****

**Anforderungsanalyse**

Retro-Indie-Spiel allgemein

**Inhaltsverzeichnis**

[1 Allgemeines 2](#_Toc435453407)

[1.1 Hintergrund 2](#_Toc435453408)

[1.2 Zweck und Ziel dieses Dokumentes 2](#_Toc435453409)

[1.3 Aufteilung der Teams <TODO> 2](#_Toc435453410)

[1.4 Abkürzungen 3](#_Toc435453411)

[2 Ziel und Rahmenbedingungen 4](#_Toc435453412)

[2.1 Ziel des Projekts 4](#_Toc435453413)

[2.1.1 Ist-Zustand 4](#_Toc435453414)

[2.1.2 Soll-Zustand 4](#_Toc435453415)

[2.2 Benutzer / Zielgruppe 4](#_Toc435453416)

[2.3 Systemlandschaft <TODO> 4](#_Toc435453417)

[2.3.1 Software 4](#_Toc435453418)

[2.3.2 Hardware 4](#_Toc435453419)

[3 Beschreibungen und Anforderungen 5](#_Toc435453420)

[3.1 Anforderung: Spielobjekte 5](#_Toc435453421)

[3.2 Anforderung: Spielerfigur 6](#_Toc435453422)

[3.3 Anforderung: Datenbankanbindung 7](#_Toc435453423)

[3.4 Anforderung: Leveldesign 8](#_Toc435453424)

[3.5 Anforderung: Grafiken 9](#_Toc435453425)

[3.6 Anforderung: Sound 10](#_Toc435453426)

[3.7 Anforderung: Systemvoraussetzung 11](#_Toc435453427)

[3.8 Anforderung: Kommerzielle Nutzung 12](#_Toc435453428)

[3.9 Anforderung: Portierbarkeit Android 13](#_Toc435453429)

[4 Historie des Dokuments 14](#_Toc435453430)

1. Allgemeines
   1. Hintergrund

Die Idee für diese Software ist im Rahmen einer Projektarbeit für das Fach „Softwareengeneering“ entstanden. Es gilt exemplarisch einen kompletten Zyklus der Softwareentwicklung zu durchleben.

Das Projekt beschäftigt sich mit der Entwicklung eines 2D-Spiels. Hierzu arbeiten 2 Teams an dem Projekt. Team 1 beschäftigt sich mit der Entwicklung des 2D-Spiels mit allen Funktionalitäten. Team 2 beschäftigt sich mit einer Datenbankanbindung. Zwischen Team 1 und Team 2 wird es eine Schnittstelle geben. Die Kommunikation beider Teilsoftwares geschieht durch URL aufrufe.

* 1. Zweck und Ziel dieses Dokumentes

In diesem Dokument wird es eine grobe Beschreibung der Software und deren Bestanteile geben. Jede Anforderung wird dann weiter in feinerer Analysen Unterteilt. Diese Analysen erfolgen in gesonderten Dokumenten. Der Leser soll ein grobes Verständnis für die Software bekommen. Hierbei wird Bezug auf funktionale und nicht funktionale Eigenschaften der Software genommen. Außerdem wird hier schon eine grobe Einteilung der Arbeitskräfte getätigt. Dieses Dokument ist hauptsächlich für die Arbeiten von Team 1 gedacht. Team 2 erstellt ein gesondertes Dokument. Es gibt nur eine Anforderung, die die Schnittstelle zu Team 2 herstellt.

* 1. Aufteilung der Teams <TODO>

|  |  |  |
| --- | --- | --- |
| **Team 1** | **Teamleiter** | Tino Blau |
| **Programmierer** | Juraj Graser |
| Dan Schieck |
| Tino Blau |
| **Grafiker** | Maria Kömmpel |
| **Dokumentation** | Katharina Wiegel |
| **Tester** | Katharina Wiegel |
| Maria Kömmpel |
| **Team 2** | **Teamleiter** |  |
|  |  |
|  |  |
|  |  |

* 1. Abkürzungen

TODO: Abkürzungen einfügen

1. Ziel und Rahmenbedingungen
   1. Ziel des Projekts
      1. Ist-Zustand

Zu Beginn des Projektes gibt es weder Vorkenntnisse in der Spieleprogrammierung oder Spieleanimation, noch eine Basis für ein 2D Spiel. Die Grundlagen für das Projekt sind die verschiedenen Tutorials und Dokumentation zur GameMaker Studio Benutzung. Alle Entwicklungen müssen von Grund auf erstellt werden.

* + 1. Soll-Zustand

Am Ende soll ein funktionierendes Spiel entstehen. Dabei wird hier auf eine Betaversion hingearbeitet. Anstatt, dass das Spiel komplett und vollständig sein soll, soll es demoartig aufgebaut werden. So soll ersichtlich werden, was das Spiel alles kann. Dabei werden die Funktionalitäten der einzelnen Spielelemente herausgehoben. Danach soll erkennbar sein, wohin die Spielidee führt. Die Software muss derart erweiterbar sein, dass selbst nach Terminende ein fertiges Produkt entstehen kann, welche allen Anforderungen gerecht wird.

* 1. Benutzer / Zielgruppe

Jeder mit einem funktionierenden Windowscomputer soll diese Software nutzen können. Öffentliche Spiele benötigen des Weiteren eine Altersfreigabe. Aus Erfahrungswerten kann man sagen, dass durch die düstere Grundstimmung das Spiel ab 12 Jahre sein wird. Natürlich ist dies nur eine Empfehlung des Projektteams. Ansonsten ist dieses Spiel für jeden Gedacht, der gern ein 2D-Spielen will.

* 1. Systemlandschaft <TODO>
     1. Software

Entwicklungsumgebung: GameMaker Studio: Pro

Grafiksoftware: Photoshop

UML: Enterprise Architekt

SQL:

PHP:

* + 1. Hardware

Zur Entwicklung wird der Arbeitslaptop, welcher von der Telekom zur Verfügung gestellt wird, verwendet. Dieser muss für die Entwicklungsumgebung verwendet werden, da ansonsten die Anforderung zur Systemvoraussetzung nicht getestet werden kann, bzw. nicht gewährleistet wird.

1. Beschreibungen und Anforderungen

Die Software besitzt verschiedene Anforderungen, die es zu beschreiben gilt. Jede Anforderung wird hier nur grob beschrieben. Eine detaillierte Beschreibung findet später statt. Es gibt zwei Kategorien von Anforderungen. Zum einen sind es die funktionalen Anforderungen, welche den Großteil ausmachen und zum anderen sind es die nicht funktionalen Anforderungen.

* 1. Anforderung: Spielobjekte

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Spielobjekte | **ID** | ANF\_RIS\_SO |
| **Wichtigkeit** | Pflicht |
| **Zuständigkeit** | GML Programmierer |
| **Weiterführende Dokumente** | <TODO Dokumente zu Spielobjektanforderung angeben> | | |

Beschreibung

Für ein Funktionierendes Spiel benötigt man Zahlreiche Spielobjekte. Zu den Spielobjekten gehören neben der Spielfigur und Gegner auch Dekorationselemente, wie Tische, Fenster oder Schränke. Die Spielfigur und die Gegner besitzen hier den größten Aufwand, was die Programmierung angeht. Statische Elemente benötigen nur zum Teil eine Programmierung, falls der Spieler mit diesen Objekten interagieren kann. Es muss also analysiert werden, welche Objekte welche Bedeutung im Spiel bekommen werden. Die Spielfigur wird hierbei in einer eigenen Anforderung betrachtet. Eine Auflistung aller Spielobjekte findet in einem gesonderten Dokument statt.

Wechselwirkung

Die meisten Spielobjekte benötigen auch Grafiken. Außerdem haben die Spielobjekte Auswirkungen auf das Leveldesign. Ein Level besteht aus mehrere Spielobjekte. Manche Spielobjekte benötigen weiterhin Sound um eine Atmosphäre zu erzeugen. Es gibt also große Verknüpfungen mit **ANF\_RIS\_G**, **ANF\_RIS\_S**, **ANF\_RIS\_SF** und **ANF\_RIS\_LD**.

Risiken

Die enge Verknüpfung mit **ANF\_RIS\_G**, **ANF\_RIS\_S**, **ANF\_RIS\_SF** und **ANF\_RIS\_LD** bedarf große Vorsicht. Änderungen innerhalb einer Anforderung haben Auswirkungen auf die anderen Anforderungen. Wird während der Implementation festgestellt, dass weitere Objekte benötigt werden, dann muss dies mit den anderen Teammitgliedern kommuniziert werden. Des Weiteren sollte zumindest eine grobe Analyse, welche Spielobjekte es geben wird, geben, damit mit den Grafiken rechtzeitig begonnen werden kann. Gerade die kurze Entwicklungszeit für zu Zeitdruck.

Grobschätzung des Aufwands

Der Aufwand wird auf 1,5 Monate geschätzt.

* 1. Anforderung: Spielerfigur

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Spielerfigur | **ID** | ANF\_RIS\_SF |
| **Wichtigkeit** | Pflicht |
| **Zuständigkeit** | Tino Blau |
| **Weiterführend** | <TODO Dokumente zu Spielobjektanforderung angeben> | | |

Beschreibung <TODO erweitern, was die Figur kann>

Die Spielfigur ist der zentrale Bestandteil des Spiels. Der Nutzer steuert ausschließlich diese Figur. Hier werden die Zentrale Funktionalitäten festgelegt. Dazu gehört neben den Bewegungen auch ein Großteil der Interaktionen mit den anderen Spielobjekten. Eine Spielfigur kann sich bewegen, springen, Gegenstände aufnehmen, sterben und viele weitere Dinge. Die Funktionalitäten können als Zustandsmaschine definiert werden.

Wechselwirkung

Die Spielfigur benötigt Animationen. So müssen alle Zustände von der Spielfigur bekannt sein, damit die Grafiken erstellt werden können (**ANF\_RIS\_G**). Für jeden Zustand kann es verschiedene Sounds geben. So könnte es ein z.B. Laufgeräusch geben (**ANF\_RIS\_S**). Des Weiteren führt die Figur Interaktionen mit den anderen Spielobjekten aus (**ANF\_RIS\_SO**).

Die Wechselwirkungen wirken sich nicht so stark aus. Neue Sounds und Grafiken müssen zwar ggf. erstellt werden, wenn die Figur ihre Funktionalitäten erweitert, aber die Konsequenzen sollten sich im Rahmen halten.

Risiken

Die Figur ist der zentrale Bestandteil und steht im Mittelpunkt des Spiels. Da die Figur mit den Spielobjekten interagiert, sollte die Figur in Zuständen programmiert werden. So können Interaktionen mit dazukommenden Objekten gewährleistet werden. Fehlerhafte Programmierung der Spielerfigur kann zum Absturz des Programmes führen. Somit muss hier mit besonderer Sorgfalt programmiert werden.

Grobschätzung des Aufwands

Da die Spielfigur den größten Teil der Programmierung ausmacht, wird der Aufwand auf 2 Monate geschätzt.

* 1. Anforderung: Datenbankanbindung

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Datenbankanbindung | **ID** | ANF\_RIS\_DBA |
| **Wichtigkeit** | Pflicht |
| **Zuständigkeit** | Team 2 |
| **Weiterführend** | <TODO Dokumente zu Spielobjektanforderung angeben> | | |

Beschreibung <TODO Alle Anforderungen aus dem untergeordneten Dokument hier grob einfügen>

Die Datenbankanbindung ist hier ein kleines Nebenprojekt. Es soll eine Schnittstelle zwischen dem RIS und einer Datenbank geben. Die Datenbankanbindung regelt die Registrierung und Anmeldungsroutine. So kann sich ein Nutzer online einloggen. Hierbei ist zu beachten, dass der Server, worauf die Datenbank läuft, simuliert werden kann. Es muss nicht extra ein Webserver gemietet werden. Einige Funktionalitäten sollen erweiterbar sein. So soll es möglich sein, dass sich mehrere Nutzer gleichzeitig einloggen oder registrieren können. Des Weiteren soll sich Gedanken über die Datensicherheit und Verschlüsselungen gemacht werden. Gerade Passwörter sind sensible Daten und müssen geschützt werden. Neben der Datenbank und evt. PHP Programmierung, muss es eine vollständige Dokumentation geben.

Wechselwirkung

Damit diese Anforderung weitestgehend unabhängig ist, wurde auf große Wechselwirkungen verzichtet. Es gibt Abhängigkeiten zu einigen Spielobjekten, hauptsächlich Menüobjekten, um eine Registrierung oder Anmeldung anzustoßen.

Risiken

Eine Risikoanalyse wird von Team 2 vorgenommen.

Grobschätzung des Aufwands

Da es ein eigenständiges Projekt ist, wird der Aufwand auf 2 Monate geschätzt.

* 1. Anforderung: Leveldesign

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Leveldesign | **ID** | ANF\_RIS\_LD |
| **Wichtigkeit** | Pflicht |
| **Zuständigkeit** | GML Programmierer |
| **Weiterführend** | <TODO Dokumente zu Spielobjektanforderung angeben> | | |

Beschreibung

Sobald alle Spielobjekte erstellt wurden, können die einzelnen Levels erstellt werden. Jeder Level wird dabei sich von anderen Level unterscheiden. Es muss darauf geachtet werden, dass die einzelnen Levels bestimmte Schwierigkeitskategorien angepasst werden. So soll der Spieler mit einem Tutorial beginnen und mit den einzelnen Spielelementen und Objekten vertraut gemacht werden. Nach einem kleinen Tutorial gibt es Einsteigerlevel, wo der Spieler das gelernte Anwenden muss. Ein Einsteigerlevel bedient sich nur an einzelne Spielelemente. Danach steigt der Schwierigkeitsgrad langsam an und die Spielelemente werden miteinander kombiniert. Der Spieler soll weiterhin nicht bemerken, dass die Level Kategorisiert sind, sprich die Übergänge müssen fließend sein.

Wechselwirkung

Jeder Level besteht aus den Spielobjekten. Außerdem benötigen wir in jedem Level eine Spielfigur. Die Definitionen, die somit in **ANF\_RIS\_SO** getroffen wurden, haben direkte Auswirkungen auf das Leveldesign.

Risiken

Das größte Problem ist, dass die Levels erst dann erstellt werden können, wenn ein Großteil der Spielobjekte programmiert sind. Somit kann es zu zeitlichen Problemen kommen. Aus diesem Grund werden vorerst nur Demolevel erstellt und kein vollständiges Spiel. Betrachtet man, dass dieses Spiel im fertigen Zustand rund 2 Stunden dauern soll, so werden an die 50 Levels benötigt. Dies ist zeitlich nicht zu stemmen. Die Demolevels zeigen das grobe Konzept der Level und was mit den Spielobjekten möglich ist.

Grobschätzung des Aufwands

Da wir nur einige Demolevel erstellen, kann der Aufwand gekürzt werden. Da man erst Level erstellen kann, nachdem die Spielobjekte erstellt wurden, wird der Aufwand auf 2 Wochen geschätzt.

* 1. Anforderung: Grafiken

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Grafiken | **ID** | ANF\_RIS\_G |
| **Wichtigkeit** | Sollte |
| **Zuständigkeit** | Maria Kömmpel |
| **Weiterführend** | <TODO Dokumente zu Spielobjektanforderung angeben> | | |

Beschreibung

Damit das Spiel auch eine gewisse Atmosphäre erzeugt, werden Grafiken benötigt. Dabei gibt es statische Bilder, als auch animierte Bilder. Jede Grafik wird selbst per Hand gezeichnet. Die Grundstimmung ist eine düstere Atmosphäre. Somit sind alle Bilder etwas düster. Die animierten Bilder sind vor allem für die Spielerfigur gedacht. Dadurch erzeugt man die Illusion einer Bewegung. Eine Grafik wird dann einfach mit einem Spielobjekt verknüpft. Den Rest erledigt die Entwicklungsumgebung. Außerdem werden Hintergründe benötigt. Es ist hierbei wichtig, dass es eine Unterscheidung von Hintergründe und Spielobjekte gibt. Spielobjekte sind alle Dekorationselemente, während ein Hintergrund eine Art „Mauer“ ist. Dadurch wird gewährleistet, dass jeder Level später variiert erstellt werden kann, ohne neue Grafiken zeichnen zu müssen. Die Grafiken werden also modularisiert. Durch die händische Zeichnungen soll das Gesamtbild gemäldeartig werden.

Wechselwirkung

Die Grafiken sind eng mit den Spielobjekten und der Spielfigur verknüpft. Es müssen erst alle Funktionalitäten der Spielfigur definiert werden, damit die passenden Animationen dazu gestaltet werden können (**ANF\_RIS\_SF**). Außerdem sollten weitestgehend alle Spielobjekte vorhanden sein (**ANF\_RIS\_SO**).

Risiken

Die Grafiken dürften den größten Zeitanteil ausmachen. Jede Grafik wird per Hand erstellt. Dadurch muss hier besonders auf die Zeit geachtet werden. Da es nur einen Zeichner geben wird, um eine kontinuierlichen Stil zu bekommen, muss am Ende nicht für jedes Objekt eine Grafik geben. Es gibt Grafiken, die sollten bis zum Projektende erstellt wurden sein und Grafiken, deren Relevanz nur niedrig sind. Grafiken mit geringer Priorität können durch Platzhalter ersetzt werden.

Grobschätzung des Aufwands

Die Grafiken haben mit den größten Aufwand. Der Aufwand wird mit 2,5 Monaten geschätzt.

* 1. Anforderung: Sound

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Sound | **ID** | ANF\_RIS\_S |
| **Wichtigkeit** | Kann |
| **Zuständigkeit** | offen |
| **Weiterführend** | <TODO Dokumente zu Spielobjektanforderung angeben> | | |

Beschreibung

Neben den Grafiken ist auch der Sound entscheidend für die Atmosphäre. So soll es Geräusche geben, die vom Spieler kommen. Der Spieler macht Laufgeräusche oder Treffergeräusche. Wenn der Spieler einen Gegenstand aufnimmt, dann soll dies mit einem Geräusch untermalt werden. Auch soll der Spieler ein Geräusch hören, falls sich Türen öffnen. Die Gegner im Spiel können ebenso Töne von sich geben. Es könnte auch eine Hintergrundmelodie geben.

Wechselwirkungen

Die Geräusche sind abhängig von den Spielobjekten (**ANF\_RIS\_SO**). Erst wenn die Objekte und ihre Bedeutung, sowie das Aussehen bekannt sind, können die Geräusche dazu entwickelt werden. Der Sound ist somit das letzte was entsteht. Wichtig hierbei ist die Betrachtung mit ANF\_RIS\_KN, welche die Kommerzielle Nutzung vorsieht. Dadurch können nicht einfach fertige Soundfiles aus dem Internet genommen werden. Ideal wäre eine eigenständige Entwicklung.

Risiken

Aktuell gibt es keine Person im Projektteam, die sich mit Sound auskennt. Dadurch werden vorerst minimalistische Sounds erzeugt. Außerdem werden nicht alle Soundeffekte bis zum Ende eingefügt. Somit ist die Priorität nur sehr niedrig und dient nur zum Feinschliff. Schlechter Sound kann mehr an der Atmosphäre zerstören, als sie erzeugt. Deswegen muss hier vorsichtig gearbeitet werden.

Geschätzter Aufwand

Da hier nur minimalistische Sounds erstellt werden, wird der Aufwand mit 1 Woche geschätzt.

* 1. Anforderung: Systemvoraussetzung

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Systemvoraussetzung | **ID** | ANF\_RIS\_SV |
| **Wichtigkeit** | Sollte |
| **Zuständigkeit** | GML Programmierer |

Beschreibung

Das Spiel soll selbst bei schwachen Rechnern mit 60 FPS laufen. Somit müssen die Programmierer auf ihren Quellcode achten. Als Faustregel sollte sich jeder Programmierer merken, dass eine bewegte Szene 60-mal pro Sekunde berechnet wird. Der Quellcode wird also 60-mal ausgeführt. Zur Optimierung sollten somit einige Events außerhalb des Step Evens programmiert werden. Eine Optimierungsmöglichkeit wäre z.B., dass bestimmte Aktionen erst bei einer aktiven Kollision berechnet werden. Da wir keine Umfangreichen Systemtests ausführen können, wird der Arbeitslaptop von der Telekom al Referenz genommen.

CPU: Dual-Core 2,7GHz

RAM: 4GB

Grafikkarte: Intel HD Graphics 4000

Festplattenspeicher: max. 1GB

Steuerung: Tastatur oder Xbox360 Controller

Risiken

Hier muss besonders auf den Quellcode geachtet werden. Zu viele Schleifen können das Spiel verlangsamen. Eine konstante FPS ist hier sehr wichtig. Der Quellcode wird 60-mal pro Sekunde ausgeführt. Zu lange Schleifen potenzieren die Berechnungszeit. Des Weiteren dürfen die Bilder nicht zu groß werden, da sonst der RAM nicht ausreicht.

* 1. Anforderung: Kommerzielle Nutzung

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Kommerzielle Nutzung | **ID** | ANF\_RIS\_KN |
| **Wichtigkeit** | Optional |
| **Zuständigkeit** | Alle |

Beschreibung

Die Kommerzielle Nutzung ist eine reine optionale Anforderung. Die Auswirkungen dieser Anforderung auf das Projekt sind, dass darauf verzichtet werden muss, Inhalte wie Grafiken, Quellcode oder Sound von Dritten zu nutzen. Dies ist zumindest bei Quellcode selbstverständlich. Bei Grafiken und Sound sieht die Sache aber anders aus. Hier nutzen viele Hobbyprogrammierer Bibliotheken aus dem Internet. Dort gibt es eine Vielzahl an kostenlosen Grafiken oder Soundfiles für Hobbyprogrammiere. Für eine private Nutzung besteht hier keinerlei Gefahr. Die meisten Anbieter verbieten aber eine kommerzielle Nutzung ihrer Angebote.

Das Spiel soll zwar nicht kommerziell vermarktet werden, doch soll somit eine reine Eigenleistung gewährleistet werden. Wenn die Urheberrechte aller Grafiken und Sounds beim Team liegen, dann könnten diese auch kommerziell genutzt werden.

Ein weiterer wichtiger Punkt für die kommerzielle Nutzung ist die Entwicklungsumgebung. Hier wurde vor dem Projektstart gewährleistet, dass alle Entwicklungen mit dem GameMaker: Pro dieser Anforderung entsprechen. Die Pro Version ist direkt für kommerzielle Entwickler entworfen.

Risiken

Gerade beim Sound kann es zu Konflikten kommen. Das Team besitzt keine Person, welche sich mit dem Erstellen von Sounds auskennt. So ist die Überlegung groß, auf diese Anforderung zu verzichten und dafür gute Sounds für die Atmosphäre aus den Internetbibliotheken zu nehmen.

* 1. Anforderung: Portierbarkeit Android

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Portierbarkeit Android | **ID** | ANF\_RIS\_PA |
| **Wichtigkeit** | Optional |
| **Zuständigkeit** | Tino Blau |

Beschreibung

Das Spiel wird primär für Windows programmiert. Trotzdem soll die Software auch auf Android Systeme portiert werden können. Hier sollen die Grundlagen dafür geschaffen werden. Die Entwicklungsumgebung erleichtert dabei die Portierung, da die Erstellung der APK durch die Angabe eines Target Systems unterstützt wird. Die Entwicklungsumgebung erstellt dann automatisch die APK, welche auf Android Systemen installiert werden kann. Die Portierung bleibt wegen der geringen Zeit für das Projekt optional

Risiken

Die wichtigsten Punkte sind zum einen die Bildschirmauflösung und zum anderen die Steuerung. Der Bildschirm hat zwar eine übliche 16:9 Auflösung, doch die die Gesamtfläche könnte die Inhalte unkenntlich machen. Hinzu kommt die Steuerung. Auf einem Windowsrechner wird die Tastatur oder der Xbox 360 Kontroller verwendet. Bei einem Android System müsste die Steuerung weitestgehend auf den Bildschirm angezeigt werden. Das vergrößert nochmals die Bildschirmproblematik.

1. Historie des Dokuments

|  |  |  |
| --- | --- | --- |
| Version | Datum | Bemerkung |
| 0.1 | 01.10.15 | Erstellung des Dokuments |
| 0.2 | 16.11.15 | Dokument erweitert, formatiert und umbenannt |
|  |  |  |