

Department of Computing and Networking

Games Degree Year 3

## First Iteration Report

## Aoibhinn McAuley C00146854

## Darren Kelly C00157765

October 2013

Table of Contents

User Stories Implemented…………………………………………………………………………………3

User Stories Not Implemented...….…………………...….………………………………………………5

Additional work completed…………………………………………………………………….………....5

Work breakdown...........................................................................................................................................6

Reflection…....................................................................................................................................................6

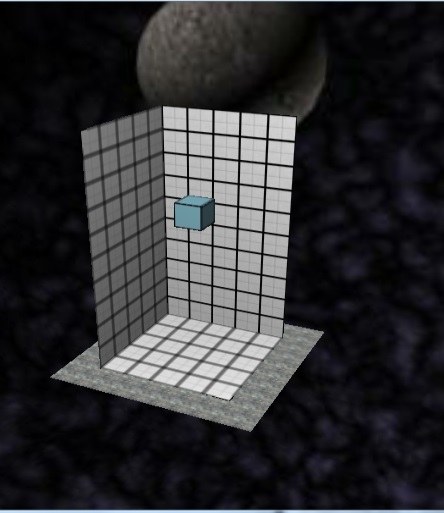
**User Stories Implemented**

**Title:** Create a game environment.

**Conditions of satisfaction**

As a user I want the game to have a textured plane. I want the matrix to be visible and the background to be appropriately textured. We completed this user story and we were happy with its outcome.

Storyboard:



This is a screenshot of our game showing a custom textured grid, a cube with a custom texture also, an appropriately textured plane and a skybox which all help add to the immersion of the game’s environment and making it more appealing to the player.

# Title: Create Shapes and have them falling slowly down the screen.

**Conditions of satisfaction**

As a user I want there to be different shapes that fall down the screen slowly in increments of a cubes width. For the first sprint the only shape we wanted to create was a cube.

# Title: Add Collision detection between shapes and plane user story

**Conditions of satisfaction:**

As a user I want the shapes to land on the plane when they reach it.



**Storyboard**:

This image shows the cube we created.

# Title: Add Collision detection between shapes and sides of matrix

**Conditions of satisfaction**

As a user I want the shapes to stay inside the matrix even if I try to move them out of it.

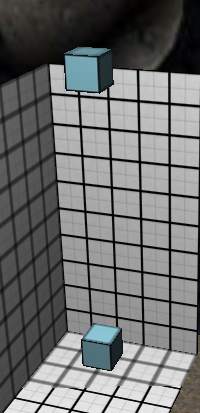
**Title:** Have a new shape spawn after the pervious one lands

**Conditions of satisfaction:**

After I have placed the current shape I want a new shape to spawn above the grid.

**Storyboard:**

This image shows a cube falling to the bottom of the grid while another one waits above it, which will start to fall when the one before it lands.

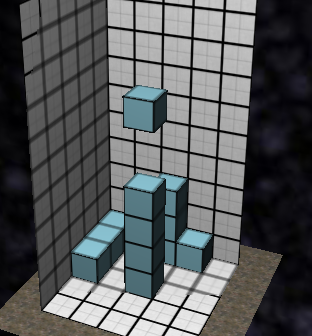
****

# Title: Add Collision detection between shapes

**Conditions of satisfaction**

As a user I want the shapes to land on each other properly. For this sprint we will use just use cubes.

**Storyboard:**

****

This screen shot shows the cube objects colliding as expected and landing on each other appropriately.

**Title:** Add User control.

**Conditions of satisfaction**

As a player I want to be able to move the shapes on the screen appropriately so that I can place them where I want and progress in the game. I want to be able to use the left and right directional keys to move the shape to the left or the right, I want to be able to instantly drop a piece by pressing the spacebar and rotate it by pressing the Z and X keys.

For this sprint we focused on moving the block left and right with the directional keys.

**Storyboard:**

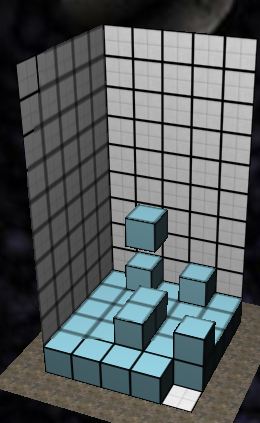
This is demonstrated in the positioning of the blocks in the screenshot above.

**Title:** Clearing a full grid square

**Conditions of satisfaction**

When I have placed enough shapes to make up a full square on the grid I want all sections of the shapes associated with it to disappear and for my score to increase.

**Storyboard:**

****

This image shows a an almost full grid square, which when the last cube falls into place, all associated cubes will vanish and any cube resting on them will subsequently fall into place on the grid.

**User Stories Not Implemented**

**Title:** Random colour cubes.

**Conditions of satisfaction:**

As a user I want the different cubes that fall to be random colours. We did not get around to completing this user story because creating a vector of cubes and getting them colliding etc. took longer than expected.

# Additional Work Completed

# Additional user stories we implemented/started that were not in our proof of concept demo were the following:

# Create a game environment.

* Add Collision detection between shapes and sides of matrix.
* Clearing a full grid square.

**Work Breakdown**

Aoibhinn: I created the plane and initial cube that was positioned at a height above the ground. I added in the grid and textured all the game entities appropriately, I also I added a skybox. When the spacebar was pressed the cube began falling down the screen as expected and the user was able to move the cube left and right with the directional keys and the cube stopped when it reached the bottom of the grid. I then created a list of the cubes and stored them in a 2D vector and had them moving appropriately. I had them colliding with the sides of the matrix and when one cube reached the bottom of the grid and stopped another would begin falling.

Darren: I continued on Aoibhinn’s creation of the vector of squares by creating an empty 3D array of pointers to cubes. When a cube had stopped moving, I passed the cube into the correct position to match the right cell in the empty pointer array of cubes. This allowed me to track whereabouts the cubes in space with use of an unseen grid. This then greatly helped me to add in collision detection between the cubes so they could land on top of each other. I also added in a for-loop to check if the cubes filled a full square grid of the 3D pointer array. I then moved them off-screen. The corresponding blocks above the cleared square then begin to descend towards their appropriate locations.

**Reflection**

* During this iteration we learned how to change and use unique textures in ogre by modifying exiting textures and editing the example script.
* We learned how to implement a timer to control the movement of an entity.
* We learned how to create multiple objects of one entity (cube) and assign a new node name and entity name to a list of objects with use of the #include <sstream> library function to convert a changed integer for each object in the list to a string.
* We learned how to manage our time efficiently to produce a working demo that fulfilled our proposal for the proof of concept iteration.
* We learned how to create, initialise and manage a 3D array of vectors.
* We learned to work as a well-functioning team and learned to distribute work between ourselves fairly.
* We also learned our way around the amazing layout of ogre and its brilliant ways we now have a much better understanding and knowledge of C++ and the ogre environment.

We don’t believe we would have approached anything differently if we had to redo this iteration. Everything seemed to go smoothly enough, a few minor issues arose but nothing we couldn’t resolve.