

# TDT4240 Programvarearkitektur

## X2

Julius Buset Asplin  
Emil Grunt  
Hvard Kindem  
Dimitry Kongevold  
Nicolay Thafvelin  
Milos Zlatkovic

April 27, 2012

### **Abstract**

This document describes product delivered by team X2 in the course TDT4240 Software Architecture. The document contains: design and implementation details, tests run on the final product and during the development, the Users manual and the list of inconsistencies between implementation and the architecture.

The main attribute that have been focus under development of the project is modifiability. Game, as in now, is just an engine with the possibilities to add new features, characters, maps and power ups. Those opportunities could be explored by an advanced user as well as a developer, or future support personnel, by adding an XML file to a specific folder.

# Contents

<b>1</b>	<b>Implementation Details</b>	<b>3</b>
<b>2</b>	<b>User's Manual</b>	<b>3</b>
2.1	Requirements . . . . .	3
2.2	Getting started . . . . .	3
2.3	Gameplay . . . . .	3
<b>3</b>	<b>Test Report</b>	<b>3</b>
<b>4</b>	<b>Relationship with the architecture</b>	<b>5</b>
<b>5</b>	<b>Problems, issues and points learned</b>	<b>6</b>

# 1 Implementation Details

## 2 User's Manual

### 2.1 Requirements

- Visual Studio 2010 [1]  
As the project is built using Visual Studio 2010, this recommended. It is possible to compile the project in earlier versions as well, but this means rebuilding the project structure or convert it.
- XNA Game Studio 4 [2]  
This is required.

### 2.2 Getting started

To compile and run the project, open it up in Visual Studio and simply press F5. This will compile and run the program. The first screen you will see is the character selection screen. Here you can select your characters by moving your pointer with the XBox control pad or the W, A, S, D keys for player 1 and the arrow keys for player 2. Player 1 select his or her character by pressing v and player 2 by pressing j. After selecting your characters, start the game by pressing return. Here you can move around with the XBox control pad or the W, A, S, D keys for player 1 and the arrow keys for player 2. You can attack using v for player 1 and j for player 2.

### 2.3 Gameplay

The goal of the game is to knowck the opposing player(s) off the map. This is done by attacking them and usingthe knockback effect of the attack to get them off. Hitting your opponen(s) will increase the power of the knockback and further improve your ability to kick them off the map. When a player is kicked off the map, he will respawn after a few seconds, loosing a "life".

## 3 Test Report

Modifiability 1: Add new character.	
Executor:	To add a new character up should be easy and possible to do with out re compile A new character should be visible in character selection menu. With out the necessarily to re compile Added a new character XML file in a specific folder
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Modifiability 2: Add new map.	
Executor:	<p>To add a new map up should be easy and possible to do with out re compile</p> <p>A new Map should be possible to chose in Map selection menu. With out the necessarily to re compile</p> <p>Added a new map XML file in a specific folder</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Modifiability 3: Add new power up.	
Executor:	<p>To add a new power up should be easy and possible to do with out re compile</p> <p>A new power up should have the possibility to drop in game. With out the necessarily to re compile</p> <p>Added a new power up XML file in a specific folder</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Modifiability 4: Replace a controller method.	
Executor:	<p>The change in controller class should not affect other, functionalities.</p> <p>The program should use the new controller method after recompile the game.</p> <p>Changed the name of the controller class in the code</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Performance 1: Frame rate	
Executor:	<p>The game should appear smooth for the user.</p> <p>The frame rate should not go below 30fps.</p> <p>Play the game for 20 min.</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Performance 2: Game delay	
Executor:	<p>The player should not notice any waiting between the pressed button and the executed action on the screen.</p> <p>The action should be done immediate after clicking.</p> <p>Start the game and execute actions/game play.</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Portability 1: Run on XBox	
Executor:	<p>The game should be runnable on Xbox 360.</p> <p>The game should run on Xbox as good as on PC.</p> <p>Uploaded game to the Xbox and started it.</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

Testability: Beta	
Executor:	<p>The game should be tested as created. for faster detections of bugs.</p> <p>The game should be able to run with out all of the features before release date.</p> <p>During the developing Simple tests have been executed.</p> <p>It was possible to run tests on the unfinished version of the game a week before delivery.</p> <p>Passed.</p>
Date:	
Stimuli:	
Expected response:	
Executed actions:	
Observed response:	
Evaluation:	

## 4 Relationship with the architecture

The architecture description could be found in ArchitecturalDoc.pdf file. The implementation has some inconsistencies with the planed architecture, caused by several different reasons. This chapter lists them and tries to explain the developers view and logic behind those changes. Model View Controller (MVC) In the architectural description modifiability is one of the primal attributes for the architecture, and had lead to the choice of MVC. Through the implementation we can observe some inconsistencies with the MVC architectural pattern. The reasons for these inconsistencies are caused majorly by the use of FarseerPhysics COT in the project. Development View and Control-Float Control-float of the program has been changed to fit better with FarseerPhysics engine, and now is different from the sequence diagram presented by the Scenario View

and Development View in the architectural description, because of the great damage from the architectural-drift. Possible reasons for inconsistencies As the development team worked with the XNA framework, there were difficulties fitting MVC into XNAs pipe and filter architecture. Though the Architecture document gave a possible solution for how to implement MVC inside a pipe and filter pipeline, the difficulties only got worse with FarseerPhysics engine, resulting into an architectural-drift. This could be avoided earlier in the process if the architects had more experience and understanding of XNA and used COTs. If the development view was more strict and detailed, so it could give more support to the developers, the possibility of an architectural-drift and erosion would be considerably lower, resulting in a much more consisting product.

## 5 Problems, issues and points learned

The main problem we have experienced during this project was the lack of time and experience. As well as communication problems with inn the group and different programming experience and habits. As our first try as software architects we underestimated the value of creating a detailed architecture and fast-forward towards the implementation. This resulted in an architectural-drift and erosion lowering the required quality of the product and increasing the implementation time. With a better planed out and detailed architecture, many of issues and problems we have encountered would be solved. For instance the use of the FarseerPhysics engine that we thought would save some of implementation time ended up causing an architectural drift. Next time well be more careful in our choice and usage of COTs. And gather enough knowledge and understanding of bough developing platform and COT, before taking a next step. If we were to start again from scratch we would try to stay closer to chosen architecture and use more time to think over the product. We would choose our architectural/design patterns more wisely a consideration to the environment/platform well use. As well as create more detailed views, so it will be more informative for the stakeholders. Specially the developing view.

## References

- [1] Visual Studio — MSDN, *Microsoft*, 2012,  
<http://msdn.microsoft.com/en-us/vstudio/aa718325>
- [2] Download: Microsoft XNA Game Studio - MSDN, *Microsoft*, 2012,  
<http://www.microsoft.com/en-us/download/details.aspx?id=23714#overview>