Swinburne University of Technology

SWE40002
Software Engineering Project B

Games Project
Excavator

Prepared for:

Dr. Viv Farrell

Mr. Andrew Trevillian

Prepared by:

Name	: Calvin Kang Chiak Kho
E-mail Address	: 101017894@student.swin.edu.au
Telephone Numbers	:0426369899
Student Numbers	: 101017894

Name	: Hung Che
E-mail Address	: 1799215@student.swin.edu.au
Telephone Numbers	: 0404038053
Student Numbers	: 1799215

Name	: Nick Wain
E-mail Address	: 9992057@student.swin.edu.au
Telephone Numbers	: 0447256338
Student Numbers	: 9992057

Table 1. Document Change Control

Version	Date	Authors	Summary of Changes
Version 1	24 October 2016	CALVIN KANG CHIAK KHO	Create a project plan for Software Engineering Project B

Table 2. Document Sign Off

Name (Position)	Signature	Date

1. Introduction

1.1 Background

The given name for this game is called as Excavator. The idea and concept art are from an underground city in Turkey which called as Derikuyu. The purpose of Derinkuyu is sheltering from raids, while keeping the city functioning. With the concept down, we then drew upon games such as Labyrinth's tile laying and rogue like game's random generation for replay ability for the core game mechanics. The key players of this project are our stakeholders, Andrew Trevillian our Games project lecturer/tutor and
Vivienne Farrell our software development convener/supervisor and Swinburne university as a whole as we are to be representatives at PAX with our game. Our project team who will be working on the game and finally PAX organization and the audiences that will be experiencing our product.

1.2 Key Project Personnel

1.2.1 Client

- Gamers
- Game Publishers

1.2.2 Stake holders

- Andrew Trevillian
  - Games project lecturer/tutor and our main supervisor for our entire production team, which includes Games students as well as Software Engineering students.
- Vivienne Farrel
  - Software Engineer convener and supervisor of our software engineering team, who is responsible for the primary development of the game.
- Swinburne University
  - Reputation of the university is tagged along with the project

Project Members

- Hung Che
  - Software engineering student, Programmer
- Calvin Kho
  - Software engineering student, Programmer
- Calvin Winner Liemena
  - Games Project student, Documentations, Artwork
- Claire Rochelmeyer
  - Games Project Student, UI design and Documentations
- John Kousoris
  - Games Project Student, Unity sound engineer
- Michael Murphy
  - Games Project Student, Artwork, Unity Sprite engineer.
- Nick Wain
  - Software engineering Student, Programmer.
- Steph
  - Games Project Student, Artwork, Unity Sprite engineer.

Below is a figure of our team structure approach to tackling this games project and roles, however we are not strictly following that structure due to our circumstances however, it can be used for potential project takeover/overhauls.

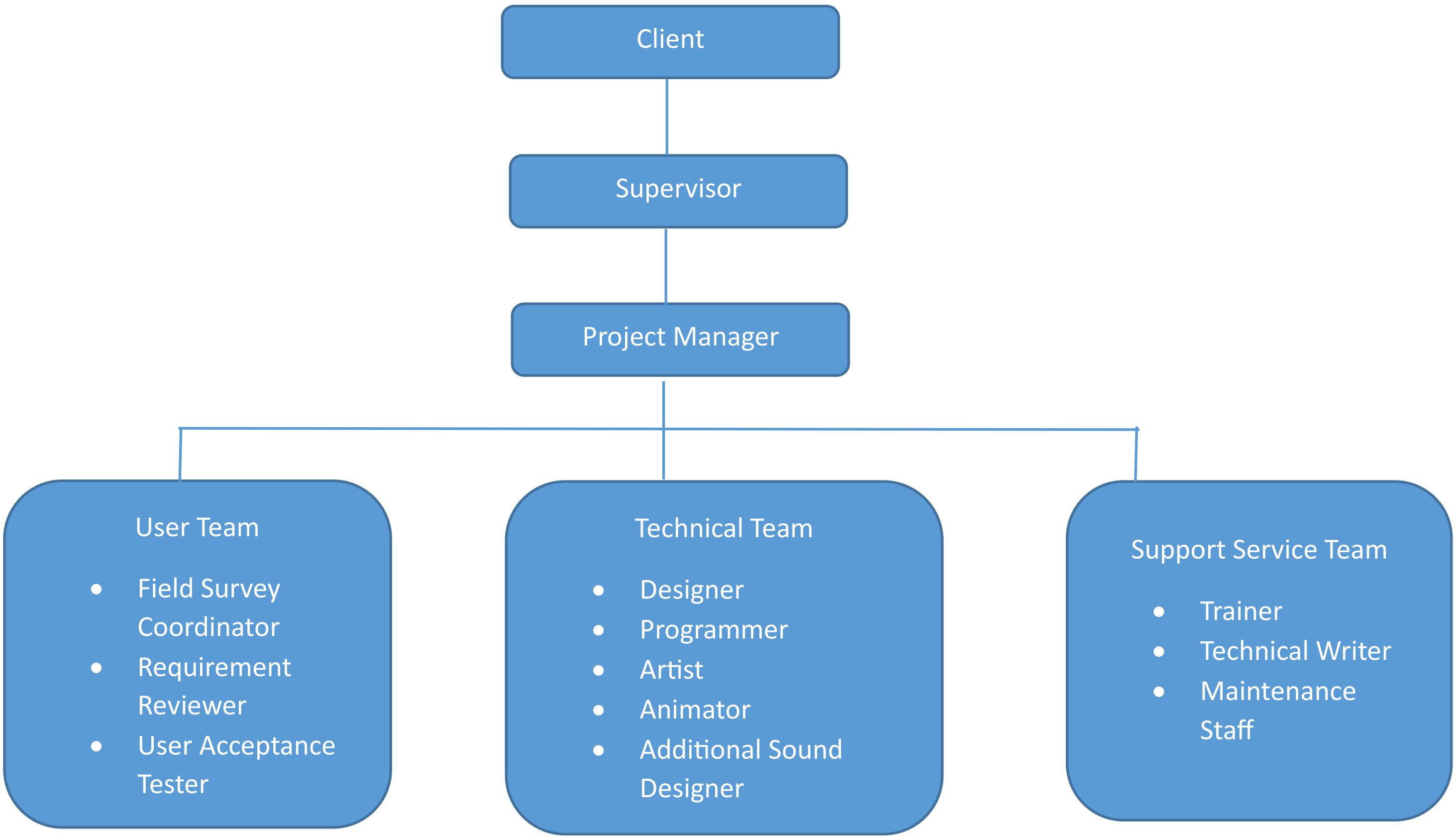


Figure 1: Project Team Structure

Roles		**Responsibilities**
Client		• Key link between the project management team and the organization's executive management • Approves any changes to the plan, scope or timeline • Works with project manager to resolve project issues
Supervisor		• Provide Project Management Methodology support • Provide sample templates and deliverables as needed
Project Manager		• Manage the project team • Review and approve all project deliverables • Ensures business and functional decisions are made • Escalate issues to sponsor when necessary • Have responsibility for managing and tracking the detailed sub-plan for their teams.
User Team	Field Survey Coordinator	• Collect and gather information for system requirements by conducting survey and interviews
	Requirement Reviewer	• Review the system requirements and ensure all the system requirements are met with business requirements
	User Acceptance Tester	• Conduct user acceptance testing and collect users' feedback about the system
Technical Team	Designer	• Design the structure of the game
	Programmer	• Write code and implement the system and database based on the design
	Artist	• Sketches ideas for the game worlds, characters and objects that going to be develop in the game • Design the levels of the game, themes and shape the look of the game
	Animator	• Portray movement and behavior of the character in the game
	Additional Sound Designer	• To create a desired effect or mood in the game
Support Service Team	Trainer	• Provide training for the client's staff on how to use the system
	Technical Writer	• Engage in technical writing and produces technical documentation
	Maintenance Staff	• Maintain the system and ensure the system is working well

2. Terms of Reference

2.1 Goals

To present an exciting 2D Android Games to public gamers. Our target audience would be from the age group of 20+, targeted at more strategic gamers who are generally older.

2.2 Objectives

The goals for this project would be:

- To present an exciting and challenging Game to play in short bursts.
- To provide customers/players the leaderboards
- A Suitable Work-Breakdown-Structure (WBS), Gantt chart to represent work down and a schedule
- Project networking diagram showing dependencies and critical path for the project
- Software development modelling
- A discussion of risks associated with the project and risk management

2.3 In Scope

- The software is a 2D mobile game, android platform.
- The application would be a prototype that is functional and plays as intended but not intended for quality testing.
- The game will be a single player game however, multiplayer interactions such as leaderboard is available.
- The game will provide a varied amount of replay value due to random generation
- UI: Adding fully navigated introductory level
- UI: Players select destination rather than moving tile-by-tile
- UI: Adding a visually represented stamina bar
- Art: Adding lighting, texture maps, and particle effects to the game's art
- Art: Improve player sprite, font, tiles, icons and panels aesthetics
- Art: Add animations for UI and overall animation improvement
- Gameplay: Add Remove Tile feature with a stamina cost
- Gameplay: Improve enemy movement and apply similar ruleset as the player
- Gameplay: Discard equipment feature
- Gameplay: Allow trapping of enemies

2.3.1 Out of Scope

- The game would not have handcrafted levels
- The game would have audio
- The game would only support portrait mode

2.4 Critical Success Factors

- Cozy game development environment
- A fixed computer lab provided specially for the game development
- At least 5 hours given for the use of the computer room
- Multiple PC Screen, desktop and basic games instruments provided
- Plug installed are more than enough to use

2.4.1 Objective success factors

Touch and drag interactions – Y/N

Clear visuals displayed on mobile display sizes of 4.0"+ - Y/N

Tutorial to how to play the game – Y/N

Number of control overlaps for different actions – 0

Screen rotation is fixed to portrait – Y/N

Adhering to android platform's ordinances such as call received – Y/N

2.4.2 Requirement success factors

- The game is able to touch and drag interactions
- Clear visuals are able to displayed on mobile screen size of 4.0
- Provide guidance of playing the game
- Fixed portrait screen rotation
- Adhering to android platform's ordinances
- Number of control for different overlap actions

2.5 Acceptance Criteria

Customer Expectations

1. The game save when the user stop playing and resume appropriately.
2. The game allowed users to play for short periods of time 2-5minutes and stop and resume later to continue playing without causing the player to be confused/disconnected from where they last stopped and when they resume.
3. The leaderboards cover two or more of the following categories.
4. High score, Amount of Tiles, Amount of moves total, Amount of steps (player moves) total points earned and lost.
5. The project provides suitable UML diagrams of **ALL** key classes and components of the application.
6. The customer's expectation of the game is to be able to understand how to play it via a tutorial and be able to play the game without any hiccups (bugs). Also customers are usually experienced with apps already on the market and thus may make assumptions and expect similar interactions to be played or controlled similarly.
    - To be able to play the game
    - Easily exit the app
    - Automatically saving progress
    - Know exactly what to do
    - Not to be confused by overlapping controls

Client Acceptance

The client expectations of the game are similar to the customers however the client is also inclined to see that the project/game was produced accordingly to design briefs and models.

- Adherence to design briefs and models
- Documentations of the process

3. Establishment

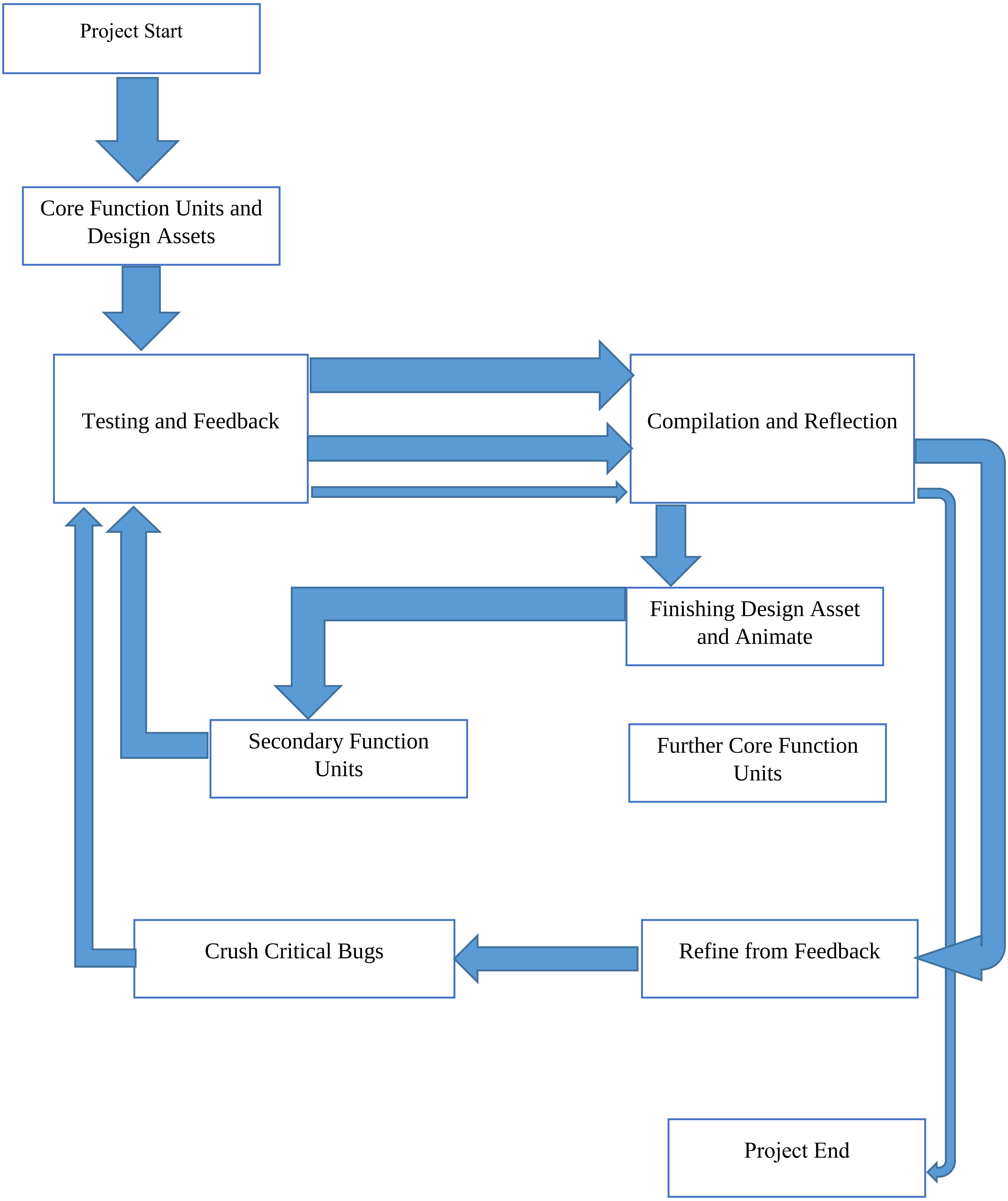
3.1 Process, Procedures and Standards

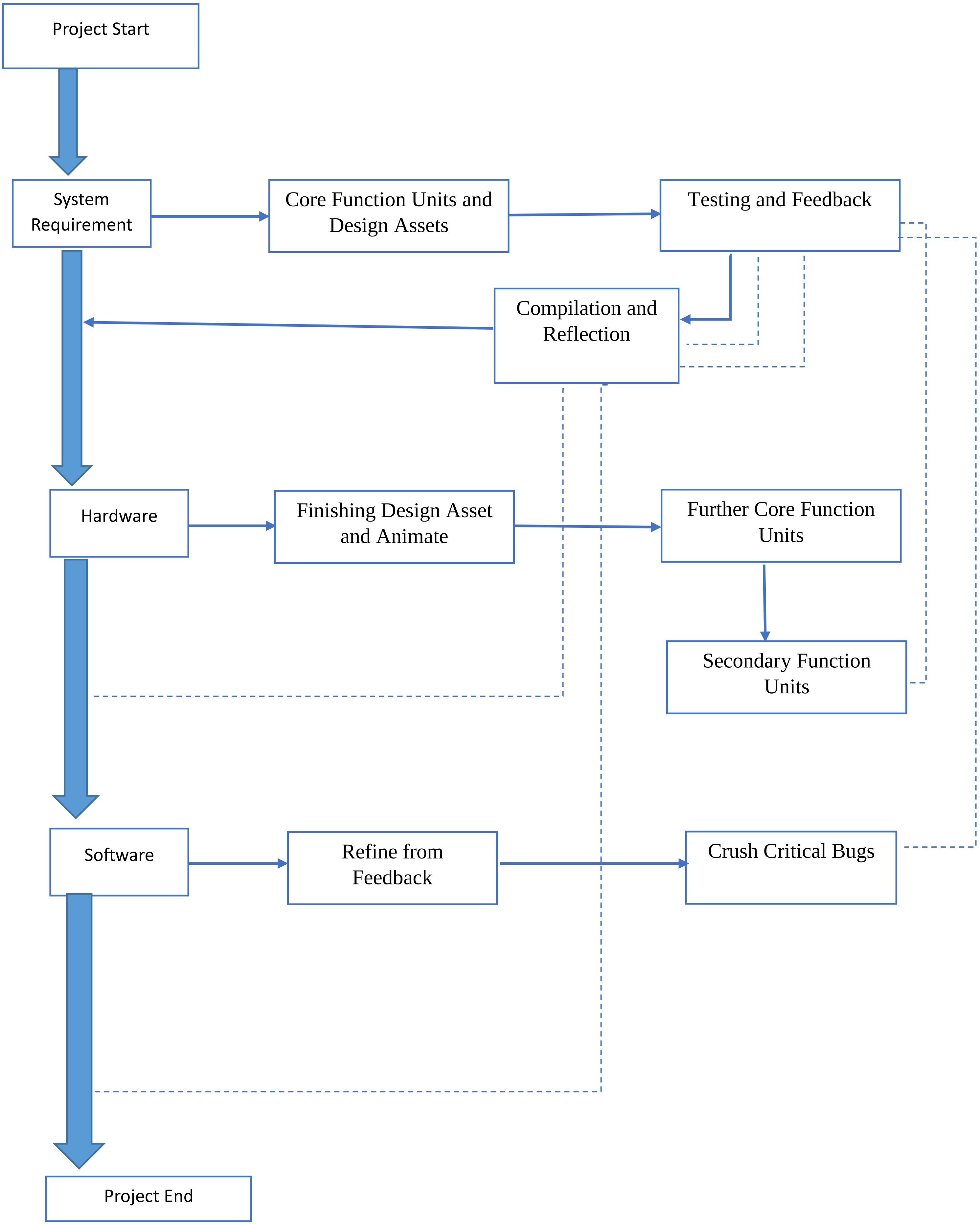
We will be approaching this project using a User-Centre Design the general outline is in the table below.

Job No.	Description	Immediate predecessor	Impact	Benefits
A	Project Start	-	-	-
B	Core Function Units and Design Assets	A	High	To ensure the final developed app match client requirements and satisfaction
C	Testing and Feedback	B	High	To ensure the app is working well with expected result and make correction based on the feedback given
D	Compilation and Reflection	C	Medium	To present a fully functional, 0 bugs app to public and get reflection from public to make improvement on the app To increase the satisfaction and numbers of use from users
E	Finishing Design Asset and Animate	D	Medium	To provide a better visual effect to attract public to install the game and increase app's value in the market
F	Further Core Function Units	E	Low	To provide freshness of the app to the users
G	Secondary Function Units	F	High	To provide a high quality and updated version kind of app
H	Testing and Feedback	G	High	To ensure the app is working well with expected result and make correction based on the feedback given
I	Compilation and Reflection	H		To present a fully functional, 0 bugs app to public and get reflection from public to make improvement on the app To increase the satisfaction and numbers of use from users
J	Refine from Feedback	I	High	To learn from mistakes and make improvements to make the app better
K	Crush Critical Bugs	J	Medium	To prevent problems of software couldn't support the features going to include in the app
L	Testing and Feedback	K	High	To ensure the app is working well with expected result and make correction based on the feedback given

M	Compilation and Reflection	L	Low	To present a fully functional, 0 bugs app to public and get reflection from public to make improvement on the app To increase the satisfaction and numbers of use from users
N	Project end	M	-	-

User-Centered Design


Project Start
Core Function Units and Design Assets
Testing and Feedback
Compilation and Reflection
Finishing Design Asset and Animate
Secondary Function Units
Further Core Function Units
Crush Critical Bugs
Refine from Feedback
Project End


Project Start
System Requirement
Core Function Units and Design Assets
Testing and Feedback
Compilation and Reflection
Hardware
Finishing Design Asset and Animate
Further Core Function Units
Secondary Function Units
Software
Refine from Feedback
Crush Critical Bugs
Project End

Usability Testing Procedure

2nd Iteration		
No	**Description**	**Explanation**
1	Core Function Units and Design Assets	To identify all the requirements required and figure out the way to code such kind of features out by using c#
2	Testing and Feedback	To understand the needs and excitement from the public by doing research which could increase the app market value once being published
3	Compilation and Reflection	Compile all the works done for the app into a prototype to test it and see the result and respond
3rd Iteration		
1	Finishing Design Assets and Animate	Completion of the design and animation of the interface of the app to make it looks like an real workable and playable app
2	Further Core Function Units	Expand the core function units about the app to update the app looks advance
3	Secondary Function Units	Having research to get more ideas of upgrading the app by adding more functions and challenges
4	Testing and Feedback	Having a test of the prototype created to ensure all the functions added are functioning as expected
5	Compilation and Reflection	Compile all the works done for the app into an updated prototype to test it and see the result and respond
4th Iteration		
1	Refine from Feedback	Make arrangement and improvement based on the feedback received to increase the satisfaction from the public in the next testing
2	Crush Critical Bugs	Edit and fix the error found in the app during the testing

		period to make the app run smoothly without any bugs
3	Testing and Feedback	Having a test of the prototype created to ensure all the functions added are functioning as expected
4	Compilation and Reflection	Compile all the works done for the app into an updated prototype to test it and see the result and respond

3.2 Project Environment

To develop this project, we have been given an opportunity to use a room which specially for developers to develop games and group meeting. Our group being given a time slot on every Tuesday from 8 in the morning until 1 in the afternoon. The equipment's and instruments in the room provided are more than enough for a game development team of 6 to use under cosy working environment.

3.3 Project Team Training Requirement

Project team will be using Unity engine to develop the game as both the developers are inexperienced we decided to have one undergo research and training with using Unity.

Familiarizing ourselves with C# and learning to the general coding conventions of C#.

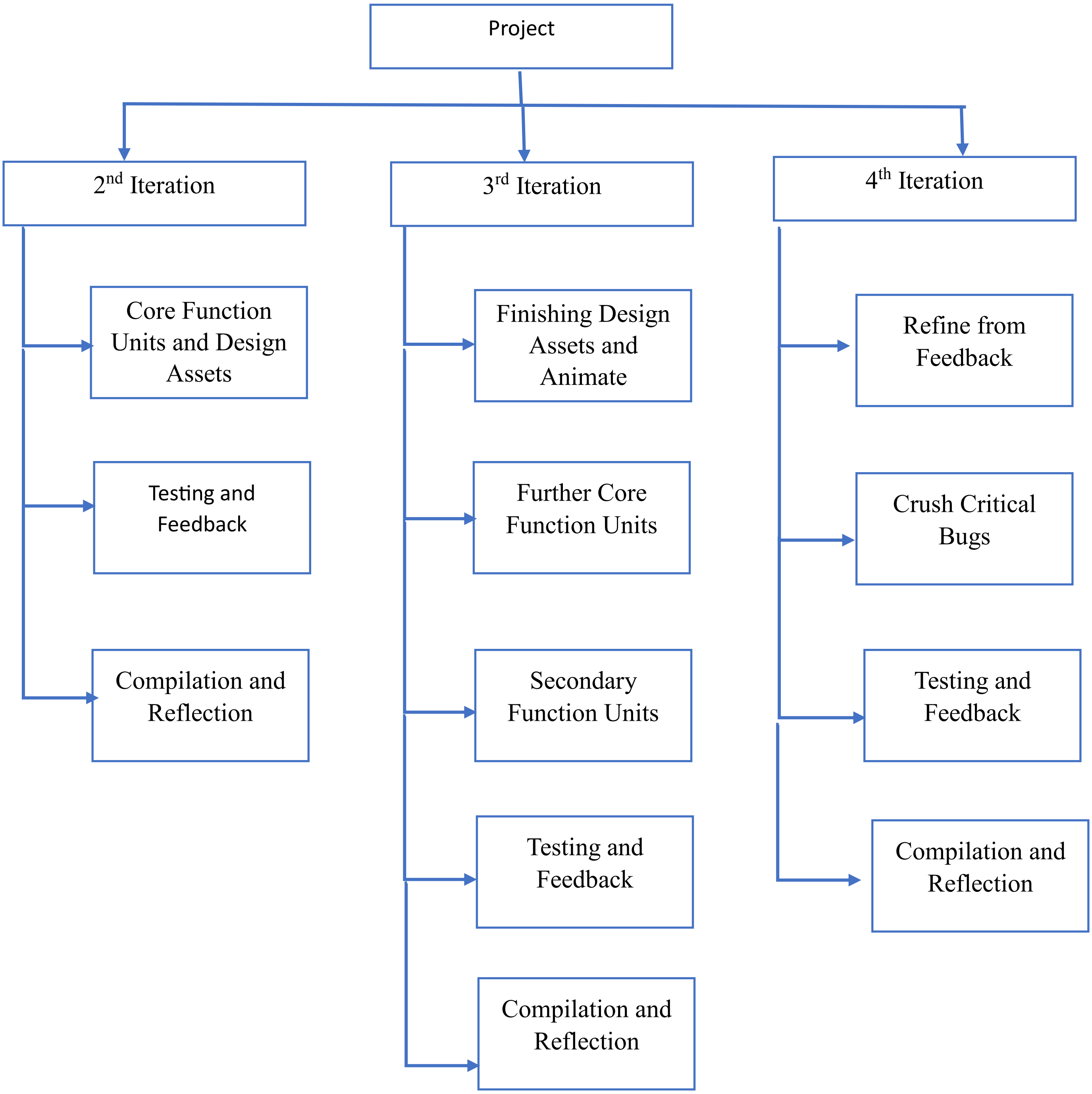
4. Activity, Deliverables and Capital Resources

4.1 Deliverables

Job No.	Description
A	Project Start
B	Refined Player Character Asset
C	Tiles Asset with Collision
D	Trap, Enemy Character and Equipment Assets
E	Core Function Unit Script
F	2nd Iteration Prototype Built
G	Animated Character, Enemy and Trap Assets
H	Refined Core Function Unit Script
I	Secondary Function Unit Script
J	Sound Draft and Feedback
K	3rd Iteration Prototype Built
L	Trailer Poster
M	Alpha Prototype Built
O	Project end

Table 1: Deliverables

4.2 Activities and Tasks



Project
2nd Iteration
Core Function Units and Design Assets
Testing and Feedback
Compilation and Reflection
3rd Iteration
Finishing Design Assets and Animate
Further Core Function Units
Secondary Function Units
Testing and Feedback
Compilation and Reflection
4th Iteration
Refine from Feedback
Crush Critical Bugs
Testing and Feedback
Compilation and Reflection

Figure 3: Work Breakdown Structure

5. Resources

5.1 Organization and Structure

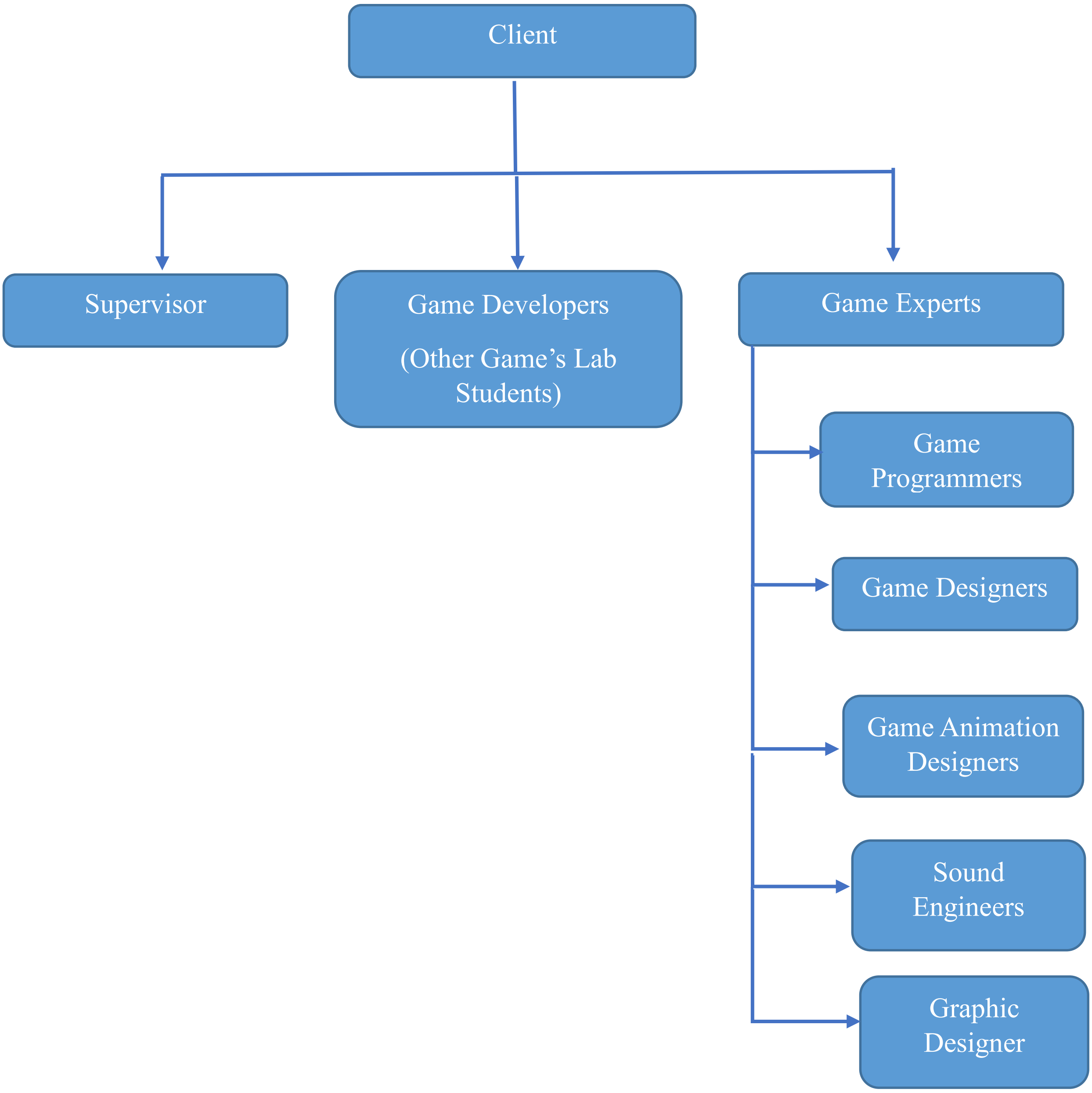


Figure 4: Organization and Structure in acceptance testing

6. Risks

6.1 Identification of Appropriate Risks

Brainstorming

Group member verbally identify risks that provide the opportunity to build on others' ideas. It is essential that all participants are familiar with the topics and relevant documentation is provided. A note-taker should be appointed to capture the ideas that are being discussed. All members must participate. Brainstorming can also be used during planning to generate strategies, possible causes of risk and other areas of impact.

Working group

Working group is a way to analyze a particular area or topic in a discussion process to identify risks that may not be too obvious to the risk identification group. The working group is usually a separate group of people working a particular area within the project that is conducting the risk identification.

Documented knowledge

Collecting documented information and data from past project on a particular area. This will provide insights to the risk on the particular area of concern.

Experiential knowledge

Experiential knowledge is information that has been collected by a person through their past project experience. Enable project team to identify the risk that commonly occurs during software development.

6.1.1 Controllable Risks

Risk of system failure

If the software was development poorly and is implemented into business operations without extensive testing, the chances of facing system failure are very high. Unexpected bugs may be found and the software may not perform as well as intended.

Poor usability

Poor interface and not user friendly. Employees will not be able to use the system as they do not know how to operate the software.

Budget risk

A cost overrun involves unexpected cost incurred in excess of budgeted amounts due to an underestimation of the actual cost during budgeting. Unpaid overtime, management efforts and risk management costs are examples of cost that are usually not accounted for during budget estimation.

Excessive schedule pressure

Irrational schedules and excessive schedule pressures are a common risk among large projects. This is because that the top management does not have the proper notion about the workload and activities involved in the project.

Requirements inflation

As the project progresses, it becomes apparent that the specification of features and requirements that contains conflicting requirements were not identified at the beginning of the project emerge will threaten cost estimates and disrupts timelines.

Scheduled risk

Software development, due to its intangible nature, it is difficult to estimate the schedule. Besides that, project schedule tends to get slip when project tasks are not completed within the allocated time.

Creeping user requirements

This risk is considered to be one of the killer risks in software development. The continuously changing of requirements of the client will cause the budget estimation, time schedules and project management to fall apart.

Unable to manage human resources

In software development, it usually requires a team of people collaborating together for an amount of time working on the project. Conflicts may arise among team members and if project manager are unable to solve these issues, it may jeopardize the project. The project team may also lack proper communication which may raise dissatisfaction among team members.

Gold plating

Refers to continuing to work on a project or task well past the point such as adding unnecessary features to the program which are not required where the extra effort is worth the value it adds.

Inaccurate metrics

In earlier days of software engineering, lines of code were used as the common of software projects. Measuring the project is bound to be inadequate and inaccurate.

6.1.2 Uncontrollable Risk

Economic conditions

If there is an economic inflation, client may request cancellation of software development due to insufficient funds. Sponsors may also retract their sponsorship causing project to come to a halt. Apart from that, economic inflation may give rise to the general level of prices of goods and services causing the increase expenditure of the project.

Changing technology

Before the software could be launch, the technology changes suddenly and the product that is developed may become obsolete as a better substitute is now available in the market.

Political instability

The political instability of the country may cause delays in the completion of the project. It may result is delays in hardware delivery or employees not available during the period of the project development.

6.2 Risk Analysis

Controllable risk	**Likelihood to occur**	**Cause**	**Impact**
System Failure	High	No extensive testing done before implementing software in business operations.	Disrupts business operations. The business operations will not be able to function optimally and this will cause the company to face great lose.
Poor Usability	Medium	No proper commenting and documentation. No user manual provided. Poor interface design.	Employees are unable to use the software. Software becomes obsolete if no people can operate it to do its task.
Budget Risk	High	Failure to take into account of overhead cost.	May cause the project to be cancelled due to insufficient funds.
Excessive Schedule Pressure	Low	Irrational client demands. Constant changing of user requirements	Poor result of end product due to hurried implementation.
Requirements Inflation	Low	Did not identify all the requirements of the project during the planning phase	Threaten cost estimates and disrupts timelines.
Schedule Risk	High	Schedule slippage. Failure to identify complex functionalities and time required to develop those functionalities.	Failure to complete project in time.
Creeping User Requirements	Low	Ever changing of requirements and demands by clients	Affects timelines and cost estimation.
Unable to Manage Human Resource	High	Poor communication among team members. Rise of conflicts in the team.	Poor teamwork and team cohesion. Task not completed properly. Delays in task completion.
Gold Plating	Low	Unnecessary addition of functions not required by clients	Cost developing product overweighs the value of product
Inaccurate Metrics	Low	Measuring scope by lines of codes	Lack of standardization, difficulty in determining source code size during requirements. Inability to measure software defects, specification and non-code deliverables.

Table 2: Risk Analysis

6.3 Risk Management

There should be people in the team whose primary activities are connected with risk management. Since there are many risks connected to software development project, the organization should formally define risk interpretation and ranking policy. This will help to separate the important from the unimportant risk in order to address the important risk as fast as possible. Project team members should continuously trace and monitor actions connected with risks.

Risk	Risk Management
System Failure	• Phased conversion method. Gradual implementation of new system overtime replacing the old system. Allows for each operation of the new system to be testing and any problems can be sorted out easily. • Extensive testing before releasing the software.
Poor Usability	• Provide proper documentation and user manuals for the system. • Identify the user requirement and user domain. Provide additional training for employees.
Budget Risk	• Compile a realistic estimate of cost with the team. Make sure that it covers all the project phases and activities and that it contains sufficient contingency. Account for all people cost as well as materials, overhead cost, hardware and software. Achieve a baseline budget. • Have a well-defined organizational procedures and practices to cost the project • Track all expenditures
Excessive Schedule Pressure	• Enlighten the top management on the task involved in the project and present an estimated time
Requirements Inflation	• Constant involvement of clients and developer throughout the development of software
Schedule Risk	• Developed a timeline and make sure that each task is completed in the allocated time • Using the Program Evaluation and Review Technique (PERT) to estimate the duration for each task
Creeping User Requirements	• Introducing proper change procedure • Documenting the impact of changing requirements in terms of function points, time and cost implication • Get a signed-off document of all the requirements of the project
Unable to Manage Human Resource	• Recruit the necessary expertise • Ensure that resources are available where team members can collaborate and share knowledge • Project manager must be good in social skills and able to solve any conflicts between team members. Interact with team members on a regular basis.
Gold Plating	• Ensure that the developers understand all the requirements clearly.
Inaccurate Metrics	• Using function points as a measuring metric rather than using lines of codes

Table 3: Risk Management

7. Schedule

7.1 Delivery Phases

7.1.1Overview

Job No.	Description	Immediate predecessor	Estimated Time (man-hours)
A	Project Start	-	0
B	Core Function Units and Design Assets	A	15
C	Testing and Feedback	B	15
D	Compilation and Reflection	C	45
E	Finishing Design Assets and Animate	D	20
F	Further Core Function Units	E	15
G	Secondary Function Units	F	35
H	Testing and Feedback	G	35
I	Compilation and Reflection	D	50
J	Refine from Feedback	I	30
K	Crush critical bugs	J	35
L	Testing and Feedback	K	25
M	Compilation and Reflection	L	25
T	Project end	M	0
	Total Hours		345

Table above shows the overview of the numbers of tasks we are going to work with during the process of developing this project. There are 12 to be completed tasks created purposely to ensure that we could achieve every tasks and produce a quality project. The 12 tasks listed will be divided into 4 phases and critical path being created to ensure we could complete the tasks in each phase under the expected completion of time.

7.1.2 Delivery Phase 1 < 2nd Iteration > (from A – D)

The 3 tasks we are going to complete in Phase 1- 2nd Iteration which includes Core Function Units and Design Assets, Testing and Feedback, Compilation and Reflection.

During the delivery of phase 1, there are 3 tasks we are going to achieve in order to proceed to phase 2 as every task created have its own dependencies and the remaining tasks would be delayed if we didn’t complete one of the task on time.

Basically, communication is the main issue happening in every team while working on a project. Our team also realized that this will be the most important issue which could affect the quality of the project as this phase needs plenty of communication by providing ideas or comments to produce the quality design and functions of the app.

In order to solve the communication issue, our team decided to create Trello, slack and GitHub for dividing the tasks to every member equally and stay in touch with the progress of the work they are working with.

7.1.3 Delivery Phase 2 < 3rd Iteration > (from E –I)

There are 5 tasks stated in Phase 2- 3rd Iteration. They are Finishing Design Assets and Animate, Further Core Function Units, Secondary Function Units, Testing and Feedback, Compilation and Reflection.

Interface and functions are the most important parts of an app as users will not get interest if the app created only containing a basic and simple interface. Our team will focus on the implementation of the design, graphic and functions of the app in this phase to allow users enjoy playing our app and won't feel bored or does not feel stimulate while playing with the app.

To avoid this issues, our team decided to do more research and plays with the apps that created by others to get ideas and figure out on how to added all those functions which are interesting to our app.

7.1.4 Delivery Phase 3 < 4th Iteration> (from J-T)

There are 4 tasks going to complete in Phase 3 which are Refine from Feedback, crush critical bugs, Testing and Feedback, Compilation and Reflection.

Our team will focus on the errors or bugs that found in the apps and from the reflection from testers. Feedback plays an important role while the development of the app ended soon as our team could make improvement to enhance the app by upgrading every of the core functions to increase user's satisfactions and interests.

In order to solve this issue, our team decided to ask for testers to test on the prototype created and get reflection from them so that we could know what we missed up and added in features that majority of the users nowadays expecting for.

7.2 External Dependencies

Job No.	Description	Immediate predecessor	Estimated Time (man-hours)
1	Project Start	-	0
2	Rework on tiles, gam icons, UI font and stamina bar	1	80
3	Fix Player's animation and Screen Resolution	2	60
4	Adjust dealing to remove tile duplication in a single hand	3	30
5	Adjust discard cost, enemy damage and stamina replenishment	4	30
6	Add fully navigated introductory level	5	60
7	Implement enemies pathing rules	6	140
8	Add new texture maps, lighting, particle and sound effects	7	45
9	Implement remove tiles feature	8	70
10	Re-design art assets to ensure cohesion and avoid any cultural offence	9	50
11	Implement player pathfinding	10	140
12	Add game analytics	11	45
13	Add additional effects and sound to maximize feedback	12	25
14	Optimizing Facebook and Twitter	13	25
15	Building Marketing Plan and Press Kit	14	25
16	Creating media for marketing	15	25
17	Design Website and Publish	16	90
18	Implement new type of enemy pathing	17	80
19	Rebalance and tweaking based on feedback	18	15
20	DVD Case Concept Art	19	15
21	Rebalance and tweaking based on feedback	20	15
22	Rebalance and tweaking based on feedback	21	15

23	Final rebalance and tweaking based on feedback	22	15
	Total Hours		1055

7.3 Assumptions

The time estimates are all given in man-hours, with the following assumptions:

- Each day a person works from 8am to 5pm, with 1 hour off for lunch, which is 8 hours per working day
- Total of 6 people are assigned to the project, which are divided into 2 teams according to their roles in the project

The support services are not included in the time for project completion except staff training, as maintenance and troubleshooting would take place after the project's completion.

7.3.1 Project Management

Assuming a team of 2 people is assigned to this section.

7.3.1.1 Project Manager Body of Knowledge

Assuming 1 project manager is managing all the staff.

7.3.2 Software

Assuming a team of 4 people is assigned to this section.

7.3.3 System Test

Assuming a team of 2 people is assigned to this section.

7.3.4 Installation

Assuming a team of 2 people is assigned to this section.

7.3.4.1 System Installation

There are 2 terminals, 1 terminal in HQ.

- Total terminals = 2 terminals

Assuming each terminal would take 1 hour each to install

- Installing one terminal = 1 hour
- Installing 2 terminals = 2 hours

7.3.5 Support Services

Assuming a team of 6 people is assigned to this section.

7.3.5.1 System Maintenance

Assuming 2 people from the team are assigned to maintain the software for 1 year, and they conduct weekly maintenance sessions of 3 hours together:

- 2 people x 3 hours per week = 6 hours’ weekly
- Each year has 52 weeks = 312 hours

7.3.5.2 System Troubleshooting

Assuming 2 people from the team is assigned to handle technical support, and both spend 2 hours a day handling technical requests:

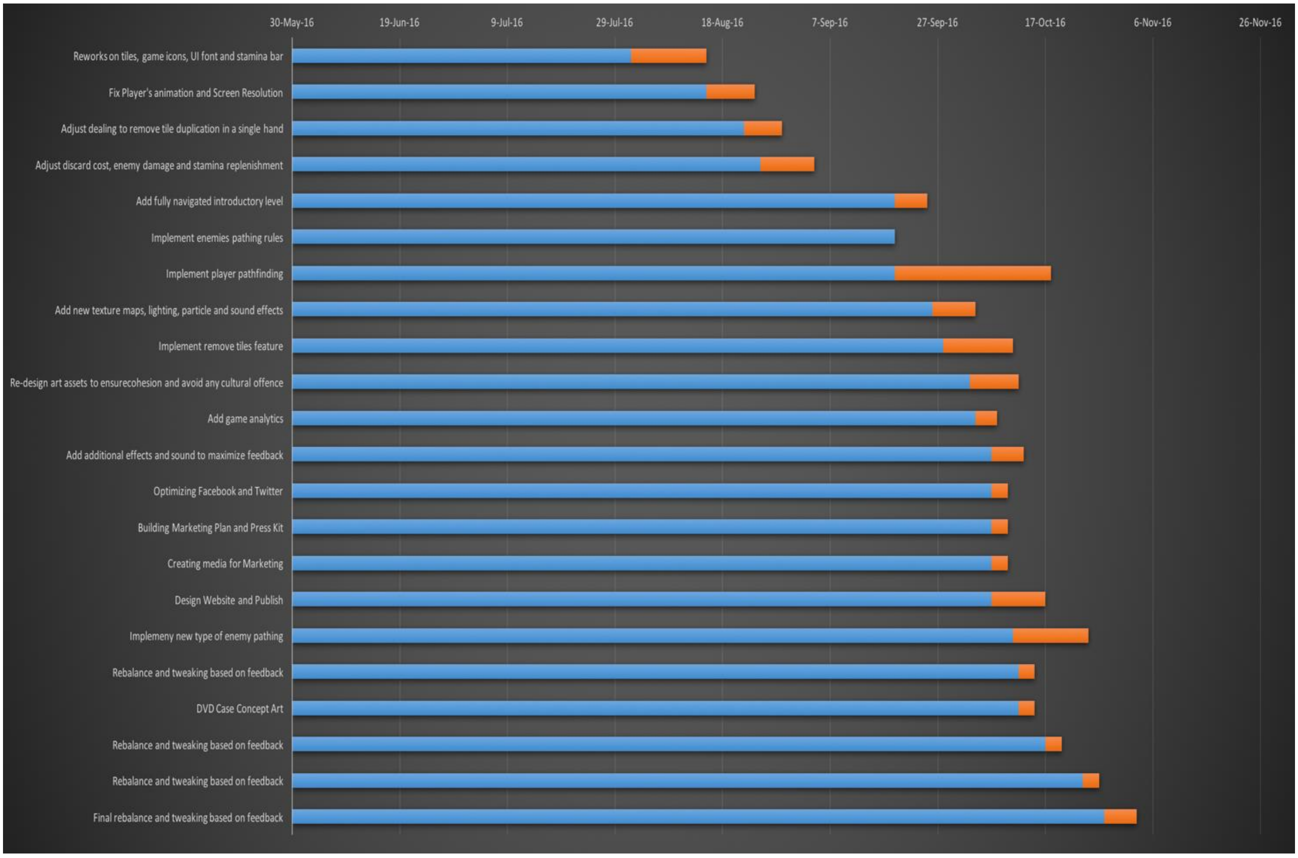
- 1 week = 10 hours
- 1 year has 52 weeks = 520 hours

7.3.5.3 Project Team Training

Assuming 2 people from the team is assigned to project team training, and they conduct 4 hours per day for a 5 days training

- 4 hours x 5 days x 2 people = 40 hours

7.4 Project Time Line (Gantt Chart)



8. Budget

Time allocation for each member will be 8 to 8.5 hours per week and their pay is AUD 30 each hour
The time completion for the first stage will takes 11 weeks to complete
There are 8 members in the group
Programmer spend 8.5 hours per week
Each calendar month = 4 weeks
Income per hour = AUD 30
Total hours spend in a month= 8.5x4=34
Total income per week = 255
Total income in a month= 34 x 30= 1020

Designer and Sound Engineer spend 8 hours per week
Each calendar month = 4 weeks
Income per hour = AUD 30
Total hours spend in a month= 8x4=32
Total income per week=240
Total income in a month= 32x30 = 960

Roles	Expected Hours Spend Per Week	Actual Hours Spend Per Week	Total Cost per Month	Total Cost for 11 Weeks	Total Cost for 52 weeks
Programmer (Team Leader)	10	8	960	2640	12480
Programmer	10	8.5	1020	2805	13260
Programmer	10	8.5	1020	2805	13260
Programmer	10	8.5	1020	2805	13260
Designer	10	8	960	2640	12480
Designer	10	8	960	2640	12480
Designer	10	8	960	2640	12480
Sound Engineer	10	8	960	2640	12480

Time allocation for each member will be 8 to 8.5 hours per week and their pay is AUD 30 each hour for current year
The pay will increase to AUD 32 in future years
The time completion for the fully developed games will takes 52 weeks to complete
There are 8 members in the group
Programmer spend 8.5 hours per week
Each calendar month = 4 weeks
Income per hour = AUD 32
Income per week= AUD 272
Total hours spend in a month= 8.5x4=34
Total = 34 x 32= 1088

Designer and Sound Engineer spend 8 hours per week
Each calendar month = 4 weeks
Income per hour = AUD 32
Income per week=AUD 256
Total hours spend in a month= 8x4=32
Total = 32x32 = 1024

Roles	Expected Hours Spend Per Week	Actual Hours Spend Per Week	Total Cost per Month	Total Cost for 11 Weeks	Total Cost for 52 weeks
Programmer (Team Leader)	10	8.5	1088	2992	14144
Programmer	10	8.5	1088	2992	14144
Programmer	10	8.5	1088	2992	14144
Programmer	10	8.5	1088	2992	14144
Designer	10	8	1024	2816	13312
Designer	10	8	1024	2816	13312
Designer	10	8	1024	2816	13312
Sound Engineer	10	8	1024	2816	13312

	Current Year		Future Years		Total Life	
	Budget	Actual	Budget	Actual	Budget	Actual
Programmer (Team Leader)	14000	12480	15000	13312	30000	25792
Programmer (Calvin Kang Chiak Kho)	14000	13260	15000	14144	30000	27404
Programmer (Hung Che)	14000	13260	15000	14144	30000	27404
Programmer (Nick Wain)	14000	13260	15000	14144	30000	27404
Designer	14000	12480	15000	13312	30000	25792
Designer	14000	12480	15000	13312	30000	25792
Designer	14000	12480	15000	13312	30000	25792

Sound Engineer	14000	12480	15000	13312	30000	25792
Total Labor Expenses	84000	77220	90000	82368	180000	159588

Table 4: Complete Project Life Estimates (Hours)

Time allocation for each member will be 8 to 8.5 hours per week and their pay is AUD 30 each hour for current year
The time completion for the first stage will takes 11 weeks to complete
The time completion for the fully developed games will takes 52 weeks to complete
There are 8 members in the group
Programmer spend 8.5 hours per week
Each calendar month = 4 weeks
Income per hour = AUD 30
Total hours spend in a month= 8.5x4=34
Total = 34 x 30= 1020

Designer spend 8 hours per week
Each calendar month = 4 weeks
Income per hour = AUD 30
Total hours spend in a month= 8x4=32
Total = 32x30 = 960

Roles	Expected Hours Spend Per Week	Actual Hours Spend Per Week	Total Cost per Month
Programmer (Team Leader)	10	8.5	272
Programmer	10	8.5	272
Programmer	10	8.5	272
Programmer	10	8.5	272
Designer	10	8	256
Designer	10	8	256
Designer	10	8	256
Sound Engineer	10	8	256

	February	March	April	May	Total Hours
Team Leader	1020	1020	1020	1020	34
Programmer	1020	1020	1020	1020	34
Programmer	1020	1020	1020	1020	34
Programmer	1020	1020	1020	1020	34
Designer	960	960	960	960	32
Designer	960	960	960	960	32
Designer	960	960	960	960	32
Sound Engineer	960	960	960	960	32
Total Labor Expenses	7920	7920	7920	7920	264

Table 5: Current Financial Year Estimates

9. References

Hamilton, G, Byatt G, Hodqkinson, J & Terril, P (2012), *A risk management implementation*, CIO, viewed 1 April 2014, <http://www.cio.com.au/article/427457/risk_management_implementation/>

Ian Sommerville, Software Engineering (9th Edition), Addison-Wesley, 2009, Chapter 1 - 3.
Bob Hughes and Mike Cotterell, Software Project Management (5th Edition), Wiley, 2009, Chapters 1 and 3

Majeed, M (2012), Human Resource Issues in Project Management, *PM*, viewed 28 March 2014, <http://project-management.com/human-resource-issues-in-project-management/>

Pressman, R (2005), 'Lecture 3', *HIT3309 Software Project Scheduling*, Learning materials on Blackboard, Swinburne University of Technology, viewed 2 April 2014

Roger S. Pressman, Software Engineering - A Practitioners Approach (7th Edition), McGraw-Hill, 2010, Chapter 26.

Rowe, C 2006, Risk identification methods, *ClearRisk Manager*, viewed 31 March 2014, <https://manager.clearrisk.com/Resources/RiskTool/Risk_Identification_Methods_-_12_Types>

Shinde, V (2007), Types of risks in software projects, *Software Testing Help*, viewed 28 March 2014, <http://www.softwaretestinghelp.com/types-of-risks-in-software-projects/>

Shri, AK 2010, Software risks and control, viewed 28 March 2014
Risk control – controllable and uncontrollable risk 2009, *BestPractice.com*, viewed 1 April 2014, <http://www.best-practice.com/risk-management-best-practices/financial-risk-management/risk-management-risk-control-%E2%80%93-controllable-and-uncontrollable-risks/>