozsiane po poziomach. Lepsze bronie znaleźć można w późniejszych poziomach, dzięki czemu gracz, który dokładnie eksploruje plansze będzie zadawał coraz więcej obrażeń. Reprezentację graficzną zbroj oraz broni zaobserwować można na rysunku 2.19.

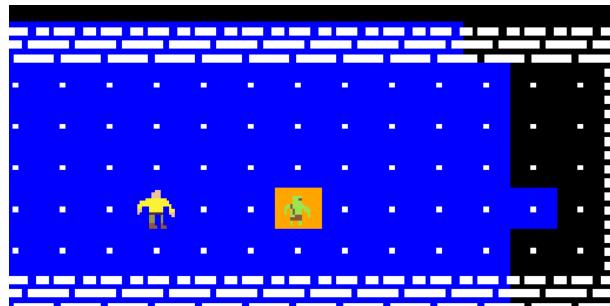
2.6.3. Mikstury

Postać gracza posiada określoną ilość punktów zdrowia 'HP', które zmniejszają się wskutek ataków przeciwników. Aby zregenerować zdrowie gracz musi znaleźć mikstury uzdrawiające. W grze dostępne są dwie wersje tych mikstur – zwykła 'Health potion' oraz większa 'Great health potion', która leczy większą ilość punktów zdrowia. Użycie mikstur jest natychmiastowe, a celem zawsze jest postać gracza. Wygląd mikstur został pokazany na rysunku 2.19.

2.6.4. Zwoje

Kolejnym typem przedmiotów są magiczne zwoje. Zwoje zapewniają wiele różnych efektów, a ich użycie wymaga wyboru celu, co zaprezentowano na rysunku 2.22. Cel ataku zaznaczony jest pomarańczowym kolorem, niebieskie pola oznaczają

pola będące w zasięgu danego zwoju. Cel wybrać można przesuwając znacznik celu klawiszami używanymi do przesuwania postaci lub kliknięciem myszki na dane pole. Na rysunku 2.23 pokazano wygląd zwojów w grze.



Rysunek 2.22: Wybór celu magicznego zwoju



Rysunek 2.23: Magiczny zwój w grze

Dostępne w grze zwoje to

- 'Magic missile scroll' – zwój magicznego pocisku, zadaje obrażenia wybranemu celowi,
- 'Fireball scroll' – zwój ognistej kuli, zadaje obrażenia przeciwnikom w małym promieniu od wybranego celu,
- 'Sleep scroll' – zwój uśpienia, usypia wybrany cel na kilka tur – nie będzie się on poruszał ani atakował,
- 'Area sleep scroll' – zwój obszarowego uśpienia, usypia kilka przeciwników w małym promieniu od wybranego celu,
- 'Teleport scroll' – zwój teleportacji, natychmiast przenosi postać gracza w wybrane miejsce.

2.7. Tester proceduralnie generowanych poziomów

Gry roguelike opierają się w głównej mierze na proceduralnie generowanych poziomach, co oznacza generowanie map na podstawie algorytmów zawierających elementy losowości. Oznacza to na przykład, że użyty w niniejszej grze algorytm do generowania jaskiń zawsze wygeneruje poziom przypominający jaskinię, lecz za każdym razem inaczej wyglądający. Takie podejście do tworzenie poziomów pozwala znacznie zaoszczędzić czas, który w standardowej metodzie byłby spędzony na ręcznie tworzenie poziomów. Mapy proceduralnie generowane są jednak podatne na błędy – na przykład miejsca, do których nie da się dojść. W celu szybszego odnajdywania i naprawiania takich i podobnych błędów, a także w celu prezentacji możliwości zaimplementowanych algorytmów stworzono menu do testowania generatorów map, zaprezentowane na rysunku 2.24. Menu to dostępne jest w głównym menu gry po wybraniu opcji 'Test Map Generators'.



Rysunek 2.24: Menu testera proceduralnie generowanych map

Klawiszami strzałek oraz klawiszem Enter wybiera się opcję, klawiszem Esc można powrócić do głównego menu gry. Opcje dostępne w menu to:

- a) 'Show steps' – zmiana opcji pokazania kroków generacji w trakcie testowania generatorów. 'true' oznacza włączenie pokazu kroków, 'false' wyłączenie,
- b) 'Test Basic Dungeon Generator' – testowanie generacji poziomu 1,
- c) 'Test Cellular Automata Cave Generator' – testowanie generacji poziomu 3,
- d) 'Test Binary Space Partitioning dungeons Generator' – testowanie generacji poziomu 2.

- e) 'Test Binary Space Partitioning Interior Generator' – testowanie generacji poziomu 5.
- f) 'Test Drunkard walk map generator' – testowanie generacji poziomu 4.

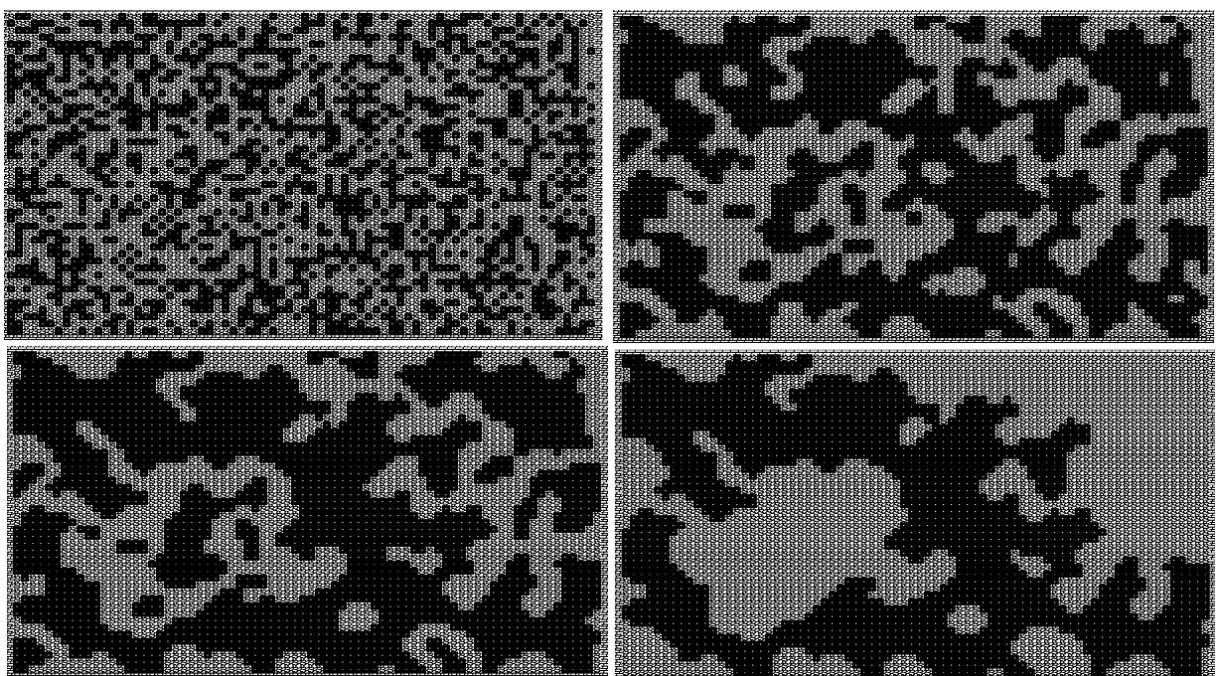
Po wybraniu generatora map otwiera się okno zaprezentowane na rysunku 2.25.

W głównej części zaprezentowany jest aktualny stan mapy w przypadku pokazu z krokami, lub finalny efekt generacji planszy w przypadku wyłączenia kroków. W dolnej części okna znajdują się informacje o numerze aktualnego kroku, oraz jego opisie. Klawiszem Esc powrócić można do menu testera, klawiszem spacji generuje się nową mapę lub przechodzi do kolejnego kroku.

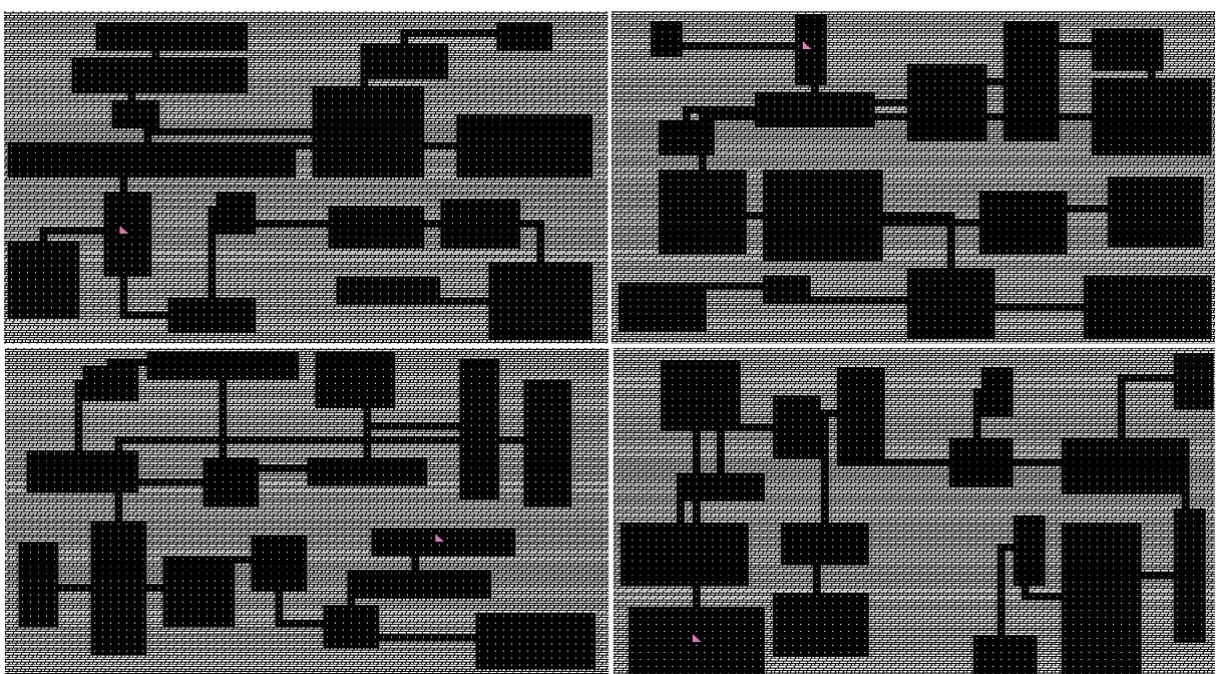


Rysunek 2.25: Okno testera wybranego generatora map

Na rysunku 2.26 przedstawiono część kroków generacji mapy do poziomu trzeciego, na rysunku 2.27 pokazano przykładowe wygenerowane mapy do poziomu drugiego.



Rysunek 2.26: Przykład testowania z krokami generacji mapy do poziomu trzeciego



Rysunek 2.27: Przykład testowania bez kroków generacji mapy do poziomu drugiego

3. Technologie i implementacja

W niniejszym rozdziale opisano technologie, architekturę oraz algorytmy użyte do implementacji gry.

3.1. Technologie

Do zaprogramowania gry użyto języka Rust, będącego kompilowalnym językiem ogólnego przeznaczenia, powstały w 2010 roku [14]. Język Rust wydajnościowo jest porównywalny do języka C++, lecz jest to język nowocześniejszy i bezpieczniejszy [15]. Język jest dużo bardziej restrykcyjny już na poziomie komplikacji, dzięki czemu błędy są częściej wykrywane w trakci komplikacji, niż dopiero po uruchomieniu programu. Użyto też wbudowanych w język narzędzi do formatowania 'cargo fmt' oraz lintera 'cargo clippy' – który wykrywa niegroźne błędy nie wykrywane przez kompilator, ale utrudniające czytelność kodu. Te cechy oraz narzędzia języka Rust znacznie ułatwiają stosowanie zasad czystego kodu [16].

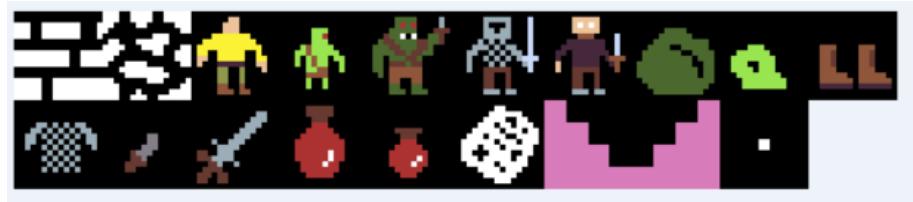
Do edycji kodu użyty został program Visual Studio Code [17] z rozszerzeniami dla języka Rust, przede wszystkim 'rust-analyzer' zapewniający kolorowanie składni, oraz automatyczne uzupełnianie kodu.

W programie użyto wielu tworzonych przez społeczność bibliotek open source – zapewniających prawo do korzystania, oraz dostęp do kodu źródłowego. Najważniejsze użyte biblioteki to:

- 'rltk' – tworzenie okna gry, obsługa grafiki oraz pętli głównej gry,
- 'specs' – implementacja architektury Entity Component System.

Gra posiada grafikę typu pixel art w formacie 16x16 pikseli, w oknie gry użyto dwukrotnie powiększonych obrazków w formacie 32x32 w celu polepszenia widoczności. Aby móc zmieścić całą mapę w jednym oknie w testerze proceduralnych generatorów map, użyty został oryginalny format 16x16. Do tworzenia grafiki użyto programu Aseprite [18] będącego wygodnym narzędziem graficznym wyspecjalizowanym w tworzeniu grafiki pixel art. Wszystkie obrazki użyte w grze znajdują się w folderze 'resources' w dwóch plikach, odpowiednio 'sprite_sheet_16x16.png' oraz 'sprite_sheet_32x32.png'. Taki format przechowywania obrazków zapewnia porządek w folderach gry, dostęp do konkretnego obrazka odbywa się przez określenie wymiarów oraz podaniu indeksu. Aby

uzyskać obrazek przedstawiający gracza z pliku 'sprite_sheet_16x16.png' przedstawionego na rysunku 3.28 należy określić rozmiary – 16x16 pikseli oraz podać indeks 2 – obrazki są indeksowane od lewej do prawej, pierwszy element ma indeks zero.



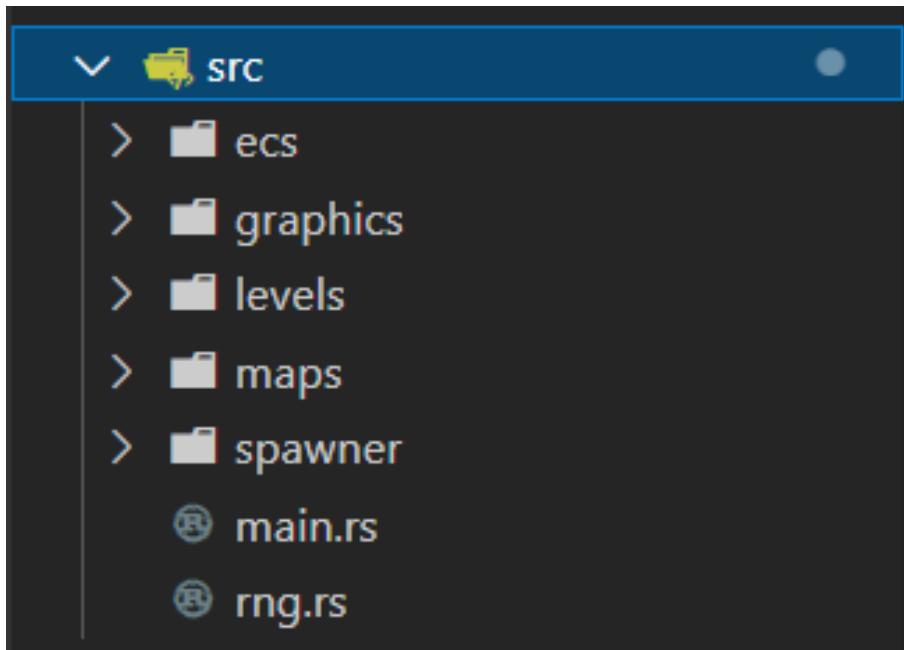
Rysunek 3.28: Grafiki użyte w grze. Od lewej: ściana, skały, postać gracza, gooblin, ork, rycerz, łotr, blop, blip, buty, zbroja, sztylet, miecz, duża mikstura zdrowia, mikstura zdrowia, magiczny zwój, schody w dół, schody w górę, podłoga

3.2. Implementacja

W niniejszym podrozdziale przedstawiono szczegóły implementacyjne, wraz z prezentacją fragmentów kodu źródłowego, wybranych elementów gry.

3.2.1. Struktura projektu

Gra roguelike posiada wiele modułów spełniających odmienne zadania. Kod projektu podzielony został na wiele modułów i podmodułów w celu lepszej organizacji. Główne folder programu zawiera foldery: 'resources' – zawierający grafiki użyte w grze, 'src' – zawierający kod źródłowy, na rysunku 3.29 przedstawiono schemat głównych modułów kodu projektu. W folderze głównym znajduje się też plik Cargo.toml określający między innymi użyte biblioteki.



Rysunek 3.29: Schemat głównych modułów struktury projektu

Moduł 'ecs' zawiera podstawowy plik projektu 'game_state.rs' będący maszyną stanów oraz obsługujący główną pętlę gry. W tym module znajdują się też podmoduły:

- 'components' – definicje komponentów,
- 'systems' – zaiwera systemy, dzieli się na kolejne podmoduły:
 - 'ai' – systemy sterujące komputerowymi przeciwnikami,'
 - 'combat' – systemy związane z walką i zadawaniem obrażeń,
 - 'effects' – efekty takie jak leczenie i teleportacja,
 - 'inventory' – systemy zarządzania ekwipunkiem, podnoszenia i używania przedmiotów,
 - 'map' – systemy związane z mapą, aktualizacja zablokowanych pól,
 - 'player' – systemy poruszania i akcji postaci gracza, oraz odczytywania wcisniętych klawiszy,
 - 'spawn' – system umieszczania obiektów na poziomie, /item 'view_system' – system obliczania pola widzenia gracza oraz przeciwników.

Moduł 'graphics' odpowiada za grafikę: tworzenie okna gry, rysowanie mapy i okien interfejsu. Znajdują się w nim funkcje do rysowania mapy gry, oraz obiektów:

przedmiotów i postaci. Posiada podmoduł 'menus' z definicją struktur odpowiedzialnych za poszczególne okna w grze, a także definicję trait-u 'WindowOptionSelector', który implementuje większość tych struktur.

Moduł 'levels' posiada definicję struktury 'Level' odpowiedzialnej za poziom, oraz 'LevelManager' odpowiedzialnej za przechowywanie i tworzenie nowych poziomów.

W module 'map' zdefiniowana jest struktura przechowująca kształt mapy: 'Map', definicja typów pól na mapie, oraz pomocnicze funkcje związane z mapą. Zawiera też podmoduł 'generators' zawierający generatory poszczególnych typów map.

Moduł 'spawner' jest odpowiedzialny za tworzenie obiektów w grze takich jak: postać gracza, przeciwnicy, przedmioty i ekwipunek. Zawiera też definicję struktur 'SpawnTable' odpowiadających za rozmieszczanie obiektów na konkretnych poziomach.

Plik 'main.rs' zawiera funkcję główną programu: 'main()'. W tej funkcji deklaruje się podstawowe konfigurację takie jak rozmiary map i okna gry, a także uruchamiana w niej jest pętla główna gry.

W pliku 'rng.rs' zawarto funkcje do losowania liczb używane w generatorach map, a także przy rozmieszczaniu na nich przeciwników i przedmiotów.

3.2.2. Architektura Entity Component System

Twórcy gier gatunku roguelike dzięki prostej grafice mogą poświęcić więcej czasu na tworzenie skomplikowanych, interesujących mechanik. Szczególnie przydatną w tworzeniu takich skomplikowanych mechanik jest architektura Entity Component System (Encja, Komponent, System), nazywana dalej ECS TODO [zrodlo ecs], skupiająca się na rozdzieleniu danych od systemów. Dzięki użyciu biblioteki 'specs' pominięto etap implementowania architektury ECS, skupiono się wyłącznie na jej użyciu. Elementami ECS są :

- 'Entity' – encja, jest to indeks reprezentujący dany obiekt w grze na przykład gracza, przeciwnika lub przedmiot,
- 'Component' – komponent, są to części składowe obiektów w grze przypisywane do konkretnych encji,
- 'System' – części logiczne architektury, konkretne systemy aktualizują stany konkretnych komponentów,

- w niektórych implementacjach stosowany też jest czwart element 'Event' – wydarzenie, służący do informowania konkretnych obiektów o takich wydarzeniach jak na przykład zadanie obrażeń, użycie przedmiotu. W przypadku tej gry wydarzenie zostały zrealizowane za pomocą odpowiednich komponentów.

Komponenty to małe części składowe, których stan aktualizowany jest przez konkretne systemy. Komponenty tworzone są jako struktury implementujące trait 'Component' z biblioteki 'specs'. Definicje komponentów znajdują się w pliku 'src/ec-s/components/mod.rs'. Przykładowe komponenty przedstawione na rysunku 3.30 to:

- 'BlocksTile' – struktura bez pól, informuje, że dany obiekt blokuje aktualnie zajmowane pole, i nie może się na nim znaleźć inny obiekt posiadający 'BlocksTile',
- 'Hp' – informuje o maksymalnej i aktualnej liczbie punktów zdrowia. Posiadacze tego komponentu mogą zostać zranieni,
- 'CombatBaseStats' – statystyki bojowe: atak i obrona, informują odpowiednio o ilości zadawanych obrażeń w trakcie walki i o ilości blokowanych obrażeń,
- 'WantsToMeleeAttack' – komponent-wydarzenie dodawany do obiektu chcącego dokonać ataku, informuje o celu ataku,
- 'SufferDamage' – komponent-wydarzenia dodawany do obiektów, którym zadane zostały obrażenia. Ilości obrażeń przedstawione są jako wektor, gdyż w jednej turze jeden obiekt może zostać zaatakowany przez kilku przeciwników.

```
#[derive(Component, Debug, Clone)]
pub struct BlocksTile {}

#[derive(Component, Debug, Clone)]
pub struct Hp {
    pub max_hp: i32,
    pub hp: i32,
}

#[derive(Component, Debug, Clone)]
pub struct CombatBaseStats {
    pub attack: i32,
    pub defense: i32,
}

#[derive(Component, Debug, Clone)]
pub struct WantsToMeleeAttack {
    pub target: Entity,
}

#[derive(Component, Debug, Clone)]
pub struct SufferDamage {
    pub amount: Vec<i32>,
}
```

Rysunek 3.30: Przykładowe struktury komponentów

4. Podsumowanie i wnioski końcowe

problemy i rozwiazania: - mala widoczosc - powiekszono widok gry - przesuwajaca sie za graczem kamera

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/
- [13] Stefanie Fogel, 'The Binding of Isaac' Card Game Surpasses 1 Million on Kickstarter in Just Over a Week, <https://variety.com/2018/gaming/news/the-binding-of-isaac-four-souls-kickstarter-1202868455/>

- [14] Steve Klabnik , Carol Nichols, The Rust Programming Language, No Starch Press, 2019.
- [15] Ben Fenwick, C++ v. Rust -Speed, Safety, & Community Comparison, <https://devetry.com/blog/c-v-rust-speed-safety-community-comparison/> .
- [16] Robert C. Martin, Czysty kod. Podręcznik dobrego programisty, Helion, 2014.
- [17] Microsoft, Visual Studio Code, 2015, <https://code.visualstudio.com> .
- [18] Igara Studio S.A, Aseprite, 2001, <https://www.aseprite.org>.

POLITECHNIKA RZESZOWSKA im. I. Łukasiewicza
Wydział Elektrotechniki i Informatyki

Rzeszów, 2022

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

RZESZOW UNIVERSITY OF TECHNOLOGY
Faculty of Electrical and Computer Engineering

Rzeszow, 2022

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