

Falltown RPG

Projekt für Labor Games

SS2016

Gruppe Falltown

Dominik Kress

Theophil Bachmann

Brian Wirth

Valentin Petrov

Tom Görner

Inhalt

Von der Idee zum Spiel.....	3
Die Projektstruktur	3
Entwicklung des Projekts.....	6
Das Ergebnis	7
Persönlicher Anteil am Projekt der einzelnen Projektmitglieder.....	13
Dominik Kress	13
Valentin Petrov	14
Tom Görner	15
Brian Wirth	16
Theophil Bachmann.....	17

Von der Idee zum Spiel

Aus der Motivation ein etwas größeres Projekt als einen normalen Siderunner oder ähnlichem zu entwerfen beschlossen wir einen Open-World Titel für Smart-Phones zu erstellen. Aus erster Inspiration kam uns dazu ein Rollenspiel im Stile der alten Browser-Based Textrollenspiele wie Legend of the green dragon, nur als Top-Down Ansicht in einfachem Gameplay wie etwa Pokemon gemischt mit einer düsteren, dystropischen Atmosphäre a la Fallout.

Diese Spielidee hat sich im Laufe des Entwickelns im ganzen noch weiter gefestigt, jedoch in den Details immer mal wieder verändert.

Die Rollenverteilung im Team war schnell durch unsere Fähigkeiten klar. Theophil hatte bereits ein paar Erfahrungen mit Audio-Engineering und einen Bruder der ähnliches Studierte also das Know-How über Audio-Bearbeitung oder zumindest den Ansprechpartner um es zu erlangen. Valentin bekundete sein Interesse für die Erstellung der Assets und Animationen. Brian und Tom verstanden sich am besten auf das produktive Entwickeln des Software-Codes und Dominik auf das Projektmanagement.

Die Projektstruktur

Zur Projektplanung nutzten wir Jira und als Versionsverwaltung Bitbucket. Das ermöglichte uns eine präzise Planung und strukturierte Herangehensweise an die Entwicklung des Spiels.

Da wir uns entschlossen Scrum als Projektmanagement Methodik zu wählen, mussten wir die uns zur Verfügung stehende Zeit für die Dauer des Kurses ab Pitch der initialen Spielidee bis zur Abgabe des fertigen Spiels erst einmal in Sprints unterteilen. Wir entschlossen uns 2-wöchige Sprints durchzuführen und damit die gesamte Projektzeit in 5 Sprints aufzuteilen. Jeden Sprint wollten wir mit einem neuen Release einer Spielversion mit evolutionär erweitertem Inhalt beenden, wobei wir den letzten Sprint für abschließende Tests und vor allem Balancing-Anpassungen aufheben wollten.

Hierdurch ergab sich ein natürlicher Zeitplan und folgende Unterteilung des Gesamtprojekts in untenstehende Versionierungen:

Releases						Manage Versions
<input type="text" value="Q"/> QUICK FILTERS: Released Unreleased						
Version	Status	Progress	Start date	Release date	Description	
Prototyp	RELEASED	<div></div>	21/Apr/16	05/May/16	Der Prototyp des Spiels	
Code Freeze	RELEASED	<div></div>	02/Jun/16	16/Jun/16	Die Feature-Done Version des Spiels die zum Code Freeze führt	
Beta	RELEASED	<div></div>	19/May/16	02/Jun/16	Die Beta Version des Spiels	
Alpha	RELEASED	<div></div>	05/May/16	19/May/16	Die Alpha Version des Spiels	

Der Ablauf also nach dem ersten Sprint ein lauffähiger Prototyp, danach eine Alpha-Version, gefolgt von einer Beta, zuletzt der Code Freeze oder eher der Feature Freeze, die für die finale Version nur noch durch die Quality Assurance geprüft und verbessert werden sollte.

Diese Versionen mussten nun noch mit Inhalt gefüllt werden, wofür wir uns gemeinsam 8 große und grobe Themen in der Entwicklung des Spiels überlegten.

VERSIONS	EPICS
	Prototyp erstellen
	Falltown-Gebiet finalisieren
	Kampfmodus finalisieren
	Afterfall-Gebiet erstellen
	Bunker Dungeon erstellen
	Bossgegner Sieg - Spieler skillen erstellen
	Ingame Menü implementieren
	Netzwerk Arenamodus erstellen

Als Resultat hatten wir die nebenstehenden Epics entwickelt, wobei der Epic „Netzwerk Arenamodus erstellen“ aus dem Input der Kritik und Anregung unserer Mitstudenten beim Game-Pitch entstanden ist.

Wir werteten die Epics nach Wichtigkeit und logischer Reihenfolge bzw. möglicher Abhängigkeiten voneinander und verteilten sie auf die einzelnen Versionen.

Dabei entstand folgender Überblick:

Version Prototyp:

Prototyp erstellen

Version Alpha:

Falltown Gebiet finalisieren
Kampfmodus finalisieren

Version Beta:

Afterfall-Gebiet erstellen
Ingame Menü implementieren

Version Code Freeze:

Bunker Dungeon erstellen
Bossgegner Sieg - Spieler skillen erstellen
Netzwerk Arenamodus erstellen

Wir rechneten damit, dass sich unser Workflow mit der Einarbeitung in die Technologie und Erarbeitung wiederverwendbaren Codes stetig steigern würde, weshalb jede Release-Version gezielt einen größeren Umfang hatte als die vorherige.

Nun mussten wir noch jeden dieser Epics auf einzelne Storys herunterbrechen um genau verdeutlichen zu können, was in jedem Epic getan werden musste. Glücklicherweise hatte dabei jeder Epic in etwa eine ähnliche Struktur. Bei jedem Epic war nämlich folgendes wichtig:

- Erstellung der Design Dokumente
- Erstellung der Assets und ggfs. Animationen
- Erstellung der Audio-Sounds
- Positionierung der Assets/Animationen und Sounds
- Implementierung möglicher Logik
- Testen aller Funktionen

Diese einzelnen Storys wurden dann den jeweils zuständigen Rollen zugeordnet, so dass jeder seine Aufgaben und Verantwortungsbereiche genau kannte. Ein Idealbeispiel für diese Struktur ist etwa der Epic „Afterfall Gebiet finalisieren“ in der Beta Version.


Labor_Games_2016_SS_RPG / LGRPG-5

Kampf Gebiet um das Falltown-Gebiet erstellen.

Edit
Comment
Assign
More ▾
To Do
In Progress
Done

Issues in Epic

✓	LGRPG-218	Logik für Afterfall Gebiet erstellen	1	DONE	Brian Wirth
✓	LGRPG-88	Positionierungen der Assets, Animationen und Audio für Afterfall Gebiet	1	DONE	Tom Goerner
✓	LGRPG-87	Audio für Afterfall Gebiet erstellen	1	DONE	Theophil Bachmann
✓	LGRPG-83	Assets und Animationen des Afterfall-Gebiets erstellen	1	DONE	Valentin Petrov
✓	LGRPG-81	Design Entwurf für Afterfall-Gebiet entwerfen	1	DONE	Dominik Kress
✓	LGRPG-110	Testen aller Afterfall Gebiet Funktionen	1	DONE	Brian Wirth

Diese Storys wurden nun für jeden Epic erstellt und abgeschätzt um einen möglicherweise zu großen Arbeitsaufwand innerhalb einer Version frühzeitig zu erkennen und anzupassen. Unsere Epic-Schätzung stellte sich jedoch als recht solide heraus, wodurch es keine weiteren Anpassungen innerhalb der Versionen gab, jedoch während der Laufzeit des Projekts einige Anpassungen der Storys durch Ungenauigkeiten oder Unstimmigkeiten der zu Projektbeginn beschlossenen Story-Beschreibung.

Durch diese Unterteilung hatten wir schon recht früh eine recht genaue Abschätzung unserer erfordernten Arbeiten, jedoch keine genaue Auflistung unserer Arbeitspakete. Wir benötigten also noch eine weitere Schicht an Arbeitsunterteilungen unterhalb der etwas groben Story-Pakete.

Hierfür entwickelten wir vor jedem Sprint die Taskunterteilung unterhalb der Storys, wobei kein Task größer als eine Tagesarbeit sein sollte. Hierzu beispielsweise die Task-Unterteilung der oben aufgeführten Story „Design Entwurf für Afterfall-Gebiet entwerfen“.


Labor_Games_2016_SS_RPG / LGRPG-81

Design Entwurf für Afterfall-Gebiet entwerfen

Edit
Comment
Assign
More ▾
To Do
In Progress
Done

Sub-Tasks

1. ✓	Design Entwurf für Afterfall Deko Objekte	1	DONE	Dominik Kress	...
2. ✓	Design Entwurf für Afterfall Gegner erstellen	1	DONE	Dominik Kress	
3. ✓	Design Entwurf für Afterfall Positionierungen erstellen	1	DONE	Dominik Kress	

Auch diesen Arbeitspaketen wurden wieder die Personen mit entsprechenden Rollen zugewiesen, womit jeder eine genaue Auflistung seiner bis ins Detail präzisen Arbeitsaufgaben hatte.

Zudem wurde durch diese genaue Arbeitspaket Aufteilung eine recht deutliche Abhängigkeit zwischen einzelnen Arbeitspaketen sichtbar, die natürlich durch Jira auch vermerkt werden konnte und so für Transparenz innerhalb des Projektablaufs sorgte.


Labor_Games_2016_SS_RPG / LGRPG-88

Positionierungen der Assets, Animationen und Audio für Afterfall Gebiet

Edit
Comment
Assign
More
To Do
In Progress
Done

ISSUE LINKS

blocks	LGRPG-227 Befragung der Tester zum Afterfall Gebiet	↑	DONE
is blocked by	LGRPG-242 Design Entwurf für Afterfall Positionierungen erstellen	↑	DONE

Sub-Tasks

1. Positionierung der Audio für Afterfall Gebiet	LGRPG-227	DONE	Tom Goerner	...
2. Positionierung der Deko Assets für Afterfall Gebiet	LGRPG-242	DONE	Tom Goerner	

Bei Sprintstart wurden dann jeweils die einzelnen Tasks erledigt, bei Erledigung aller einer Story zugewiesenen Tasks die Story, bei Erledigung aller einem Epic zugewiesenen Stories der Epic und bei Erledigung aller einer Version zugewiesener Epics letztendlich diese Version released.

Eine genaue Übersicht der so entwickelten Arbeitspakete ist im Projekt in Jira unter folgendem Link abzurufen:

<https://jira.it.hs-heilbronn.de/projects/LGRPG/summary>

Eine genaue Auflistung aller Tickets auf Jira und damit den komplett Dokumentierten Projektumfang kann auch dem Excel Sheet „Jira-Backlog“ im Backlog Ordner dieser Dokumentationsstruktur entnommen werden.

Entwicklung des Projekts

Höchste Priorität durch die höchsten Abhängigkeiten hatte zu Beginn jeden Sprints immer die Erstellung der Design Dokumente. Diese wurden in einzelnen den Tasks der Design-Stories entsprechenden Kapiteln unterteilt und den jeweiligen Tasks als Anhang beigefügt. Dabei handelt es sich bei den Design-Dokumenten weniger um Richtlinien wie etwas auszusehen habe, sondern grobe Orientierungen wie es aussehen könnte. Die einzelnen Zuständigen der Umsetzung (zum Beispiel für die vorgeschlagenen Assets, Positionierung etc.) hatten dabei immer „das letzte Wort“ je nach ihren Möglichkeiten der Umsetzung. Probleme gab es hierbei keine, auch weil das Team an den wenigen nennenswerten Gelegenheiten konstruktiv über die Design-Entwürfe oder Abweichungen davon diskutieren konnte.

Alle Design-Entwürfe sind auch noch im Ordner Design innerhalb dieser Dokumentationsstruktur zu finden. Die Datei „GesamtDesign.pdf“ ist dabei eine zusammengefasste Version aller einzelnen Taskumfassender Kapitel.

So wurden beispielsweise die Gegner von einer Vielzahl verschiedener Kreaturen von Zombies über kämpfende Alltagsgegenstände bis hin zu mutierten Tieren aufgrund der Vereinfachung zu lediglich verschiedenartigen Menschen verändert oder eine Ursprünglich vom Design-Entwurf abweichende Positionierung der Häuser innerhalb des Falltown Gebiets nochmal erneut Überarbeitet um dem ursprünglichen Design-Entwurf hierzu zu entsprechen.

Assets, Sprites und Audiofiles konnten dank unserer gewählten Web-Technologie einfach innerhalb des Projektordners abgelegt und somit im Versionskontrollsystem Git hinterlegt werden. Um dieses Versionskontrollsystem als erste Instanz unserer Tests zu haben und auch hier strukturiert vorzugehen war der Plan einen langen Weg vom Master hinweg abzuzweigen.

Hierbei sollte es für jede Version einen Branch vom Master geben. Dieser Versionsbranch sollte sich dann in die einzelnen im zugehörigen Epics verzweigen, die dann Branches für die einzelnen ihnen zugehörigen Stories halten, falls diese es benötigten. Bei jedem Abschluss einer Arbeit am einzelnen Storybranch sollte um zur nächst höheren Instanz zu kommen ein Pullrequest gestellt werden, der dann jeweils von mindestens 2 Prüfern bestätigt werden sollte. Hierzu hätten wir dann ein System etabliert in dem der Master ausnahmslos immer stabil wäre, da eine Änderung oder Erweiterung mehrere Instanzen und Integrationen hinweg getestet werden würde.

Aufgrund des Aufwands und der Komplexität dieses Plans etablierte sich jedoch recht schnell vom Master selbst verschieden Feature-, Bugfix-, oder Hotfix-Banches abzuzweigen. Dass jegliche Änderung lediglich über von mindestens 2 Personen geprüften Pull-Requests erfolgen sollte blieb jedoch bestehen. Jeder direkte Push in den Master wurde mit einer Runde Bier für die komplette Gruppe bestraft, kam jedoch (leider) nur noch einmal vor.

Die Verzweigung des Projekts kann in Bitbucket im Projekt unter

https://bitbucket.it.hs-heilbronn.de/projects/LG/repos/labor_games_2016_ss_rpg/branches

abgerufen werden.

Ein anschaulicher Graph zur Verzweigung des Projekts ist als „RevisionGraph.jpg“ im Ordner Diagramme in der Dokumentationsstruktur zu finden.

Der endgültige, stabile Master des Projekts befindet sich unter folgendem Link:

https://bitbucket.it.hs-heilbronn.de/projects/LG/repos/labor_games_2016_ss_rpg/browse

Das Ergebnis

Das endgültige Spiel hatte in etwa den Umfang, den wir anfänglich auch planten. Lediglich ein Epic – nämlich der Online – Arenakampf Modus - konnte nicht implementiert werden. Auch aufgrund zwei „verschlafener“ Sprints mussten im letzten Sprint 2 Releases nacheinander entwickelt werden, weshalb für die Testphase nicht so viel Zeit wie erwartet blieb und wir hier einige Einschränkungen hinnehmen mussten. Durch unsere disziplinierte Selbstkontrolle durch das oben erwähnte Mindestens-2-Prüfer-System und unseren ziemlich frühen Start von Tests mit Gruppenfremden Testern ist die Version auch ohne die geplanten ausführlichen Tests angemessen stabil.

Das Spiel ist sowohl durch die gebaute APK auf Android Smartphones als auch durch die Web-Technologie bedingt über jeden Webserver (zum Beispiel Wampp) gehostet auch am PC spielbar. Durch die Cordova-Struktur kann das Spiel zudem ohne Aufwand mit MacOS auf iOS portiert werden. Ein Windows Build ist dank Cordova/Phonegap ebenso möglich.

Da das endgültige Spiel recht umfangreich ist empfiehlt sich es einmal anzuspielden um einen genauen Eindruck davon zu bekommen. Folgend jedoch ein paar erste Eindrücke des fertigen Spiels.

Alle dargestellten Funktionen und Features sind bereits fertig implementiert und voll funktionsfähig. Wird etwa ein Spiel Laden Button dargestellt, kann davon ausgegangen werden, dass die Funktion des Speicherns und Ladens eines Spielstandes – unseres Wissens nach - funktionstüchtig ist.

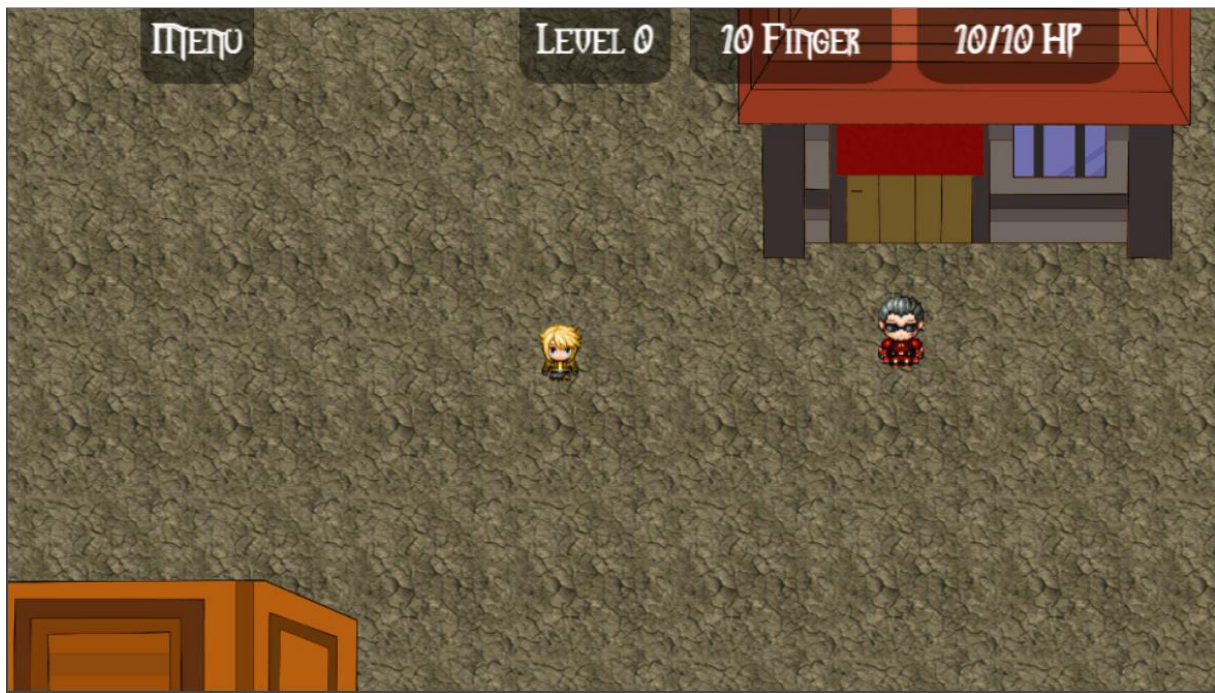
Wir bieten keine Gewähr auf auftretende Fehler/Bugs, die mögliche Features von deren Funktionalität wir ausgehen einschränken oder deren Funktionalität völlig aufheben.



- Der Ladebildschirm mit dem Falltown RPG Logo.



- Das Hauptmenü mit den verschiedenen Auswahlpunkten.



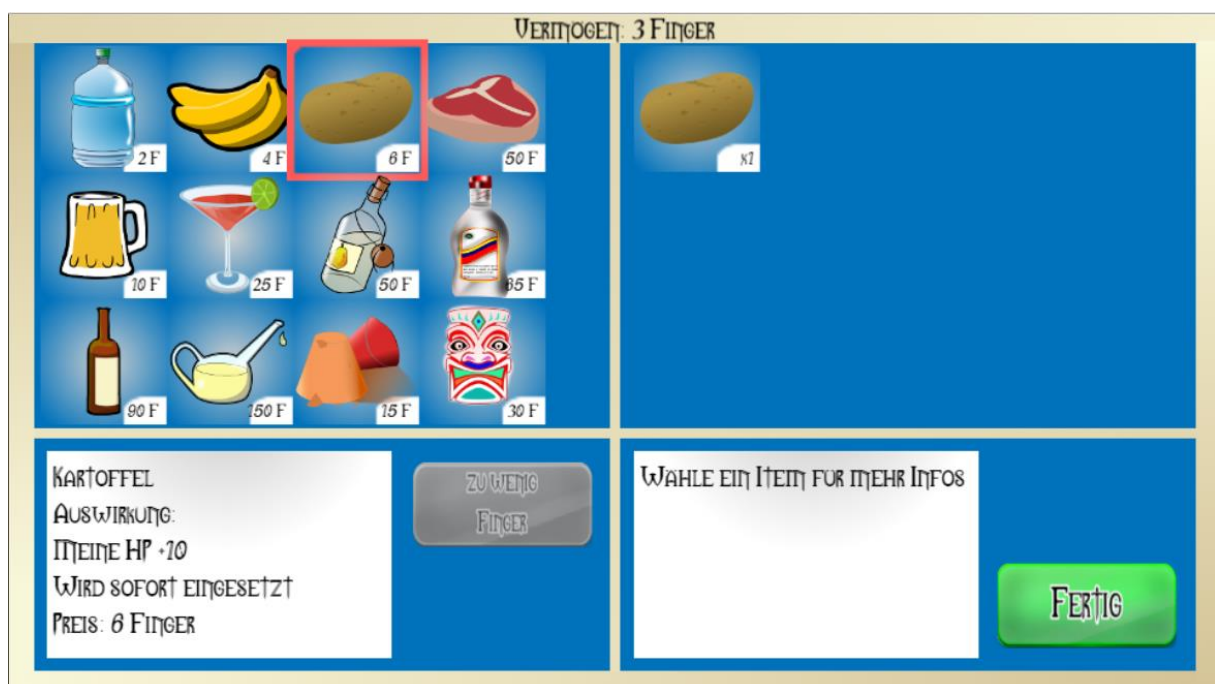
- Das Interface des freien Modus mit derzeitiger Position in Falltown. Zentral ist der Spieler, rechts der Lehrer und das Lehrerhaus.



- Die Kampfansicht und in diesem speziellen Fall das Kampf-Tutorial gegen den Lehrer



- Interaktion mit anderen Dorfbewohnern in Falltown, hier die für die Story relevante Katzendame.



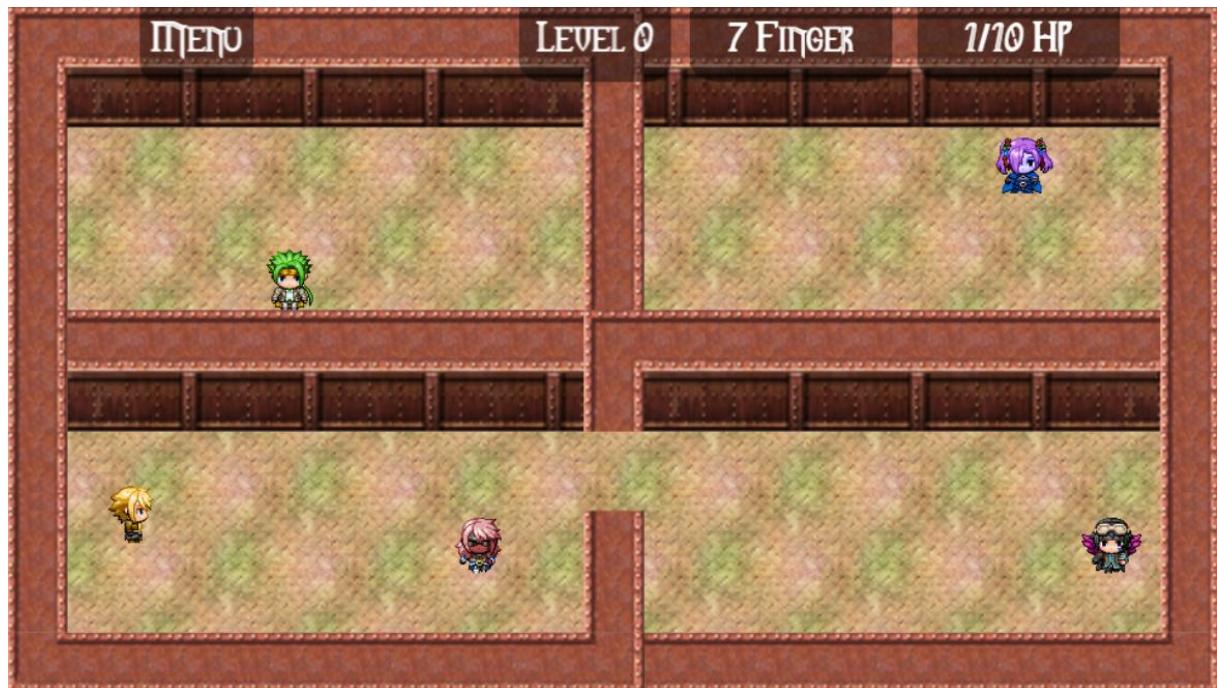
- Das Verkaufsmenü des Nahrungsmittelhändlers. Dieser bietet defensive Gegenstände an die etwa die HP wieder auffüllen, die eigene Präzision erhöhen etc. oder offensive Gegenstände, die dem Gegner über Dauer Schaden anfügen oder die Präzision senken, etc. Neben dem Nahrungsmittelladen gibt es noch den Schmied, der Waffen und Rüstungen verkauft. Links das Inventar des Händlers, rechts das eigene. Alle Gegenstände können gekauft und für einen geringeren Preis wiederverkauft werden.



- Das Afterfall-Gebiet mit verschiedenen Gegnern, die einen Angreifen, sobald man ihnen zu nahekommt



- Kampf gegen einen Gegner. In diesem Fall der erste Bossgegner.



- Das Bossgegner Dungeon nachdem der erste Gegner besiegt wird. Der Spieler kommt erst durch Sieg gegen den letzten Gegner oder Niederlage gegen einen der Gegner wieder aus dem Dungeon. Jede Kammer wird erst geöffnet, wenn der Gegner in der vorherigen besiegt wurde. Die Gegner werden in jeder Kammer stärker.



- Das Skillverteilungsmenü. Nach dem Sieg gegen den letzten der 4 Bossgegner folgt ein kurzer Story Teil und danach kann der Spieler verschiedene Skill Punkte im oben dargestellten Menü verteilen, die direkten Einfluss auf den Handel mit den Händlern oder den Kampf gegen Gegner hat.
- Nach der Skillauswahl startet der Spieler in einem höheren Level wieder in Falltown und kann sich von neuem gegen die Banditen ausrüsten.

Persönlicher Anteil am Projekt der einzelnen Projektmitglieder

Dominik Kress

Ich hatte hauptsächlich die Rolle des Producers, also des Projektleiters. Das heißt der erste Entwurf eines Projektplans sowie jegliche Projektmanagement Aufgaben fielen auf mich. Ich entwarf also die Projektstruktur mit den Arbeitspaketen und pflegte während der Projektlaufzeit die Ticketverwaltung in Jira, hatte also vor allem am Start des Projekts alle entwickelten Storys einzupflegen und während des Projektlaufs anzupassen, sowie zu Beginn jedes Sprints die einzelnen Tasks herunter zu brechen und einzupflegen, war also Reporter aller Tickets.

Außerdem war ich für die Dokumentation zuständig. Vom ersten Projektpitch, über den Projekt-Backlog bis hin zur Abschlussdokumentation lag die Verantwortung hierfür bei mir.

Neben dem Projektmanagement kam die erste Idee für das Projekt von mir, also entwickelte ich auch die Design-Entwürfe um diese anfängliche Idee auch mit dem Input der anderen Projektmitglieder zu verfeinern und an bisher entwickeltes und implementiertes anzupassen. Das heißt auch alle Design-Dokumente stammen von mir.

Als Projektleiter und Game Designer hatte ich also eine besondere Verpflichtung auch die Kommunikation der Teammitglieder zu koordinieren und hierbei immer darauf zu achten Ideen, Kritik und Änderungswünsche der Teammitglieder aufzunehmen und gegebenenfalls umzusetzen, ohne dabei das einheitliche Bild des gesamten Game Designs und die Persistenz der Story, beziehungsweise Spielidee und des Gameplays nicht aus den Augen zu verlieren.

Neben den administrativen Arbeiten half ich auch beim produktiven Entwickeln des Projekts. So stammt beispielsweise die Steuerung des Spielers innerhalb des Prototypen von mir. Zudem war ich für die Rekonstruktion bzw. Repositionierung von Falltown zuständig und hatte die Verantwortung über die persistente Story-Integration. Die Konversationen der zwei freien Falltownbewohner, das Tutorial des Lehrers, der Cutszene Dialog bei Sieg gegen den Endgegner sowie die komplette Katzendamen Sequenz nach dem Kampf stammen auch von mir.

Ansonsten hatte ich wie jeder andere im Projekt auch die Pflichten eines Testers und habe in vielen Pull-Requests einen der Prüfer gestellt und so noch verschiedene andere Bereiche und Funktionalitäten mitgetestet.

Zuletzt hatte ich als Projektleiter neben meiner eigenen Zwischenpräsentation noch die Verantwortung über die Präsentation zum Game-Pitch sowie die Abschlusspräsentation des endgültigen Spiels.

Valentin Petrov

Mein Name ist Valentin Petrov Im Folgenden gehe ich auf meine Aufgaben im Projekt ein.

Grafik Design

Meine Hauptaufgabe in diesem Projekt war das Designen von jeglichen Gegenständen im Spiel. Hierbei habe ich mich als Grundlegendes Werkzeug für die Open Source Anwendungen Gimp und Inkscape entschieden. Der Vorteil der Anwendungen war, dass sie kostenlos und frei zugänglich waren. Inkscape ein Vektor Zeichnungsprogramm zum Zeichnen von Gebäuden und Gegenständen (Assets) genutzt. Der Vorteil bei dem Programm war, man konnte extrem Groß zeichnen und es anschließend auf die gewünschte Größe runterskalieren ohne Qualitätsverlust. Gimp wiederum habe ich für Verbesserungen/ Veränderungen genutzt oder auch wurden einige Texturen mit Gimp erstellt. Sowie der Bunker Boden, habe ich mit mehreren Ebenen gearbeitet. Die Oberste Ebene war der Farbverlauf, darunter kam eine rostige Metall Struktur, so sah es nach altem Metall aus. Zum Schluss noch eine Metall Struktur damit es auch wirklich nach Metall aussah. Für die Spieler und NPC habe ich aus Zeitgründen mich für ein Programm entschieden, dass die Sprites für jeden Spieler generieren konnte. So war der größte Teil der Arbeit, der für die Charakter design gering gehalten wurden.

Testing:

Wie jeder in unsrem Team habe ich immer zum einem Pullrequest das Spiel auf Fehler und Unstimmigkeiten getestet. Dabei habe ich darauf geachtet, welche Veränderungen wurden gemacht und was bei den Veränderungen schief gehen konnte. Es sollten auch schon lauffähige Implementierungen weiter hin lauffähig bleiben. Bei jedem Pullrequest wurde das Spiel von Anfang an versucht jede mögliche Kombination durch zuspieren. Sofern es die Zeit zugelassen hatte.

Fazit:

Das Team war in seiner Zusammenstellung, sehr gut aufeinander abgestimmt. Und es hat deswegen auch viele Spaß gemacht so ein umfangreiches Spiel zu entwickeln. Wenn es nochmal so eine Möglichkeit gäbe hätte ich die Jira Tickets anders benannt um sie besser voneinander zu unterscheiden. Wichtig auch hier die Zeit des Sprints einhalten.

Tom Görner

Mein Name ist Tom Görner und hier beschreibe ich meine Aufgaben:

Programmieren:

In einem früheren Stand hatten wir in Falltown Dorfbewohner die sich zufällig durch die Landschaft bewegen sollten. Dafür habe ich einen einfachen Algorithmus geschrieben der aber nicht verwendet wurde, da die Dorfbewohner rausgenommen wurden, stattdessen wird diese Logik jetzt bei den Gegnern in Afterfall benutzt. Für die Gegner war eine leichte Modifikation des Algorithmuses erforderlich damit diese wenn man in die Nähe dieser Gegner kommt. Später brauchte dieser Algorithmus eine Modifikation da sich die Gegner an Gebäudeecken verhakt hatten.

Bugfixing

Brian hatte ein Plugin eingebaut, was einen Screenshake-Effekt zur Verfügung stellt. Dieses Plugin wurde im Kampfmodus benutzt wenn man selber getroffen wurde. Wenn man im Dungeon kämpft, hat der Effekt aus am Anfang unerfindlichen Gründen nicht funktioniert. Nach einigem Herumprobieren bin ich dann darauf gekommen dass dies mit der Weltengröße zusammenhängt und der Shake nur funktioniert wenn die Welt größer als der sichtbare Bereich ist.

Gegen anfang als ich das Spritesheet der Stadtmauer einbauen wollte hat es nicht wollen obwohl die Größe dieses Spritesheets den Anforderungen von Phaser (beide Kantenlängen eine zweierpotenz) entsprochen hat. Dieser Bug ließ sich recht schnell auf einen Phaserbug zurückführen da das Problem nur unter WebGL aufgetreten ist. Aktualisieren auf eine neue Phaser-Version hat den Fehler behoben.

Brian Wirth

Theophil Bachmann