Rhys Richards – Stephen Thomas – Nathan Thomson

Pac Man – Group Project

Table of Contents

[Phase One – Requirements 2](#_Toc437274438)

[Project Manager 2](#_Toc437274439)

[User requirements 2](#_Toc437274440)

[System requirements 2](#_Toc437274441)

[Questions for the user 2](#_Toc437274442)

[Designer 1 3](#_Toc437274443)

[User requirements 3](#_Toc437274444)

[System requirements 3](#_Toc437274445)

[Questions for the user 3](#_Toc437274446)

[Designer 2 4](#_Toc437274447)

[User requirements 4](#_Toc437274448)

[System requirements 4](#_Toc437274449)

[Questions for the user 4](#_Toc437274450)

[Developer 1 5](#_Toc437274451)

[User requirements 5](#_Toc437274452)

[Questions for the user 5](#_Toc437274453)

[Developer 2 6](#_Toc437274454)

[User requirements 6](#_Toc437274455)

[Questions for the user 6](#_Toc437274456)

# Group Meetings

## Meeting 1

**Members Present:** Rhys Richards, Stephen Thomas, Nathan Thomson.

**Members Absent:** None.

**Agenda:** Project Specification Analysis Discussion.

**Meeting Date | Time:** Thursday 26/11/2015, 15:00

Discussion and deciphering of the project brief.

**By Next Meeting:** Thursday 03/12/2015

Read through the brief and understand fully what is needed in the project phase 1.

## Meeting 2

**Members Present:** Rhys Richards, Stephen Thomas, Nathan Thomson.

**Members Absent:** None.

**Agenda:** Requirements phase discussion and work process.

**Meeting Date | Time:** Thursday 03/12/2015, 15:00

Discussion and deciphering of the requirement’s phase.

Split the requirements between members of the group.

**By Next Meeting:** Tuesday 08/12/2015

Write the requirements that were delegated to each group member.

Write questions for the user in order to extract the remaining requirements in the document.

Next meeting will be before we meet with the user.

# Phase One – Requirements

## Project Manager

### User requirements

We’re looking to include small one-player games, principally in this case a Pac-man like game (title to be decided,) which like the game itself gives a sense of anarchy within a familiar setting. We're really looking for some clever twists on the original to give a sense of a world where any rules can be broken. At the same time, we need to balance the game play, so whatever additional behaviour the pac-man character has must be echoed or countered by the ghosts.

The art for the games selected will be developed at a later date and so programmer art, or even blank sprites or blocks are acceptable for this prototyping stage. In these games the focus must be on *fast paced accessible and fun game play rather than exciting graphics.*  The format of the demos can be left as simply a playable level or as a complete mini game with front screen and high scores.

I'll need to see the full requirements for the game to ensure the game play, style etc. meet the brief before the game progresses to the demo stage.

* User should be able to break the rules i.e. jump over walls
* User should be challenged by the ghosts having the same capability as the character or something to counter it.
* User should be able to either access the playable level or the complete mini game
* User may be able to see their high scores
* User should see controls on front screen

### System requirements

* Should allow the user to play the game as one player
* Be accessible as a mini game whether level or complete mini game

### Questions for the user

* What would make the game-play “fun”
* What would make the game “fast paced” accessible?
* Is it better to do a playable level or a complete mini game?
* What would the high scores be in? i.e. overall score or separate variables such as time completed in, number of deaths if has checkpoints. etc.
* How would the game accomplish having a “sense of anarchy”?
* What “clever twists” on the original will the game include?

## Designer 1

### User requirements

It’s important that the game play is fast and the controls are easy. The specification for the artwork required also has to be delivered as a priority. The look we’re hoping for is a mixture of games from 70’s and 80’s arcades, possibly interspersed with some re-takes of modern games in a similar style. The games should have music and other affects and possibly some scripted sections to ease level development. Ideally the game should have some kind of splash screen to start, mimicking original arcade game, followed by a screen of instructions and possibly a brief story regarding the game, this may be expanded into multiple frames or even a montage in production. The game play is all important while the games are going to be short they need to be engaging.

* User should feel the game-play “as being ‘fast’
* User should feel that the game controls are ‘easy’
* User should see a screen of instructions with the controls and the objective

### System requirements

* Game-play should be fast paced
* Game controls should be easy
* Art work is based off 70’s and 80’s retro look
* scripted levels to ease level development
* Game should have music
* Game should show splash screen upon launch
* Game should show instruction screen after the splash screen with controls listed
* Game should be short
* Game should be engaging
* Game may show a brief story after

### Questions for the user

* How would we make the requirement “game-play should be fast” quantifiable?
* How would we make the requirement “game-controls should be easy” quantifiable?
* what ‘retro games’ from the 70’s and 80’s would the game be based off?
* what modern games would be good to base the art work off?
* What style of music would the game incorporate?
* What would make the game ‘short’ and ‘engaging’? How would that be accomplished?
* what would the story of the game be?

## Designer 2

### User requirements

We’re looking to really advance the coin-op feeling to the games possibly with some sort of heads up display which resembles an old arcade cabinet. The demos produced will probably be re-worked to make them more artistic. The game play will be what makes the game and decides which ones are used in the final game. The sound and graphics can either mimic the original game exactly or make a more up to date version with better graphics and sounds. I like the splash screen idea, perhaps use a classic looking splash screen then a much more modern looking game. Alternatively, perhaps components of the game could be very retro while others look much more modern, for example a pack man game with the maze in 2D neon with 3D Pac man and ghosts.

* The game should really feel like it is being played on a coin operated arcade machine with a heads up display to match.
* The sound and graphics will be like the original or with a more modern twist
* Could have a classic looking splash screen with a more modern looking game.

### System requirements

* The demo “fill in” art will be reworked to be more artistic
* Classic looking splash screen
* The sound or graphics can either mimic the original game or a more modern version
* Could be a mix of old style and new assets

### Questions for the user

* Can you clarify on what you mean by “The game play will be what makes the game and decides which ones are used in the final game”?

## Developer 1

### User requirements

I would advise the use of both fixed and variable rate game timers, a fixed timer on the logic (at 60fps) and a variable timer on the rendering (capped at 60fps). These can be synchronised so that if the logic fails to update the image also skips an update until the frame rate drops below a threshold. The graphics API of choice is OpenGL as this is being used in development of the game along with Open AL as a sound API. The demo may be developed using Win32 or SDL and simple sprites though these will need to be converted into OpenGL at a later stage. To minimised development time open source or built in image loading libraries such as Corona etc. could be used.

* The user desires some ‘simple sprites’ in the game, this art style will tie in very nicely with the coin operated art style of the UI.

System requirements

* A fixed timer is needed in this game, the timer must be set at 60fps for the logic side and a variable cap for the graphics side at 60fps.
* Open GL Is a must for this game, as the project brief requires us to build this game using Open GL library’s.
* As well as using Open GL, Open AL will be our sound platform for the game in question, this works out perfectly as the Open GL and AL library’s sync together.
* We could look to implement image loading libraries that are already available to us, as stated this could save us many hours in the development stage.

### Questions for the user

* Can we see some example sprites in order to create a better picture for the images, or could we have some vague ideas as to what the game should look like in order to piece the art style together?
* Why is a fixed timer needed on the graphics side of things? Surely if the user’s computer is capable of over 60fps they should be able to achieve this!

## Developer 2

### User requirements

The focus of the development effort should be on producing a well written and well documented game demo, which can easily be translated into mini game with several levels at a later stage. The final game will have to be frugal in its use of memory and you may wish to design the game with this in mind. There are few constraints on the design of the demo though it would probably help if the same libraries as used in game are used for the mini games (i.e. OpenAL, OpenGL and Corona). This opportunity could be used to investigate other technologies such as FMOD and SDL however this is not essential and may detract from the task at hand.

* The user desires that we produce a ‘well written’ and ‘well documented’ game.
* The user desires that the game demo can be translated into a full fledge game filled with small mini games and several stages game.

System requirements

* The system should allocate and de-allocate its own memory, this system will have to be implemented into the game.
* The mini game aspect of the game should still use the same library’s, sounds and art style that the original game did.
* We could look into other options in terms of libraries for the system requirements, we could look into implementing SDL into the game in order to buff the graphics.

### Questions for the user

* What is meant by ‘well written’ and ‘well documented’? How would we explain what is meant by these? I can understand the requirement but unsure as to how far we should take them.
* Am I correct in understanding that the game should allocate and de-allocate its own memory?
* I am unsure of the idea behind a game and a mini game? Why are their two components to this game when it would be a lot simpler and better in my opinion if we just stuck with the game itself.
* Do we have to use these other options like SDL or whatever?