+

Do It Tomorrow

Project Plan

Tank Game

Team M – Daniel Hogan

Colm Mulhall

Conor Sargent

Summary

The objective of this project isto create a clone of the 1980 classic arcade game ‘BattleZone’. The game must make use of 3D vector based graphics. The team is expected to have a playable prototype up and running for this year’s Games Fleadh at the Limerick Institute of Technology.

Team

**Name**: ‘Do It Tomorrow’

* Conor Sargent – Team Leader/Tester
* Colm Mulhall – Documentation/Tester
* Daniel Hogan – Lead Coder

Project Title

Tank Game

Version Number: 0.8

# Change History

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Comments** |
| 14/2/13 | Colm Mulhall | Project plan first created. Week one and two documented. (Version 0.1) |
| 22/2/13 | Colm Mulhall | Week three write-up and summary added. (Version 0.2) |
| 5/3/13 | Colm Mulhall | Document updated with progress of weeks four and five. Approach added. (Version 0.5) |
| 18/4/13 | Colm Mulhall | Week six to nine added. (Version 0.8) |

Approach

# Overall Approach

We decided that the “Scrum” methodology would be the best framework to use to manage this project. It is an agile method of production management. The reason that we chose this approach was because of the nature games development. The usual sequential approach such as the waterfall model or the star model does not suit games development as it is difficult to plan ahead.

# Requirements Phase

We have until week 11 to get the project done. We have decided that by week 10 all code work in the game should be finished. This means that by Friday 19th April we have to get our game ready for submission. We are aware of the risk that we might fall behind if coding doesn’t go to plan and we have taken this into account.

**Design Phase**

It was agreed upon at the start by all team members that we would stay as true as possible to the original 1980 game “BattleZone”. This meant that we would try and replicate the design of the original game as closely as possible.

**Implementation Phase**

We have decided to create the game using the XNA framework in Microsoft Visual Studio. This is a gaming specific framework. The game will run on Windows machines. We will use GitHub to store our code and allow us to update the project. This has the added bonus of enabling us to revert back to older code if necessary.

**Testing Phase**

We will be testing the game throughout the project. At the end we will have final tests which will comprehensively cover all features of the game in an effort to find any missed bugs or to add features where they may be needed. It is important to regularly test the game so that we can spot bugs as quickly as possible and fix them before they become a bigger problem.

Technical Requirements

We are creating the game in C# using the XNA framework in Microsoft Visual Studio 2010. The game will run on Windows.

Project Plan

# Week 1

# On our first week we had to pitch our proposal for our project. We decided to choose the tank game. We had to write up our proposed approach and present it.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Proposal document | High | Write a one page proposal describing why we want to do the tank game and how we would go about it. |
| Proposal presentation | High | Create a presentation. Assign which slides each team member will present. |
| Present our  proposal | High | Present our proposed project approach in front of the class. |

# Week 2

We have been allocated the project that we wanted. We now have to begin work on our design document and present our design this Friday.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Begin Design Document | High | Make a start on the design document. This needs to be submitted at the end of the week. |
| Present our design | High | We need to present our design to the class and our supervisors. We will each take two slides. |
| Create UML diagrams | High | As part of our design document we need to create several UML diagrams describing the game in detail. |

# Week 3

This week we created our code base and set up our project on GitHub. We also needed to assign team roles to each member.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Code Base | High | Set up the code base for the game. |
| GitHub setup | Medium | Create Github account and set up a repository for our game. |
| Team  roles | High | Assign each team member a role in the team. |

# Week 4

The models for the tank and bullets were created in Blender this week. Work began on scaling them correctly into the game. The ability to shoot these bullets was then implemented.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Create Models | High | Create models for tank and bullets in Blender. |
| Scale Models | High | Scale the models correctly in the game. |
| Insert Bullets | High | Once the models have been created and scaled, the bullets are then implemented into the game. |

# Week 5

Focus moved to getting collision detection working between game entities. This included collisions between the player, enemy and bullets. An obstacle was added to be tested with collisions. A radar was also added this week that will display the location of the enemy relative to the player. Obstacles were then added which spawn randomly in the game each time it runs.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Collisions | High | Implement collisions between game entities. |
| Obstacle | Medium | Insert and obstacle into the game to be tested with collisions. |
| Radar | Medium | Radar to display player and enemy locations added. |
| Random Obstacles | High | Obstacles were added to the game, which spawn in random locations. |

# Week 6

After the random obstacles were added, we had to test collisions again. We also had to ensure that obstacles don’t spawn on the enemy or the player at the start of a game. The enemy tank needed work done on it so that it shot at the player. The bullet model also spawns closer to the tank turret. A basic scoring system was also added this week. Some sounds were also added. We have added three levels into the game in anticipation for the Games Fleadh next week.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Obstacle Collisions | High | Ensure that the obstacles don’t spawn on the player or the enemy each time the game starts. |
| Enemy shooting | High | Ensure that the obstacles don’t spawn on the player or the enemy each time the game starts. |
| Levels | High | Add three levels to make a fully playable game. |

# Week 7

This week we got work done on the radar, health and visuals. Player health was implemented and a background of mountains was added. The radar was also updated to display the players’ position better.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Fix Radar | Medium | Fix radar to ensure that the players’ position is correctly mapped. |
| Player Health | High | Implement player health into the game. |
| Visuals | Medium | Stay true to the original game by making sure that the mountain background is added and that all objects are green. |

# Week 8

Scoring was finished this week. A menu was implemented so that a player can choose to start the game instead of being thrown straight into it. Enemy bullets were tweaked so that they no longer go through obstacles.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Finish Scoring | High | Scoring for the player finished. |
| Create Menu | High | A main menu for the player to select when to start a game. |
| Fix Enemy Bullets | Medium | Fix enemy bullets so that they do not go through obstacles. |

# Week 9

The radar was finished. It now shows what direction the player is facing rather than just the location. The tank model was scaled back correctly. Bullets now follow the direction that the tank is facing, making the shooting system more accurate.

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Description** |
| Finish Radar | Medium | Radar now changes direction based on the players look. |
| Scale Tank | High | The tank was scaled back correctly. |
| Bullet Direction | High | Bullets now follow the direction that the tank is facing. |