Realizacja Projektu Informatycznego 2013/2014

**Projekt**

**Gra w Unity – The Mighty Marian**

Etap 1

Infrastruktura projektu

Autorzy:

Tobiasz Biernacki

Krzysztof Jasiak

Dominika Sokołowska

SPIS TREŚCI

1. Opis projektu i produktu
2. Organizacja zespołu projektu
3. Komunikacja w zespole
4. Dokumentacja
5. Współdzielenie dokumentów i kodu
6. Narzędzia wspomagające
7. Źródła
8. Opis projektu i produktu

Celem naszego projektu jest stworzenie gry przy użyciu silnika Unity, która łączyć będzie elementy survivalu oraz klasycznego RPG. Planujemy stworzyć produkt, który zapewni użytkownikowi wiele godzin niezapomnianej rozrywki i wyjątkową grywalność. Grafika będzie łączyć w sobie elementy 2D i 3D.

Główny bohater – Marian, to zwykły chłopak wychowany na skraju lasu, spragniony przygody i chwały. Razem z kolegami (Nieomylnym Mistrzem Gry Gerwazym oraz Złym Czarodziejem z Wieży - Andrzejem), przy jednym stole prowadzą rozgrywkę fabularną, w której duże znaczenie ma rzut kością. Aby dotrzeć do finału swojej przygody, Marian będzie musiał przebyć nieznane krainy, uciec przed wieloma potworami, zwiedzić niezliczone jaskinie, zebrać odpowiedni ekwipunek, nabrać krzepy i zaskarbić sobie przychylność mnóstwa losowych postaci, których problemami zostanie obarczony.

Projekt realizujemy w ramach pracy inżynierskiej. Planujemy stworzyć grę na systemy Windows i Android, dlatego dużą rolę odegra zaprojektowanie interfejsu i sytemu walki, które będą wygodne do obsługi na urządzeniach dotykowych, a jednocześnie będą zgodne z myszką i klawiaturą. Głównym elementem naszej gry, który ma znacząco poprawić grywalność i wydłużyć czas przyjemnej i pełnej niespodzianek rozgrywki, jest stworzenie map generowanych losowo, zarówno dla terenów otwartych, miasteczek, jak i lochów oraz rozmieszczenia na nich wrogów, sojuszników i przedmiotów w odpowiednich proporcjach.

The Mighty Marian to gra przeznaczona dla osób w różnym wieku. Ze względu na dużą ilość elementów humorystycznych i stosunkową trudność rozgrywki zapewni wiele godzin doskonałej zabawy dorosłym użytkownikom. Nieskomplikowana, utrzymana w kreskówkowo-pikselowym stylu grafika, przyciągnie najmłodszych graczy, którzy być może przyzwyczajeni do niewymagających za wiele od użytkownika gier, mogą zostać zniechęceni do dalszej zabawy. Naszym celem jest otrzymanie kategorii wiekowa to „od dwunastego roku życia” w klasyfikacji PEGI, ponieważ w grze występować może przemoc skierowana przeciwko fikcyjnym postaciom, a młodsze dzieci mogą przestraszyć się potworów, którym stawić czoła będzie musiał główny bohater. Docelowa grupa wiekowa to młodzież od dwunastego roku życia oraz młodzi dorośli, choć zachęcamy do rozgrywki wszystkich powyżej dolnej granicy wiekowej.

Projekt realizowany będzie w silniku Unity, z użyciem języka C#. Do zarządzania kodem użyjemy repozytorium SVN. Czas realizacji projektu jest ograniczony terminem egzaminu inżynierskiego.

1. Organizacja zespołu projektu

Nasz zespół składa się z trzech osób. Wszyscy w równym stopniu odpowiedzialni jesteśmy za koncepcję, projekt i końcowy wynik przedsięwzięcia. Ze względu na to, że od trzech lat studiujemy informatykę w jednej grupie, mamy raczej podobne doświadczenie i kompetencje.

|  |  |
| --- | --- |
| Krzysztof Jasiak | Umiejętności i role w zespole  Na co dzień pracuje z technologią .NET i C#, więc najlepiej z nas posługuje się językiem programowania, w którym tworzymy nasz projekt. Oprócz zadań które realizujemy wszyscy odpowiada również za zarządzanie konfiguracją oraz spójność dokumentacji.  Adres kontaktowy: sharow1234@gmail.com |
| Tobiasz Biernacki | Umiejętności i role w zespole  Najlepiej ze wszystkich członków zespołu orientuje się w silniku Unity i jego możliwościach. Ma też największą wiedzę o grafice 3D.  Adres kontaktowy: tobiasz931@gmail.com |
| Dominika Sokołowska | Umiejętności i role w zespole  Interesuje się algorytmami do generowania labiryntów oraz map, dlatego będzie wkładać najwięcej pracy w tą część projektu. Stworzy też znaczną część grafiki.  Adres kontaktowy: d.a.sokolowska@gmail.com |

Praca głównie wykonywana będzie w trybie rozproszonym, każdy będzie pracować w swoim domu. Jednak możliwe są również spotkania poświęcone projektowi, ponieważ spędzamy razem dużo czasu na uczelni.

1. Komunikacja w zespole

Komunikacja będzie odbywać w typowy dla nas sposób. Zespół liczy 3 osoby, więc nie stanowi problemu komunikowanie się głównie za pomocą konferencji na Skype. Poza tym o wielu aspektach będziemy rozmawiać korzystając z licznych okazji w których się widujemy. W najbardziej doraźnych przypadkach za krótką wymianę zdań będzie odpowiadał komunikator na facebooku i/lub poprzez code google. W razie potrzeby możliwe, że będziemy organizować spotkania, jednak przynajmniej na aktualnym etapie nie ma sensu precyzować konkretnej częstotliwości takich spotkań. Miejsce spotkań jest bliżej nie określone, ale można przyjąć, że byłby to budynek ETI. Komunikacja nie będzie sformalizowana. Różne zasoby będziemy sobie przekazywać za pomocą maili, lub serwisów takich jak dropbox. Za kod i dokumentacje odpowiadać będzie code google, jednak w praktyce może to ulec zmianie. Jeśli chodzi o komunikację zewnętrzną, to wszyscy jesteśmy w równy sposób reprezentantami projektu. Ludzi potencjalnie zainteresowanych naszym projektem zamierzamy znaleźć za pomocą blogu/forum (na przykład tworząc temat na forum o projektach w Unity). W ten sposób będzie też można się z nimi komunikować. Z promotorem projektu komunikujemy się mailowo i ewentualne spotkania odbywają się w budynku ETI.

1. Dokumentacja

Schemat nazewnictwa plików będzie zgodny z zasadami pisania kodu w języku C# zebranymi przez Lance Hunt, jest to najlepszy dokument zapewniający czystość kodu, trzymanie się standardów języka oraz poprawnych schematów nazw. Nad poprawnością składni kodu czuwać będzie dodatek do Visual Studio: StyleCop.

Projekt będzie składał się z plików .cs zawierających kod programu oraz plików projektowych unity.

Szablon dokumentu projektu:

<Nazwa klasy>

<Parametry>

<Opis Funkcjonalny>

Takie opisy będą pojawiać się na wiki projektu, dostępnej na code google.

Dążymy do pisania samo tłumaczącego się kodu- nazwy funkcji, klas i zmiennych powinny opisywać co dana rzecz robi.

Do wersjonowania projektu będziemy używać code google, systemem wersjonowania będzie Subversion (SVN), check-iny i check-outy kodu będą odbywać się dzięki wtyczce do Visual Studio.

Porządek w dokumentacji utrzymuje Krzysiek

1. Współdzielenie dokumentów i kodu

Korzystamy z code google, link do repozytorium: <https://code.google.com/p/themightymarian/>

Sposób dostępu: bycie na liście właścicieli projektu lub kontrybutorów, integracja z Visual Studio dzięki wtyczce VisualSVN

1. Narzędzia wspomagające

* Development:
  + Unity: <https://unity3d.com/>
  + Visual Studio 2012: <http://www.visualstudio.com/>
* Build server:
  + TeamCity[: http://www.jetbrains.com/teamcity/](file:///C:\Users\tobia_000\Downloads\:%20http:\www.jetbrains.com\teamcity\) (Server buildów zintegrowany z repozytorium)
* Zarządzanie zadaniami, agile board:
  + YouTrack: <http://www.jetbrains.com/youtrack/>
  + RedMine: <http://www.redmine.org/>
* Dodatki do Visual Studio:
  + ReSharper: <http://www.jetbrains.com/resharper/>(Testowanie, jakość kodu)
  + VisualSVN: [http://www.visualsvn.com/](http://www.visualsvn.com/%20) (Połączenie z repozytorium)
  + StyleCop: <http://stylecop.codeplex.com/> (Jakość kodu, poprawność ze standardem programowania)

1. Źródła

* Informacje o kategoriach wiekowych PEGI – czas dostępu: 10 mar 14

<http://www.pegi.info/pl/index/id/369/>

* C# coding standards by Lance Hunt -czas dostępu 10 03 2014 <http://weblogs.asp.net/lhunt/pages/CSharp-Coding-Standards-document.aspx>