TDT4240 Software Architecture

Smash Bros
Implementation Document
XNA Group 2

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

Primary quality attribute:
Modifiability
Secondary quality attribute:
Usability

Contents

1 Introduction

This document is a description of the product delivered by Team X2 in the course TDT4240 Software Architecture. The document contains design, and implementation details, tests results from testing done during development and on the final product, the user manual, and a discussion on the architecture implementation.

During development the main focus was on modifiability. The game has the possibility to be expanded with new features, characters, maps and power ups. The game uses the FarSeer Physics Engine 3.2 (FPE) a COTS for physics simulation and collision detection. FPE lets the developer easily create new dynamic objects in the game world, which could be used to create dynamic levels at a later date.

2 Implementation Details

As mentioned before the game is built using the FarSeer Pyshics Engine version 3.2 (FPE). This is an open source physics engine for XNA. To cut down on the development time, we decided to use this as it handles 2D collision detection and has a lot of physics features. Of these features, we are using the collision detection and basic gravity and reflection physics.

The overarching architecture is modelled after the Model View Controller (MVC) pattern. This makes it easy get get an overview of the project as it strictly differentiates between data, views and input/data handling.

The game also makes extensive use of the Observer Pattern(OP) and State Machines.

Some structures in the game can be serialized to the JavaScript Object Notation (JSON) standard.

3 User's Manual

3.1 Requirements

- Visual Studio 2010 [VS10]
  The game uses XNA 4.0 so earlier versions of Visual Studio are not guaranteed to work this also requires the project to be converted.
- XNA Game Studio 4 [XNA]
  The game is built on XNA Game Studio 4.0

3.2 Compiling and Running the game

1. Open the SmashBros.sln file located in the TDT4240-X2/SmashBros folder.

2. Right click the SmashBros Project in the Solution Explorer and press "Set as Startup Project".

3. Go to the menu and press Debug>Start Debugging, or press F5.

4. The project should compile and start the game.

3.3 Gameplay

The first screen you will see is the character selection screen. Here you can select your character 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 the V key, and player 2 by pressing the J key. After selecting your characters, start the game by pressing the Return key. The game then lets you choose a level, which is selected in the same way as the character. As before you press the Return key to continue. When the game starts you can move around with the XBox control pad, the W, A, S, D keys for player 1, and the arrow keys for player 2. You can attack using the V key for player 1 and the J key for player 2. The goal of the game is to knock the opposing player, or players, off the map. This is done by attacking them using the knockback effect to push them off the map. Hitting your opponent, or opponents, will increase the power of the knockback and further improve your ability to knock them off the map. When a player's character is knocked off the map, he will respawn after a few seconds, loosing a "life".

4 Test Report

4.1 Functional Requirements

Choose character	
Executor:	Milos
Date:	29.04.2012
Time used:	1 min
Evaluation:	Players are able to choose characters correctly and continue to the map select screen. Note that some characters are duplicates.

Map selection	
Executor:	Milos
Date:	29.04.2012
Time used:	1 min
Evaluation:	Players are able to choose what map to play and continue to the game. Note that the maps on the list are duplicates.

Move your character	
Executor:	Julius
Date:	29.04.2012
Time used:	2 min
Evaluation:	The characters are able to move around the map and able to jump.

Attack enemy character	
Executor:	Julius
Date:	29.04.2012
Time used:	5 min
Evaluation:	All players are able to attack enemy players.

Multiple characters with different looks	
Executor:	Håvard
Date:	29.04.2012
Time used:	5 min
Evaluation:	There are currently 3 implemented characters. The coon, spiderman and fat superman.

Multiple map support	
Executor:	Håvard
Date:	29.04.2012
Time used:	5 min
Evaluation:	The game supports multiple map through it's own map loader.

Display and pick up power-ups	
Executor:	Nicolay
Date:	29.04.2012
Time used:	10 min
Evaluation:	Functions correcly. Pick ups fall from the sky and the players are able to pick it up.

Display selected characters and map	
Executor:	Dimitry
Date:	29.04.2012
Time used:	5 min
Evaluation:	Works as intended. The selected characters are shown and the map is displayed.

Realistic physics	
Executor:	Emil
Date:	29.04.2012
Time used:	15 min
Evaluation:	Obtained through FarSeer

Music and sound effects	
Executor:	Håvard
Date:	29.04.2012
Time used:	5 min
Evaluation:	Sound effects on attacks as well as when starting the game. No music.

4.2 Quality Requirements

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

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

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

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

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

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

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

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

5 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.

6 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 we'll 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 we'll use. As well as create more detailed views, so it will be more informative for the stakeholders. Specially the developing view.

References

[VS10] Visual Studio | MSDN, *Microsoft*, 2012,
http://msdn.microsoft.com/en-us/vstudio/aa718325

[XNA] Download: Microsoft XNA Game Studio - MSDN, *Microsoft*, 2012,
http://www.microsoft.com/en-us/download/details.aspx?id=23714#overview

[FsPe] Download: Microsoft XNA Game Studio - MSDN, *Microsoft*, 2012,
http://http://farseerphysics.codeplex.com/