

Blog Post Summary

10. Juni 2017

TEXTVENTURER

André Schmitt, Dominik Vogel, Simon Vollmer

Inhalt

[PROJECT START 3](#_Toc484879237)

[OUR TEAM 3](#_Toc484879238)

[SOFTWARE REQUIREMENTS SPECIFICATION 5](#_Toc484879239)

[USE CASE DIAGRAMM 7](#_Toc484879240)

[YOUTRACK 7](#_Toc484879241)

[TESTING SOFTWARE 7](#_Toc484879242)

[CLASS-DIAGRAM (CRC) 8](#_Toc484879243)

[SOFTWARE ARCHITECTURE 9](#_Toc484879244)

[GANTT-CHART 9](#_Toc484879245)

[MIDTERM SUMMARY 10](#_Toc484879246)

[RISK MANAGEMENT 11](#_Toc484879247)

[FUNCTION POINT CALCULATION 11](#_Toc484879249)

[TEST PLAN 13](#_Toc484879250)

[REFACTORING 13](#_Toc484879251)

[METRICS 14](#_Toc484879252)

# PROJECT START

TextVenturer is, as the name implies, a Text-based Adventure.

We want to bring the retro game style of text adventures back to the present.

It will be possible to enter different scenarios, which you can discover alone or with friends.

These scenarios are saved as userfriendly readable scripts and we might add an even more easy-to-use editor for it, if we have enough time.

# OUR TEAM

Like we said before our team consists of three Students. Dominik Vogel, Simon Vollmer and André Schmitt.

AREAS OF RESPONSIBILITY:

|  |  |
| --- | --- |
| **Function** | **Name** |
| Implementation | Dominik Vogel, André Schmitt |
| Design | Simon Vollmer, André Schmitt, Dominik Vogel |
| Configuration Manager | André Schmitt |
| Tester | Simon Vollmer, André Schmitt, Dominik Vogel |
| Test Manager | Simon Vollmer |
| Project Management | Simon Vollmer, André Schmitt |

We are going to use the Programming Language C++ to write our program.

First we want to implement it as a Windows Desktop Application and if we have enough time we will get a website with our game running.

For an IDE we are going to use Visual Studio including a GitHub extension.

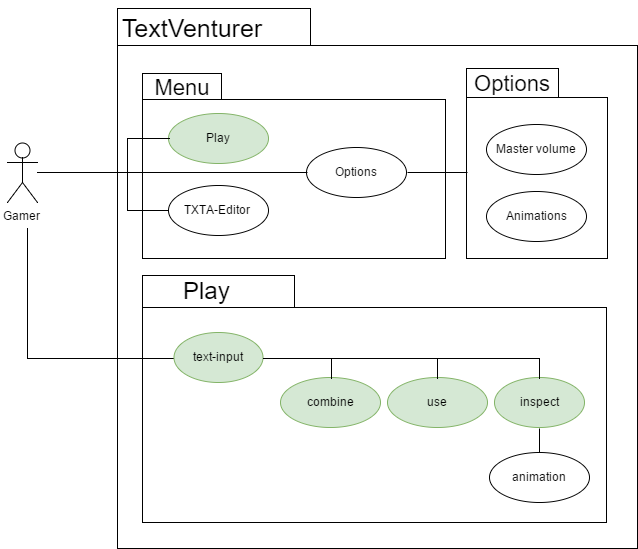
# SOFTWARE REQUIREMENTS SPECIFICATION

We want to keep you up to date.

Here is our newest version of our Software-Requirements-Specification.

You can look at it with the following link:  
<https://github.com/SchmittAndre/TextVenturer/blob/master/Software-Requirements-Specification.pdf>

Also, here is a UseCase-Diagram of the project:

[](https://github.com/SchmittAndre/TextVenturer/blob/master/Use-Case-Diagramm.png)

# USE CASE DIAGRAMM

As you know for a Text-Adventure you need some rudimentary things like describing a [room](https://github.com/SchmittAndre/TextVenturer/blob/master/PAP/Enter_room.pdf), [combining things](https://github.com/SchmittAndre/TextVenturer/blob/master/PAP/combine.pdf), [use an Item](https://github.com/SchmittAndre/TextVenturer/blob/master/PAP/use.pdf), [pick up a Item](https://github.com/SchmittAndre/TextVenturer/blob/master/PAP/pick_up.pdf) or listing your current [inventory](https://github.com/SchmittAndre/TextVenturer/blob/master/PAP/get_inventory.pdf). So here are our Use Case Diagramms for these actions.

# YOUTRACK

If you want to stay up to date about our TextVenturer Project you can follow us on  ~~Jira~~. There you can see our status on what we have accomplished and all our open tasks.

edit:

Since we got problems with Jira we switched over to YouTrack. You can visit it [here](https://textventurer.myjetbrains.com/youtrack/dashboard?id=5d0dda45-ce23-4c54-a2b0-165f09cde144).

# TESTING SOFTWARE

Hey fans,

we tried to find a tool to auto test the game. After hours of searching our pro programmer Dominik Vogel just started to program a self-made testing tool.

And it works!

So now we will be even faster in finishing our project and u will soon be able to test our Alpha.

If you want to see his code you can see it here:

<https://github.com/SchmittAndre/TextVenturer/tree/master/InputSiumlator>

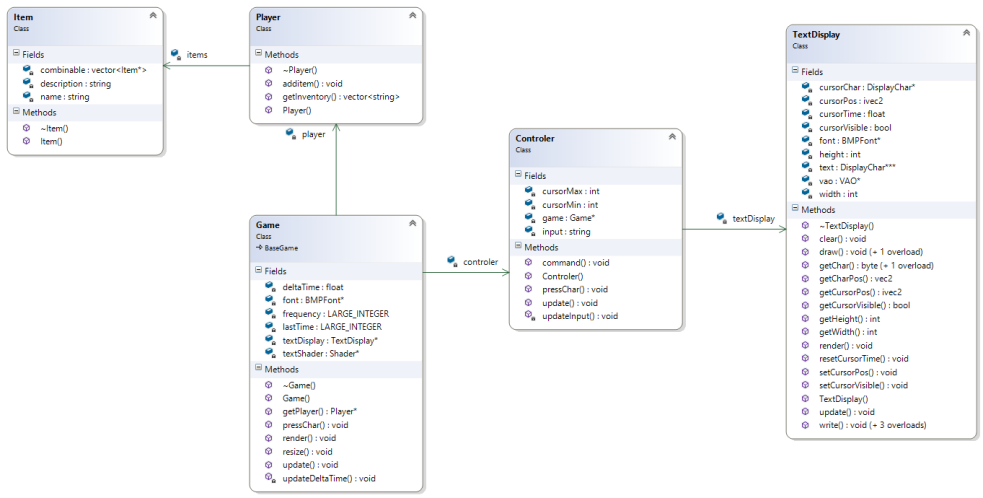
Greetings TextVenturer

# CLASS-DIAGRAM (CRC)

Hey fans,

we finally got our Class-Diagram. Since we created it with VisualStudio it went without any major problems.

Now you can see how our classes work together in this awesome looking diagram.

[](https://textventurer.files.wordpress.com/2016/11/classdiagram.png)

Greetings

TextVenturer-Team

# SOFTWARE ARCHITECTURE

Hey Guys,

today we want to present you our Software architecture.

You may see it [here](https://github.com/SchmittAndre/TextVenturer/blob/master/Softwarearchitecture.pdf).

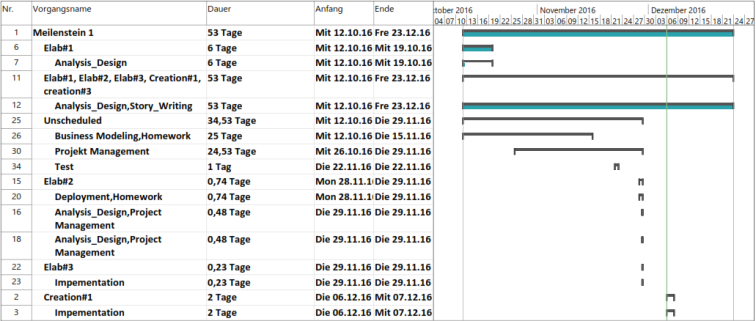
Greetings TextVenturer

# GANTT-CHART

Hey Guys,

today we want to show you our Gantt Diagram.

Big thanks to Project VNV. Without your parser we never would’ve gotten this chart to work.

[](https://textventurer.files.wordpress.com/2016/12/gant-diagramm.png)Gantt-Chart 6.12.16

Greetings TextVenturer

# MIDTERM SUMMARY

Hey Fans,

we are already halftime through and we want to show you a quick summary of what we have done so far.

Project Vision: <https://textventurer.wordpress.com/2016/10/12/project-start/>

GitHub Repo: <https://github.com/SchmittAndre/TextVenturer>

Project Management: <https://textventurer.wordpress.com/2016/10/18/our-team/>

Use Cases: <https://textventurer.wordpress.com/2016/10/30/use-case-diagramm/>

Software Requirement Specifications: <https://github.com/SchmittAndre/TextVenturer/blob/master/Software-Requirements-Specification.pdf>

Test Cases: <https://github.com/SchmittAndre/TextVenturer/tree/master/InputSiumlator/scripts>

Our Testing Tool: <https://textventurer.wordpress.com/2016/11/15/testing-software/>

Gannt Chart: <https://textventurer.wordpress.com/2016/12/06/gantt-chart/>

Burndown-Diagramms: [YouTrack](https://textventurer.myjetbrains.com/youtrack/dashboard?id=5d0dda45-ce23-4c54-a2b0-165f09cde144)

Demo download: <https://github.com/SchmittAndre/TextVenturer/releases>

Code View:  <https://github.com/SchmittAndre/TextVenturer/tree/master/SoftwareEngineering>

Software Architecture: <https://github.com/SchmittAndre/TextVenturer/blob/master/Softwarearchitecture.docx>

Environmental Setup: We used VisualStudio as our IDE for our C++ programm. For visualising we used OpenGL and our Testing Tool is written in Lazarus.

Automated testing: Coming soon!

Presentation: Coming soon!

# RISK MANAGEMENT

Because our Projekt and Teamwork pose various risks, we had to analyse and rank those risks.  
Therefore, we compiled a list of our top 5 risks:

# 

# FUNCTION POINT CALCULATION

Today we’re going to show you how we tried to predict the future using Function-Points. We want to know, how long it would probably take to implement a new feature into our Game. To do this, we rated all our old modules and calculated the so called Function-Points for it using this website: [TINY TOOLS](http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/harvey/FP_Calc.html)

With the results, we can then draw an averaged line, so we can simply read off the required time for new modules. Obviously this is only an estimation, and there might be a big margin, but it still gives a rough idea of how much time a specific module will take up.

[Function Points Calculation](https://github.com/SchmittAndre/TextVenturer/blob/master/FP-Prediction.xlsx)

# TEST PLAN

For a software, to be bug-free, is very important. Therefore we concluded all the aspects of our testing in the following document:

[Test Plan](https://github.com/SchmittAndre/TextVenturer/blob/master/Documentation/TestPlan.pdf)

We want to make sure, that our users don’t stumble upon bug after bug, killing all the fun of the game.

Our testing consists of two parts.

Normal UnitTests, which ensure, that our classes and their respective functions work flawless

Input Simulator, to test, if the adventures, writtien in the TextVenturer-Scripting language, can be completed with the correct inputs

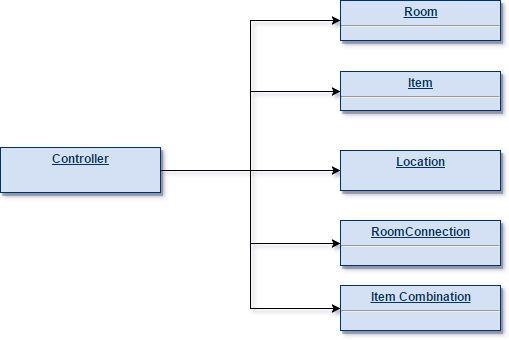
# REFACTORING

The last week we tried to refactor an example-project as practice. You can see our results at our individual git repos:  
[Andre](https://github.com/SchmittAndre/SE2_05_Refactoring)[Dominik](https://github.com/Possseidon/Fowler)[Simon](https://github.com/remllov91/SE_Refactoring)

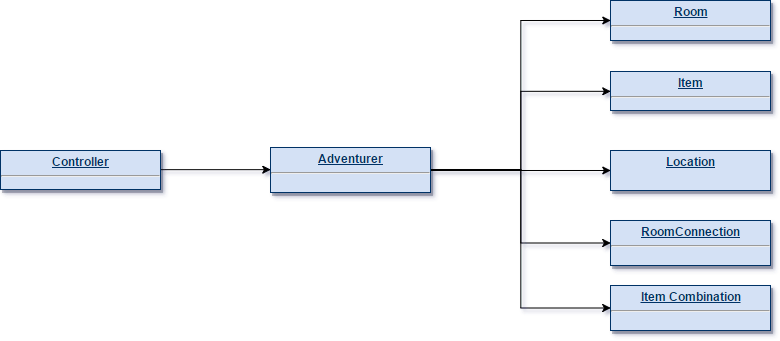
# PATTERN

For a long time we wanted only one adventure in our game, but as it was getting boring quickly, we implemented a container-pattern, to contain all the aspects of the adventure into one class. This allows us to have multiple instances of this class and load or save them dynamically with ease. Furthermore you can extend that new class without the problem of it possibly getting confusing over time, if a lot of more things get added.

Simplified, our class structure, before the change, looked roughly like this:



After the implementation it now looks like follows:



Greetings TextVenturer

# METRICS

During the course of the last week we worked on metrics. [Here](http://h2688162.stratoserver.net:8080/www/CppDependReport.html#Main) you can see our first analysis.  
As you can see, we worked with “[CppDepend](http://www.cppdepend.com/" \t "_blank)“, which is an easy to use code analysis tool. It is unfortunately only a trial though and cost a fairly great amount of money, in case we want to keep using it. There aren’t many alternatives, and all of them are also paid.

The generated [TreeMap](http://h2688162.stratoserver.net:8080/www/NDependReportFiles/VisualNDependView.png" \t "_blank)shows, that a lot of problems are coming from the loadpng header. That header is not written by ourselves, and we therefore won’t change anything major in it. We need it, as the named reveals, to load the font, which is saved as PNG.

The only really “critical” problems are a few long functions, which had their reasoning, but might eventually get changed around, when we have time for it and decide to do so.

# INSTALLATION

Hey Guys and welcome back to TextVenturer,

since copy pasting the contents of a zip is annoying, we finally created an installer for our game. Therefore, all you need to do, is get the newest installer here: <https://github.com/SchmittAndre/TextVenturer/releases>

and install the game, like you do with any other. This also allows you to uninstall it in a clean way again as well.

We hope you enjoy our game!

Greetings, TextVenturer

# CONTINUOUS INTEGRATION

As we were working on our program, we realized that even though it compiles fine on one computer, it had problems on others. So we signed up at the Continuos Integration Service called Visual Studio Team Services, or short VSTS.

The Problem with this service is in fact, that it is made for companies who don't want to release their scrumming to the public eye. Therefore we don't have any way of opening it up publicly. The only way for you to take a look at it is, giving us your microsoft account name, so that we can add you as a reader.

[textventurer.visualstudio.com](https://textventurer.visualstudio.com/TextVenturer/_home)

# FINAL

So here is our summary of the whole project TextVenturer:

* [GitHub](https://github.com/SchmittAndre/TextVenturer)
* [Youtrack](https://textventurer.myjetbrains.com/youtrack/dashboard?id=5d0dda45-ce23-4c54-a2b0-165f09cde144)
* [Visual Studio Team Services](https://textventurer.visualstudio.com/TextVenturer)

Homework:

* [Project Vision](https://textventurer.wordpress.com/2016/10/12/project-start/)
* [Team Roles](https://textventurer.wordpress.com/2016/10/18/our-team/)
* [Software Requirements Specification](https://textventurer.wordpress.com/2016/10/23/software-requirements-specification/)
* [Overall Use Case Diagram](https://github.com/Damian1234523/Cheetah/blob/master/Documentation%20and%20Planning/UML%20Software%20Engineering.pdf)
* [Use Cases](https://textventurer.wordpress.com/2016/10/30/use-case-diagramm/)
* [Youtrack](https://textventurer.wordpress.com/2016/11/08/youtrack/)
* [Automated Testing](https://textventurer.wordpress.com/2016/11/15/testing-software/)
* [UML Class Diagram](https://textventurer.wordpress.com/2016/11/22/class-diagramm-crc/)
* [Database UML](https://github.com/Damian1234523/Cheetah/blob/master/Documentation%20and%20Planning/database%20uml.png)
* [Software Architecture](https://textventurer.wordpress.com/2016/11/29/software-architecture/)
* [Gantt Chart](https://textventurer.wordpress.com/2016/12/06/gantt-chart/)
* [Risk Management](https://textventurer.wordpress.com/2017/04/18/risk-management/)
* [Function Points](https://textventurer.wordpress.com/2017/05/02/function-point-calculation/)
* [Test plan](https://textventurer.wordpress.com/2017/05/08/test-plan/)
* [Refactoring](https://textventurer.wordpress.com/2017/05/16/refactoring/)
* [Continuos Integration](https://textventurer.wordpress.com/2017/06/19/continuous-integration/)
* [Pattern](https://textventurer.wordpress.com/2017/05/29/pattern/)
* Test Coverage Coming soon
* [Metrics](https://textventurer.wordpress.com/2017/05/30/metrics/)
* [Installation](https://textventurer.wordpress.com/2017/06/13/installation/)

TextVenturer

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 28.11.2016 | 1.0 | First release | A.Schmitt S.Vollmer |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 20

1.1 Purpose 20

1.2 Scope 20

1.3 Definitions, Acronyms, and Abbreviations 20

1.4 References 20

1.5 Overview 20

2. Architectural Representation 20

3. Architectural Goals and Constraints 21

4. Use-Case View 21

4.1 Use-Case Realizations **Fehler! Textmarke nicht definiert.**

5. Logical View 21

5.1 Overview 21

5.2 Architecturally Significant Design Packages **Fehler! Textmarke nicht definiert.**

6. Process View 21

7. Deployment View 21

8. Implementation View 22

8.1 Overview **Fehler! Textmarke nicht definiert.**

8.2 Layers **Fehler! Textmarke nicht definiert.**

9. Data View (optional) 22

10. Size and Performance 22

11. Quality 22

# Introduction

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Scope

everything

## Definitions, Acronyms, and Abbreviations

MVC- Model View Controller

## References

<http://dhbwse2016.pbworks.com/w/file/fetch/113168836/03_ArchitectureReverseEngineering_2016.pdf>

## Overview

This document show our software architecture.

# Architectural Representation

# Architectural Goals and Constraints

Our System architecture is based on an easy changeability of the components like changing the “View-Class”

# Use-Case View

n/a

# Logical View

## https://textventurer.files.wordpress.com/2016/11/classdiagram.png

# Process View

n/a

# Deployment View

tbd

# Implementation View

n/a

# Data View (optional)

tbd

# Size and Performance

n/a

# Quality

n/a

TextVenturer

Use-Case Specification: Alias

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <11/APR/17> | <1.0> | First Uploaded | Simon Vollmer |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Case Name 26

1.1 Brief Description 26

2. Flow of Events 26

2.1 Basic Flow 26

2.2 Alternative Flows 27

2.2.1 amount of inventory zero 47

3. Special Requirements 27

4. Preconditions 28

5. Postconditions 28

6. Extension Points 28

Use-Case Specification: Use

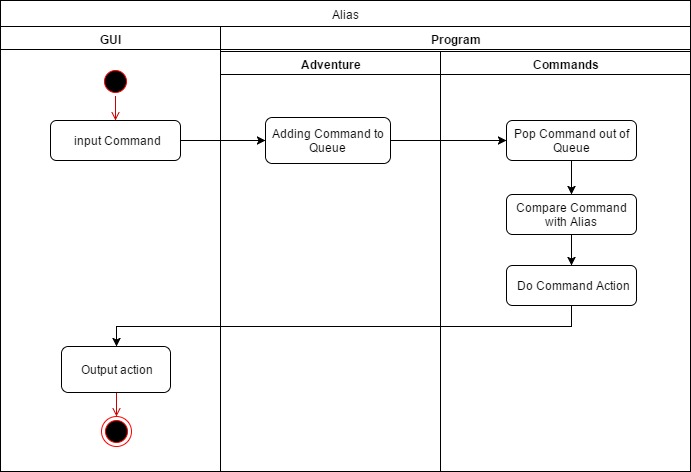
# Use-Case Name

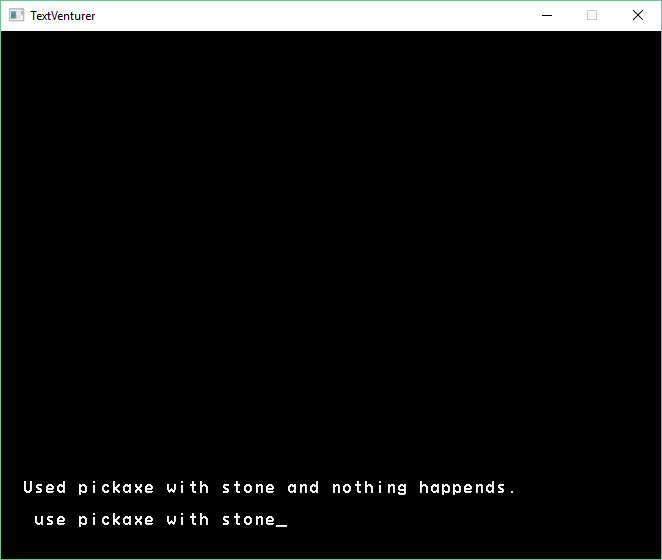
## Brief Description

This is our UC Diagram which starts if you type in a command.

# Flow of Events

## Basic Flow





## Alternative Flows

### Command is not available

If the chosen Command is not available an error massage appears on the UI

# Special Requirements

n/a

# Preconditions

n/a

# Postconditions

Wait for next input

# Extension Points

n/a

TextVenturer

Use-Case Specification: Combine

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <01/NOV/16> | <1.0> | First upload | André Schmitt, Dominik Vogel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Case Name 31

1.1 Brief Description 31

2. Flow of Events 31

2.1 Basic Flow 31

3. Special Requirements 32

4. Preconditions 32

5. Postconditions 32

6. Extension Points 33

Use-Case Specification: <Use-Case Name>

# Use-Case Name

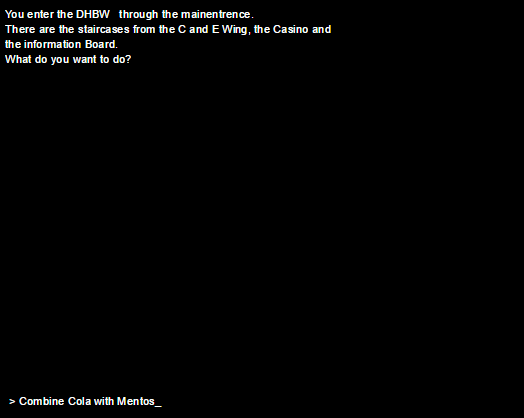
## Brief Description

This is our UC Diagram which starts if you type in “combine $itemA with $itemB”

# Flow of Events

## Basic Flow





# Special Requirements

n/a

# Preconditions

n/a

# Postconditions

Show name of new Item

# Extension Points

n/a

TextVenturer

Use-Case Specification: Use

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <12/DEC/16> | <1.0> | First Uploaded | André Schmitt |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Case Name 26

1.1 Brief Description 26

2. Flow of Events 26

2.1 Basic Flow 26

2.2 Alternative Flows 27

2.2.1 amount of inventory zero 47

3. Special Requirements 27

4. Preconditions 28

5. Postconditions 28

6. Extension Points 28

Use-Case Specification: Use

# Use-Case Name

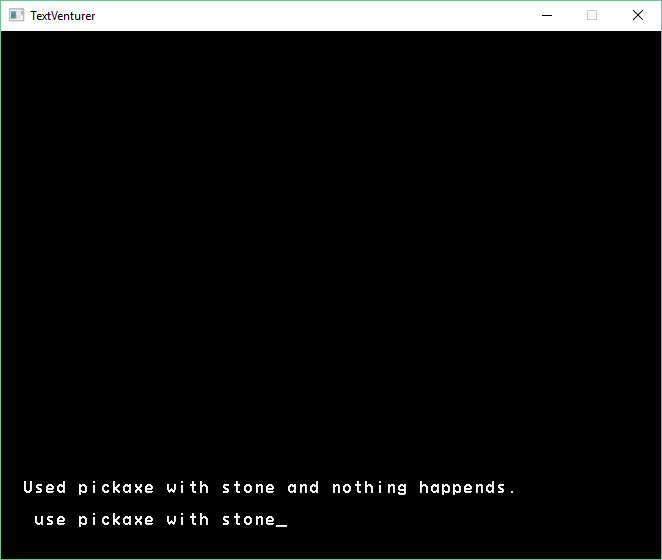
## Brief Description

This is our UC Diagram which starts if you type in “use <item> with <object>”

# Flow of Events

## Basic Flow





## Alternative Flows

### Items are not combinable

If the chosen Items are not combinable an error massage appears on the UI

# Special Requirements

n/a

# Preconditions

n/a

# Postconditions

Wait for next input

# Extension Points

n/a

TextVenturer

Use-Case Specification: Pick up

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <05/DEC/16> | <1.0> | First upload | André Schmitt |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Case Name 31

1.1 Brief Description 31

2. Flow of Events 31

2.1 Basic Flow 31

3. Special Requirements 32

4. Preconditions 32

5. Postconditions 32

6. Extension Points 33

Use-Case Specification: <Use-Case Name>

# Use-Case Name

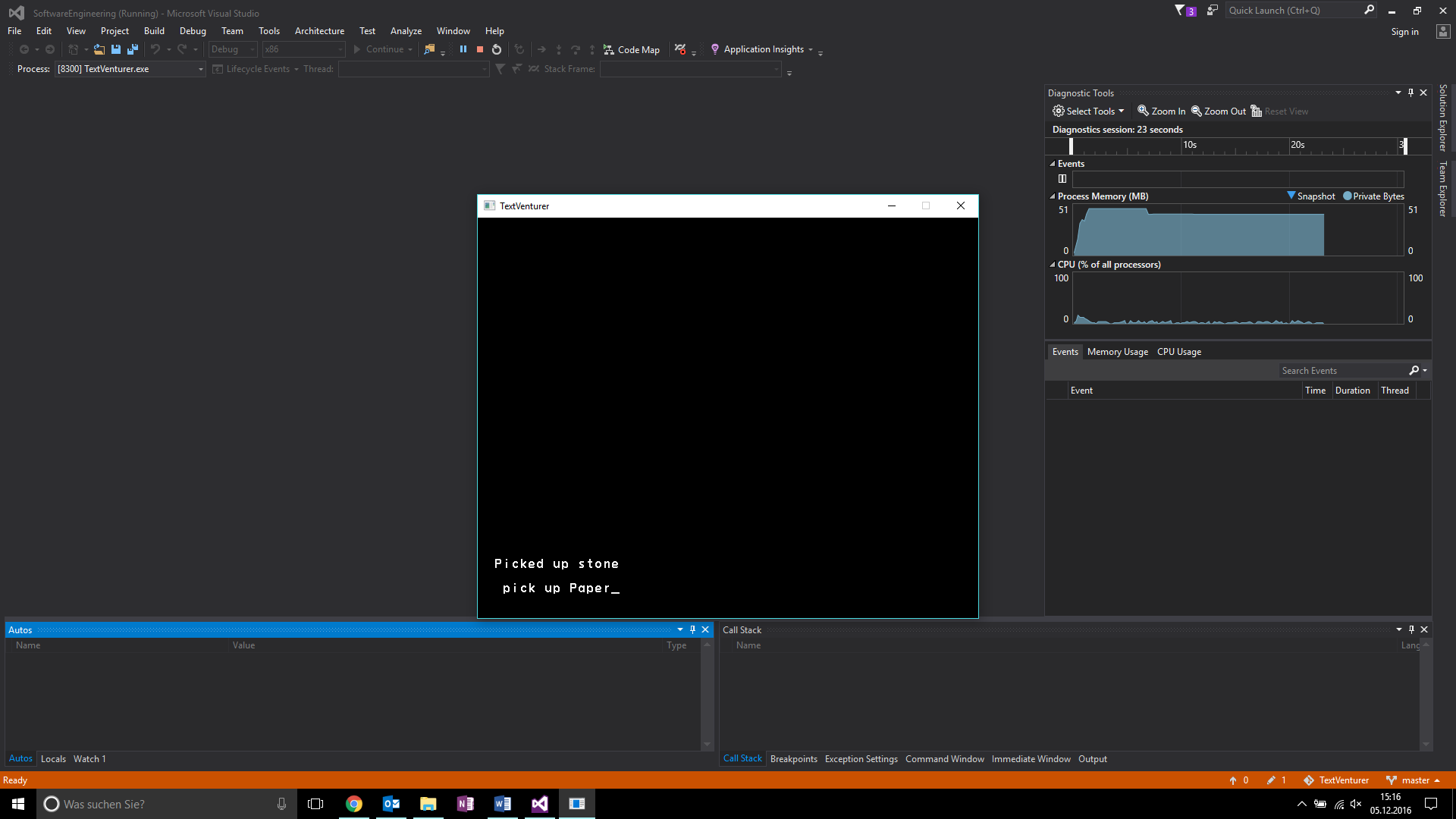
## Brief Description

This is our UC Diagram which starts if you type in “Pick up $itemA”

# Flow of Events

## Basic Flow

Download



# Special Requirements

n/a

# Preconditions

n/a

# Postconditions

Show acknowledge that you pick up sth. And what you just picked up

# Extension Points

n/a

TextVenturer

Use-Case Specification: get inventory

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <01/NOV/16> | <1.0> | First Uploaded | André Schmitt |
| <07/DEC/16> | <1.001> | Correct some things | André Schmitt |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Case Name 26

1.1 Brief Description 26

2. Flow of Events 26

2.1 Basic Flow 26

2.2 Alternative Flows 27

2.2.1 amount of inventory zero 47

3. Special Requirements 27

4. Preconditions 28

5. Postconditions 28

6. Extension Points 28

Use-Case Specification: Get inventory

# Use-Case Name

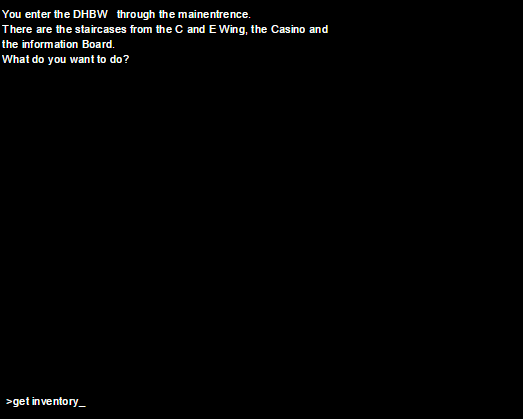
## Brief Description

This is our UC Diagram which starts if you type in get inventory

# Flow of Events

## Basic Flow





## Alternative Flows

### amount of inventory zero

If the number of things in your inventory is zero it Outputs “there is nothing in your Inventory”

# Special Requirements

n/a

# Preconditions

n/a

# Postconditions

Wait for next input

# Extension Points

n/a

TextVenturer

Use-Case Specification: Enter a room

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/11/16> | <0.1> | <UC-Enter Room> | <Simon Vollmer, Dominik Vogel, André Schmitt> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-Case Name 2

1.1 Brief Description 2

2. Flow of Events 2

2.1 Basic Flow 3

2.2 Alternative Flows 3

2.2.1 < First Alternative Flow > **Fehler! Textmarke nicht definiert.**

2.2.2 < Second Alternative Flow > **Fehler! Textmarke nicht definiert.**

3. Special Requirements 3

3.1 < First Special Requirement > **Fehler! Textmarke nicht definiert.**

4. Preconditions 3

4.1 < Precondition One > 4

5. Postconditions 4

5.1 < Postcondition One > **Fehler! Textmarke nicht definiert.**

6. Extension Points 4

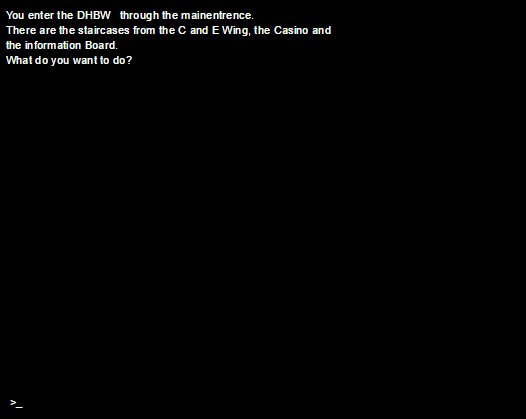
6.1 <Name of Extension Point> **Fehler! Textmarke nicht definiert.**

Use-Case Specification: <Use-Case Name>

# Use-Case Name

## Brief Description

In the following UC-Diagram u can see what happens if you enter a new room.C:\Users\SimonVollmer\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Raum.png



# Flow of Events

## Basic Flow

The purpose of the “Room Enter” UC is to get every information whenever you enter a new room.

## Alternative Flows

n/a

# Special Requirements

n/a

# Preconditions

n/a

## < Precondition One >

# Postconditions

n/a

# Extension Points

n/a