POZNAN UNIVERSITY OF TECHNOLOGY

Tymoteusz Bleja
Paweł Husak
Patryk Imosa
Magalena Łątkowska

Internetowa gra edukacyjna ucząca podstaw pracy z programem git

Praca inżynierska

Promotor: dr hab. inż. Marek Andrzej Wojciechowski

Poznań, 2017

Spis treści

Rozdział 1

Wstęp

1.1 Zasady odnośnie pisania pracy

- Piszemy w formie bezosobowej. Można też niby w 1.os liczby mnogiej czyli Źrobiliśmy...", ale niektórzy akceptują tylko bezosobową, czyli Źrobiono...".
- słów niepolskich - „Nie można więc bezpośrednio w tekście używać słów angielskich. Jeżeli już — powinny być wyróżnione kursywą" - niektórzy podobno bardzo hejcą za angielskie słowa niestety.
- Jak chodzi o bibliografię, to w wolnej chwili dołączcie linki do stron z jakich korzystaliście, ja to potem ogarnę i zapiszę w takiej formie jak trzeba. Ale spokojnie, bo robienie bibliografii raczej zostawię na koniec.
- Ogólnie nie przejmujcie się strukturą, formatowaniem czy innymi formalnymi bzdetami, ja potem będę to ogarniać żeby było wg zasad więc nie traćcie czasu na ogarnianie takich rzeczy.

Między innymi: skąd wgl pomysł - bo Git jest super i konieczny a ciężko się go samemu nauczyć, nauka z wielu źródeł jest chujowa, większość ma tylko blade pojęcie a potem idzie do pracy i dupa - co z tego że nauczyli się na studiach programować jak nie potrafią korzystać z Gita i współpracować z zespołem

Cel i zakres pracy

cel - nauka fajna łatwa i przyjemna, oraz praktyczna, obycie z typowymi scenariuszami jakie mogą być potrzebne w pracy

Coś o tym dlaczego akurat przeglądarkowa gra, czemu z grafiką 3D itp.

Cytat z karty pracy : Zapoznanie się z systemem Git. Opracowanie koncepcji interaktywnego samouczka do nauki podstaw korzystania z systemu Git. Opracowanie architektury systemu. Implementacja i testowanie systemu. Przygotowanie dokumentacji technicznej i użytkowej.

Rozdział 2

Podstawy teoretyczne

2.1 Sysytem kontroli wersji Git

Może rozdział powinien się nazywać „Wstęp teoretyczny", albo od razu „System kontroli wersji Git" lub samo „Git", i wtedy bez tej podsekcji - i tak w rozdziale nie może być tylko jedna

Krótki opis systemów kontroli wersji, do czego służą i jakie dają korzyści. Wyjaśnienie dlaczego Git się wyróżnia, co ma szczególnego i bardziej obszerny jego opis.

Jakaś subsekcja o tym dlaczego programiści powinny go znać i potrafić używać jako narzędzia w pracy, do czego im się przyda i że Git często nie jest możliwością, lecz koniecznością.

Rozdział 3

Projekt

3.1 Założenia

Najważniejsze cele, co ma spełniać gra, czego ma nauczyć i w jaki sposób

3.2 Przebieg gry

Krótki opis rozgrywki i elementów interfejsu (może screeny? nie jestem pewna czy już w sekcji „projekt" czy dopiero w sekcji „implementacja")

3.3 Zadania

Ta sekcja nie tyle ma być o zadaniach w sensie Taskach, taskGraph.json itp, ale o zadaniach w kontekście co ma zrobić użytkownik, czego i w jaki sposób ma się przy tym nauczyć.

3.3.1 Konstrukcja zadania

Na początku krótki opis zadania, że składa się z kroków itd.

3.3.2 Podpowiedzi/Pomoc

Nie wiem jak nazwać tę subsekcję, bo HelpDrawer to nie do końca pomoc czy podpowiedzi, tylko content do nauczenia, nie umiem tego ładnie nazwać na razie.

Coś że gdy w zadaniu jest krok z nową komendą to wyskakuje pomoc i co w niej jest itp.

3.3.3 Punkty za rozwiązanie

Coś o punktacji, że zadania należy realizować w określonym czasie, dlaczego zrobiliśmy tak - aby samouczek był bardziej grą, wciągającą, ulubione słowo Tymona - gamifikacja, korzyści z takiego podejścia itp.

Opis konstrukcji zadania i kroku, z czego się składają itp. gdzieś raczej w implementacji

Coś o tagach i TagKnowledgeRatio???

Rozdział 4

Implementacja

4.1 Wykorzystane technologie

4.1.1 Redux

krótki opis i zalety, czemu wybraliśmy

4.1.2 React

krótki opis i zalety, czemu wybraliśmy

4.1.3 Babylon.js

krótki opis i zalety, czemu wybraliśmy

4.1.4 Pozostałe?

4.2 GraphCreator

Na początku może opis konstrukcji zadania i kroku, z czego się składają itp. Że są w jsonie i uciążliwe byłoby pisanie ich poprzez modyfikację jsona, dlatego właśnie powstał GraphCreator. Do czego służy, jak się z niego korzysta, potem subsekcja o implementacji.

4.3 Komponenty/elementy(?) aplikacji

4.3.1 Lista zadań

Screen jakiś, jak działa, o implementacji

4.3.2 Konsola

Screen jakiś, jak działa, o implementacji

4.3.3 Pomoc (HelpDrawer)

Screen jakiś, jak działa, o implementacji

4.3.4 Drzewko repo

Screen jakiś, jak działa, o implementacji

4.3.5 Canvas

Tutaj krótko co to, że wyłapuje zmiany stanu i jak wpisano dobrą komendę to przekazuje do Engine3D który jest odpowiedzialny za grafikę i robi co ma się stać.

4.4 Grafika 3D

Tu będzie sporo, do tego stopnie sporo, że nie wiem jeszcze jak to zaplanować i porozdzielać, względem czego.

Czy np podsekcje takie jak: repo3d - gałęzie, commity, co to są i jak powstają na akcje, o ich teksturze, obramowaniu, w tym o solidExplode ground - co to, jak działa itp particle w tle (wg elementów aplikacji)

Czy może raczej podsekcje wg 'elementów' Babylona: Meshe, SolidParticle, Particle, Materiały, Tekstury, Shadery

Rozdział 5

Podsumowanie

Dodatek A

Przewodnik użytkownika

Praktyczne info dla opornego użytkownika, jak ma korzystać, między innymi że jest opcja scrolla aby oddalić, jakie przyciski do obsługi helpa itp. itd., krótki opis fragmentu rozgrywki co się dzieje po czym i dlaczego i jak ma na to reagować użytkownik i takie tam.