



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Wydział Matematyki i Informatyki

Kamil Tyrek, Mateusz Hyps, Jakub Kozubal
Numer albumu: 434797, 434699, 434726

Projekt i implementacja gry „The Lore: Story of the fallen warrior”

Project and implementation of game „The Lore: Story of the fallen warrior”

Praca inżynierska na kierunku **informatyka**
napisana pod opieką
dr Bartłomieja Przybylskiego

Poznań, styczeń 2021

Poznań, 4 stycznia 2021 r.

Oświadczenie

Ja, niżej podpisany **Kamil Tyrek, Mateusz Hypś, Jakub Kozubal**, student Wydziału Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu oświadczam, że przedkładaną pracę dyplomową pt. *Projekt i implementacja gry „The Lore: Story of the fallen warrior”* napisałem samodzielnie. Oznacza to, że przy pisaniu pracy, poza niezbędnymi konsultacjami, nie korzystałem z pomocy innych osób, a w szczególności nie zlecałem opracowania rozprawy lub jej części innym osobom, ani nie odpisywałem tej rozprawy lub jej części od innych osób. Oświadczam również, że egzemplarz pracy dyplomowej w wersji drukowanej jest całkowicie zgodny z egzemplarzem pracy dyplomowej w wersji elektronicznej. Jednocześnie przyjmuję do wiadomości, że przypisanie sobie, w pracy dyplomowej, autorstwa istotnego fragmentu lub innych elementów cudzego utworu lub ustalenia naukowego stanowi podstawę stwierdzenia nieważności postępowania w sprawie nadania tytułu zawodowego.

[TAK] wyrażam zgodę na udostępnianie mojej pracy w czytelni Archiwum UAM

[TAK] wyrażam zgodę na udostępnianie mojej pracy w zakresie koniecznym do ochrony mojego prawa do autorstwa lub praw osób trzecich

Spis treści

Rozdział 1. Wstęp	5
1.1. Cel i założenia projektu	5
1.2. Organizacja pracy	6
1.3. Podział prac	7
1.3.1. Kamil Tyrek - tworzenie elementów logicznych i ich algorytmika	7
1.3.2. Mateusz Hyps - Zarządzanie projektem gry komputerowej z wykorzystaniem Agile	7
1.3.3. Jakub Kozubal - Fizyka postaci	7
1.4. Użyta technologia - Unity	8
Rozdział 2. Zarządzanie projektem gry komputerowej z wykorzystaniem Agile	9
2.1. Zarządzanie projektem informatycznym	10
2.1.1. Proces zarządzania projektem i jego geneza	10
2.1.2. Cykl życia projektu informatycznego	10
2.1.3. Metody pomocne w zarządzaniu projektami informatycznymi	10
2.1.4. Ryzyko oraz powody niepowodzeń projektów informatycznych	10
2.2. Zarządzanie projektem gry komputerowej - The Lore	10
2.2.1. Proces planowania projektu	10
2.2.2. Wykorzystane systemy i metody projektowe	10
2.2.3. Rola i wpływ kierownika projektu	10
2.2.4. Cykl życia projektu The Lore	10
Rozdział 3. Zagadki logiczne	11
3.1. Przesuwane puzzle	11
3.1.1. Omówienie zagadnienia i ogólne założenia	11
3.1.2. Algorytmika	12
3.1.3. Przedstawienie przykładu w grze The Lore	14
3.2. Zagadka z rurami	14
3.2.1. Omówienie zagadnienia	14
3.2.2. Algorytmika	14
3.2.3. Przedstawienie przykładu w grze	14
3.3. Zagadka z otwieraniem skrzyni	14
3.3.1. Omówienie zagadnienia	14
3.3.2. Algorytmika	14

3.4. Labirynt	14
3.4.1. Omówienie zagadnienia	14
3.4.2. Algorytmika	14
3.4.3. Przykłady użycia	14
Rozdział 4. Fizyka postaci	15
4.1. Tworzenie postaci	16
4.1.1. Omówienie zagadnienia procesu tworzenia postaci	16
4.1.2. Proces graficzny tworzenia postaci	16
4.1.3. Wdrożenie postaci do projektu w Unity	16
4.1.4. Sposoby animowania postaci	16
4.2. Poruszanie się	16
4.2.1. Omówienie zagadnienia	16
4.2.2. Fizyka w grach 2D, a w prawdziwym świecie	16
4.2.3. Poruszanie się oraz kolizje postaci	16
4.2.4. Skakanie	16
4.2.5. Otoczenie wpływające na fizykę postaci	16
4.2.6. Umiejętności związane z poruszaniem się	16

ROZDZIAŁ 1

Wstęp

1.1. CEL I ZAŁOŻENIA PROJEKTU

Celem projektu jest stworzenie gry platformowej, zawierającej elementy zręcznościowe oraz łamigłówek. Głównym założeniem projektu jest gra, która zacieka swoją fabułą oraz trudnością. Wstępnie prosty zamysł samouczka, w większości gier jest raczej elementem wprowadzającym do rozgrywki, która powinna robić się trudniejsza w im dalszym etapie rozgrywki się znajdujemy, w przypadku naszej gry jest odwrócony. Samouczek poza wprowadzaniu poszczególnych elementów rozgrywki, którymi jest między innymi przedstawienie sterowania oraz funkcjonalności i wstępnych mechanik jest również wymagający, od gracza zależy jakie umiejętności w trakcie rozgrywki będzie rozwijać, aby przejście kolejnych poziomów było łatwiejsze (z perspektywy gracza). Dużą wagę w projekcie przywiązujemy do mini-gier, które występują w trakcie przechodzenia poszczególnych poziomów. Występują dwa rodzaje mini-gier: opcjonalne (te które przejść możemy w celu sprawdzenia siebie oraz zdobycia punktów doświadczenia) oraz wymagane, które trzeba przejść, aby znaleźć się w dalszym etapie gry. Użytkownik posiada do dyspozycji punkty doświadczenia, drzewko umiejętności oraz ekwipunek. Punkty doświadczenia możemy wydawać bezpośrednio w drzewku umiejętności, w którym zadaniem gracza jest wybranie odpowiednich umiejętności zależnie od tego jaką strategię rozgrywki chce przyjąć. Warto pamiętać jednak, że im głębiej będziemy rozwijać daną gałąź tym umiejętności będą bardziej pomocne co sprawia, że gracz musi zastanowić się dobrze nad decyzjami dotyczącymi rozwijania konkretnych umiejętności w drzewku. Ekwipunek służy do zdobywania przedmiotów potrzebnych w trakcie rozgrywki m.in. do otwierania drzwi czy rozpoczęcia opcjonalnej mini-gry. W projekcie są wykorzystane 2 style tworzenia poziomów: ortographic i perspective. Wszystkie animacje w projekcie są tworzone z użyciem technologii inverse

kinematics co pozwala na stałe dodawanie nowych animacji i poprawianie już istniejących.

1.2. ORGANIZACJA PRACY

Charakter projektu sprawił, iż nie można w naszym zespole jasno podzielić typów zadań, które realizujemy w ramach projektu. Wynikiem tego jest to, iż nowe funkcjonalności są realizowane zazwyczaj przez jedną osobę od początku do końca, nie ma podziału, podobnego jak w przypadku aplikacji z frontendem i backendem.

Jako metodykę pracy przyjęto Scrum. Głównymi powodami tej decyzji jest doświadczenie części zespołu w tej metodyce oraz przejrzystość i rozsądne zarządzanie pracą. Nie rozważano innych metodyk pracy. Przy zarządzaniu pracą wspomaga nas serwis JIRA, który pozwala zarządzać regularne sprinty – w pierwszym semestrze dwutygodniowe, w drugim semestrze tygodniowe.

Kod źródłowy zarządzany jest poprzez GitHub. Dla przejrzystości pracy, każdy commit na GitHub oznaczany jest id zadania na Jirze, dzięki czemu wchodząc w zadanie widzimy commit powiązany z jego rozwiązaniem. W momencie rozpoczęcia zadania deweloper, jeśli następuje potrzeba, tworzy podzadania do zadań na Jirze. Dotyczy to większych zadań, których wykonanie polega na tworzeniu większej ilości funkcjonalności. Dzięki temu realizując nowe zadania, o podobnej budowie, można sugerować się podobnym sposobem działania zadania. Po każdym commicie programista wyznacza ile czasu poświęcił na zadanie, poprzez wbudowaną w Jirze funkcjonalność "Log Work". Gdy zadanie zostanie zakończone, programista oznacza je statusem "DONE".

Z racji wybranej metodyki, dokonano również przydzielenia odpowiednich ról członkom zespołu. Podział ról wygląda następująco:

- Kamil Tyrek - Development Team, Scrum Master
- Jakub Kozubal -Development Team, Product Owner
- Mateusz Hypś - Development Team

1.3. PODZIAŁ PRAC

1.3.1. Kamil Tyrek - tworzenie elementów logicznych i ich algorytmika

W tym rozdziale zostaną przedstawione sposoby tworzenia elementów logicznych. Część z przedstawionych łamigłówek została użyta w projekcie końcowym. Przykładem są tutaj przesuwane puzzle - rozgrywka polegająca na przesunięciu elementu, celem ułożenia poprawnego obrazka. Algorytmika stojąca za losowaniem kolejności elementów nie jest trywialna, ponieważ źle wylosowana kolejność puzzli powoduje, iż mogą być niemożliwe do ułożenia.

Zostanie przedstawiona też logika mini-gry z ustawieniem odpowiednich ruch, celem połączenia dwóch końców rur. Podobnie jak w poprzednim przykładzie, do rozwiązania tej zagadki potrzebna jest odpowiednia liczba rur danego typu - pionowe, poziome, skątne.

Następnym przykładem będzie logika stojąca za rozgrywką, która nie jest dostępna w końcowym projekcie, a chodzi tutaj o generowanie labiryntu. Nie trywialnym problemem jest wygenerowanie takiej planszy, aby możliwe było przejście z punktu A do punktu B. W tym rozdziale postaramy się przedstawić rozwiązanie tego problemu, przy użyciu odpowiednich algorytmów.

1.3.2. Mateusz Hypś - Zarządzanie projektem gry komputerowej z wykorzystaniem Agile

W rozdziale zostanie opisany proces zarządzania projektem z wykorzystaniem metodyki zwinnej. Omówione i porównane zostaną podejścia Agile i Agile Game Development oraz zastosowania tych metodyk, w porównaniu z innymi stosowanymi w praktyce. Przedstawione zostaną najważniejsze idee stojące za metodykami Agile m.in. framework SCRUM. Następnie omówione zostanie zastosowanie tych metodyk w projekcie "The Lore", z uwzględnieniem problemów w trakcie realizacji tego projektu.

1.3.3. Jakub Kozubal - Fizyka postaci

W tym rozdziale zostaną przedstawione sposoby tworzenia poruszania się postaci. Zaznaczone zostaną główne różnice między fizyką rzeczywistą, a tą stosowaną w grach jak i przedstawienie wielu sposobów rozwiązania tej samej funkcjonalności. Ponadto przedstawiony zostanie proces tworzenia postaci. Do tego pojawi się też opisanie problemów takich jak sterowanie postaci w

powietrzu oraz oddziaływanie sił z otoczenia (m.in. poruszające się platformy). Zostanie także przedstawione i przeanalizowane działanie każdej umiejętności występującej w drzewku odpowiedzialnej za poruszanie się.

1.4. UŻYTA TECHNOLOGIA - UNITY

W naszym projekcie postanowiliśmy wybrać UNITY jako środowisko do stworzenia naszej gry. Wynikało to z możliwości jakie oferuje oraz z jakości i czytelności, stworzonej przez twórców, oficjalnej dokumentacji. Pozwala nam na tworzenie gier dwuwymiarowych czy trójwymiarowych oraz interaktywnych materiałów, na przykład animacje czy wizualizacje. W razie problemów możemy również wykorzystywać oficjalne forum na którym rzesza użytkowników dzieli się swoimi wskazówkami a także oficjalny sklep – Asset Store, w którym możemy wykupić materiały potrzebne do naszej gry. UNITY działa na każdym systemie operacyjnym, tj. Windows, macOS oraz Linux. Gwarantuje nam również możliwość stworzenia aplikacji nie tylko na komputery osobiste, ale także przeglądarki internetowe, konsole gier wideo oraz urządzenia mobilne. Dzięki aktualizacji silnika do wersji 5.1.1 ta lista wzrasta do 22 platform sprzętowych, w tym gogle wirtualnej rzeczywistości takie jak Oculus Rift. W przeszłości można było tworzyć aplikacje w trzech językach:

- UnityScript (swego rodzaju pochodna JavaScript'u)
- C#
- Boo

Jednak wraz z piątą wersją silnika (wydaną w roku 2015) możliwość pisania w języku Boo została usunięta, pozostała tylko wsteczna kompatybilność w postaci możliwości kompilacji skryptów przez środowisko MonoDevelop. Podobny los dotknął UnityScript, którego wsparcie zakończyło się na wersji 2018.2 (najnowsza wersja stabilna to 2019.3.4). Z tych względów nasz wybór musiał paść na język C#, w którym zostały napisane wszystkie nasze skrypty. Jako jedyny jest wciąż wspierany przez autorów, co zaowocowało drastycznym wzrostem popularności wśród użytkowników.

ROZDZIAŁ 2

Zarządzanie projektem gry komputerowej z wykorzystaniem Agile

2.1. ZARZĄDZANIE PROJEKTEM INFORMATYCZNYM

2.1.1. Proces zarządzania projektem i jego geneza

2.1.2. Cykl życia projektu informatycznego

2.1.3. Metody pomocne w zarządzaniu projektami informatycznymi

2.1.4. Ryzyko oraz powody niepowodzeń projektów informatycznych

2.2. ZARZĄDZANIE PROJEKTEM GRY KOMPUTEROWEJ - THE LORE

2.2.1. Proces planowania projektu

2.2.2. Wykorzystane systemy i metody projektowe

2.2.3. Rola i wpływ kierownika projektu

2.2.4. Cykl życia projektu The Lore

ROZDZIAŁ 3

Zagadki logiczne

3.1. PRZESUWANE PUZZLE

3.1.1. Omówienie zagadnienia i ogólne założenia

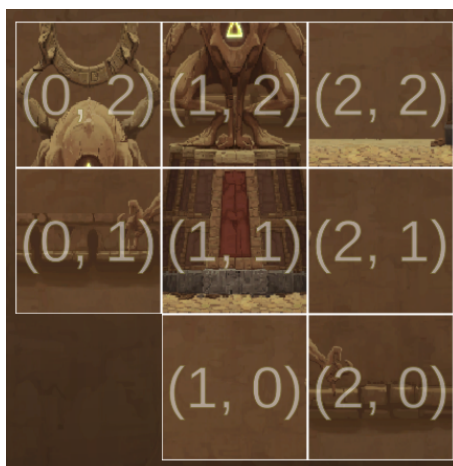
Przesuwane puzzle to układanka, złożona zazwyczaj z kwadratowej liczby elementów, najczęściej jest to szesnaście pól. Pola są jednakowych rozmiarów i oznaczone są liczbami od 1 do $(n-1)$, gdzie n to liczba dostępnych pól w układance. Jedno pole jest puste, pozwala to na przeniesienie sąsiednich elementów puzzli względem siebie. Rozgrywka kończy się, gdy ułożymy puzzle w odpowiedniej kolejności, według rosnącego porządku liczb lub powstania odpowiedniego obrazka. Trudno określić kto odpowiada za stworzenie zagadki. Wiadomym jest, że w 1878 roku pochodzący ze Stanów Zjednoczonych Samuel Loyd wypromował układankę, jednak prawdopodobnie nie jest to jego pomysł. Dość popularną nazwą na rozgrywkę jest "piętnastka", określającą ilość dostępnych pól w najpopularniejszym ułożeniu - 4×4 .

W grze The Lore gracze staną przez rozwiązaniem zagadki gdzie do dyspozycji mamy dziewięć pól. Podczas projektowania układanki w grze uznano, iż zagadka może być dość trudna, a korzyści płynące z rozwiązania jej będą nieadekwatne do straconego czasu, stąd ilość pól jest mniejsza niż w najpopularniejszej wersji rozgrywki.

3.1.2. Algorytmika

Pojedynczy element puzzle - PuzzleBlock

Każdy pojedynczy element puzzla jest zainicjalizowany jako obiekt, który określamy jako PuzzleBlock. Obiekt posiada koordynaty, które określają jego położenie w przestrzeni na układance.



Rysunek 3.1. Koordynaty każdego pola. Przykład z gry The Lore

Tak jak na załączonym przykładzie, wartość x rośnie w prawą stronę, zaś Y w górę, gdzie x dotyczy położenia w poziomie, a y w pionie. W założeniach przedstawiona została logika, która mówiła, iż element może zostać przemieszczony wtedy i tylko wtedy, gdy pole obok niego jest wolne, czyli nie posiada żadnego PuzzleBlock - elementu z liczbą lub obrazkiem. Dla lepszego efektu wizualnego, w grze The Lore element porusza się stopniowo, aby sprawiał wrażenie, iż porusza się realistycznie. Z racji, iż w Unity każda akcja wykonuje się podczas pokazywania kolejnej klatki, element porusza się w minimalnym stopniu przez sekundę. (TODO: tu można jakoś opisać, o ile jest sens - nie wpływa to na algorytmikę)

Plansza - Puzzle. Losowanie kolejności puzzli

Plansza rozgrywki posiada 8 elementów PuzzleBlock oraz jedno puste pole. Algorytm powinien wylosować dla 8 pól ich położenie na planszy - tak jak w wyżej przedstawionym przykładzie. W tym celu na początku losujemy dla każdego bloku wartość od 0 do 8. W C# możemy to zrobić przy użyciu funkcji `System.Random().Next(a, b)`. Przykładowo:

Wyciąg 3-1. Fragment klasy `Puzzle.cs`

```
1 private int randomPosition()
2 {
3     int pos = 0;
4     do
5     {
6         pos = new System.Random().Next(0, 9);
7     }
8     while (isOnBoard[pos]);
9     isOnBoard[pos] = true;
10    return pos;
11 }
```

Gdzie `isOnBoard[pos]` jest tablicą, która weryfikuje, czy dane pole nie jest już zajęte. Jeśli jest, ponownie losujemy wartość dla `PuzzleBlock`. Oczywiście to nie koniec - z tej wartości musimy stworzyć położenie w postaci (x, y) , gdzie x to położenie w poziomie, a y w pionie. W tym celu jedna z tych wartości będzie resztą dzielenia wylosowanej pozycji przez trzy (ilość pól w linii), a druga ilorazem całkowitym pozycji i liczby trzy.

3.1.3. Przedstawienie przykładu w grze The Lore

3.2. ZAGADKA Z RURAMI

3.2.1. Omówienie zagadnienia

3.2.2. Algorytmika

3.2.3. Przedstawienie przykładu w grze

3.3. ZAGADKA Z OTWIERANIEM SKRZYNI

3.3.1. Omówienie zagadnienia

3.3.2. Algorytmika

3.4. LABIRYNT

3.4.1. Omówienie zagadnienia

3.4.2. Algorytmika

3.4.3. Przykłady użycia

ROZDZIAŁ 4

Fizyka postaci

4.1. TWORZENIE POSTACI

4.1.1. Omówienie zagadnienia procesu tworzenia postaci

4.1.2. Proces graficzny tworzenia postaci

4.1.3. Wdrożenie postaci do projektu w Unity

4.1.4. Sposoby animowania postaci

4.2. PORUSZANIE SIĘ

4.2.1. Omówienie zagadnienia

4.2.2. Fizyka w grach 2D, a w prawdziwym świecie

4.2.3. Poruszanie się oraz kolizje postaci

4.2.4. Skakanie

4.2.5. Otoczenie wpływające na fizykę postaci

4.2.6. Umiejętności związane z poruszaniem się