

# SCC210 Group Project

D1 Report Writing

Jidong YUAN

# SCC210 Assessment

Deliverable	Title	Weighting %	Submission
<b>D1</b>	Design Report	40	One per group
<b>D2</b>	Final Presentation	30	Group
<b>D3</b>	In-Lab Demonstration	20	Group
<b>D4</b>	Self-Reflective Report	10	One per person

# D1: Design Report

This report documents your work during term 1 towards the initial ideas for your project and the design of the project your group has chosen to implement. **The maximum number of pages is 20** and anything after page 20 will not be marked. The design report should include:

- Various project ideas from each group member
- Task list throughout the term (for weeks 3-8)
- Desired project features
- Software requirements
- Design principles that influenced the design of your project
- Set of rules for your project
- Project plan for implementation of your project in term 2, including milestones and deliverables for weeks 5-12
- Acceptance tests
- Activity network with critical path, and Gantt chart

# DI Design Report

- Introduction and Project Ideas. [20% marks]
- Project Design (for chosen project) [40% marks]
- Software Engineering [15% marks]
- Implementation Plans [15% marks]
- Structure and presentation of the report [10% marks]
- Appendices

# 1. *Introduction and Project Ideas.* [20% marks]

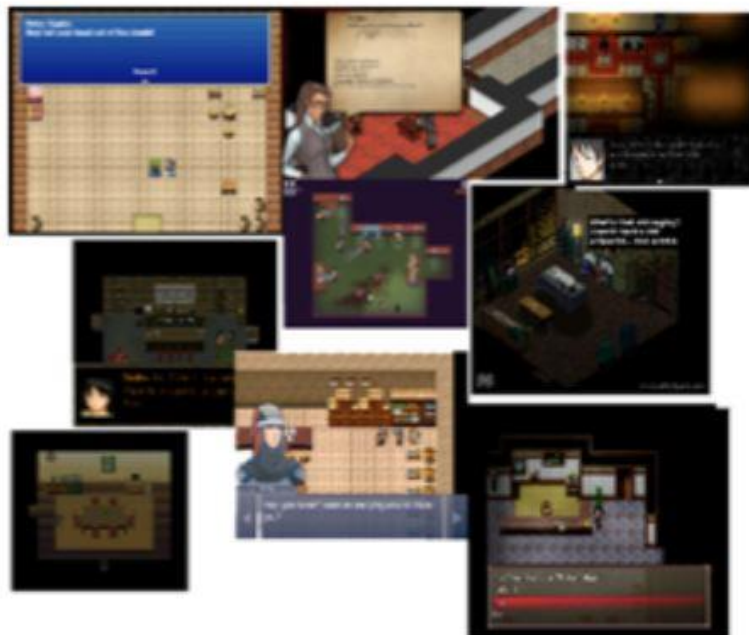
- Brief summary of the main tasks of the project. Various project ideas from each group member.

## Game Design Influences

During the idea stage of our development we all came up with vastly different ideas taking inspirations from many different games, before we whittled them down and tried to tie them together. For our final game idea we decided to go along with a murder mystery set in the 18<sup>th</sup> century we took this inspiration from a variety of sources. Some of which influenced our decisions more than some. The most obvious non-game influences are the classic murder mysteries of Sherlock Holme, one of the most well-known detectives set in the Victorian era.

This would become our setting for the game, where the story would unfold inside an 18<sup>th</sup> century mansion that the player would have to solve some mystery while collecting clues. But basing our game upon a book series would not pan out very well for a game. After looking at other detective games that were already in the market (see Fig.1). We quickly noticed that they all used a top down approach similar to that seen in the more popular Zelda and Pokémon games using tiled based systems that would allow the player to explore the world freely.

Fig.1. Other detective games



## 2. *Project Design* [40% marks]

- 2.1 Desired project features
- *“The background of the game is there is a person, the Nanny, who has gone missing from the manor. The player takes on the role of a detective, helped out by his mysterious assistant, who will traverse the hall, solving puzzles, finding clues and trying to work out what has happened to the Nanny! The first, main, part of the game involves traversing the manor house and talking to the characters, solving any puzzles, finding clues and assuming and analyzing the clues for what happened to the nanny.*
- *At the end of the game, based on the clues collected and facts found out from the rest of the game, the player need to make a decision: which character “dunnit” and the game will have function to “accuse” one of these characters. The ending of the game will depend on the decision the player makes, as well as (hopefully) different decisions the player made throughout the whole game. (Different plot paths). “*

- Game Features and Game play Elements
- The game should perform in real time; meaning that user input and feedback is provided on an 'as-it-happens' basis. For example, the user may be prompted for input, is able to provide that input immediately, and a response is given immediately (such as character movement).
- The game should provide a 'click-and-start' button which allows the user to decide when to initiate the game. The button will launch the game and all of its components immediately upon clicking.
- Obstacles in the game will "appear and disappear", meaning that as the player moves new objects will randomly be 'spawned on the border of the frame, as the ones that are no longer visible on the frame are removed. This will allow each instance of the game world to be completely random and 'limitless'.
- There will be collision detection between the player character and the in-game objects/obstacles. This collision detection will serve as the basis for the game, where 'survival' is the objective. If the player 'collides' with obstacles it will trigger an in-game event.
- The game will be played from a 2-dimensional top-down perspective, where the frame of the game and thus the player, is looking 'down' at the character and its surroundings.



## 2.2 Design principles that influenced design of project.

# Some Different Design Principles (Shigeru Miyamoto – Nintendo – Super Mario etc)

- Start with a **simple** concept
  - “running, climbing, jumping”
- **Design around the computer’s limitations**
  - Character wears overalls so easier to see arms move
  - Wears a hat because don’t have to have hair
  - Has mustache because couldn’t draw nose and mouth
- **Minimize the player’s confusion**
  - What to do should be **clear without consulting a manual**
- The importance of **play testing**
- **Incorporate a smooth learning curve**
- **Accommodate all skill levels**



# Sid Meier (Civilization)

- **Player should have fun**, not designer, programmer, or computer
- Great game-play is a stream of **interesting decisions that the player must resolve**
- The inverted pyramid of decision making (**have few decisions to deal with first**, and then let them multiply until the player is totally engrossed)
- **Put the player in his dreams**, where he or she is the hero

# So What Principles Make A Good Game?

1. Pursuing and achieving goals (challenges)
2. Interactivity
3. Feedback about position relative to goals
4. Interesting choices required to achieve goals
5. Consistency and fairness
6. Avoid repetition

But the central principle is FUN – of some kind..

## 2.3 Set of rules for project

*e.g.* Game Project

- Constitutive Rules – define the game
- Operational Rules – shape game play
- Implicit Rules – unwritten but accepted by players

# Rules – a stolen example...

- **Constitutive**

The game is fit for a single player experience at any one time. The player should complete the game in a single sitting in order to be as involved with the story as possible. Players negotiate the maps in an ordered manner, interacting with characters and forming assumptions on which they will decide the conclusion to the game. The players receive visual feedback of the progress of their character through the game universe via interactive text conversations with AI, visual notification of the character moving and occasional audio stimuli. The game progresses through a storyline, and the player wins the game if he successfully chooses the correct person as the culprit at the end of the game, the fact that he won or lost will be reflected in the ending of the game. The concept of choosing the culprit will be introduced via dialogue during the game, as will most other fundamental constitutive rules, other than ones that can be gained via experience (such as the implicit rule that you can't walk through walls). The player will then enjoy credits at the end of the game.

- **Operational**

When the game begins, players will have the chance to select from a number of graphical character options. Players navigate the maps by use of keyboard controls, with extra keyboard-only functions to advance chat options, interact with NPC's and discover clues. The player will have a number of interfaces on screen to remind him of controls, display certain other pertinent information such as chat boxes and NPC display pictures. Should the player enter a puzzle, a separate GUI will show and the player can interact with this via keyboard input also. Clicking with the mouse on the game in any way, except to manage the window (Maximise/minimise) will have no effect.

The player will have a range of areas of the house that he is allowed to visit at any point in the game. This will be enforced via chat options with the other characters and will stop the player being able to traverse the game in an incorrect order (i.e the player will not be able to find out information that he should not until thresholds in the game are reached. At these thresholds, new chat options will appear on relevant characters).

It is not possible for the players to leave the rooms specified in the game, nor enter any room that has no enterable door.

- **Implicit**

The game will be launched by conventional methods and will be stored on disk. The game will behave on disk as any regular executable program would and will be able to be copied, renamed, deleted and moved as the user wishes. The game will be protected by copyright as it is the intellectual property of the authors. Playing the game will not have a permanent effect on the user's PC, and will be replayable.

### 3. Software *Engineering* [15% marks]

3.1 Software requirements – functional and non-functional requirements as well as specific software requirements

# Functional Requirements

1. The game shall allow user input.
  - 1.1. The game shall be only controlled with the keyboard.
2. The game shall be displayed on a visual display.
  - 2.1. The display shall be a fixed resolution.
  - 2.2. The display shall be allowed to full screen.
    - 2.2.1. This shall include windowed mode.
3. The game shall be a desktop application.
  - 3.1. The game shall be windows only.
4. The game shall have multiple screens for the user to navigate.
  - 4.1. The game shall have a main menu screen.
    - 4.1.1. The main menu screen shall navigate to the game screen.
    - 4.1.2. The main menu screen shall navigate to the instructions screen.
    - 4.1.3. The main menu screen shall navigate to the credits screen.
  - 4.2. The game shall have a game screen.
    - 4.2.1. The game screen shall navigate to every screen apart from the credits.
  - 4.3. The game shall have an instructions screen.
    - 4.3.1. The instructions screen shall revert to the user's previous screen.
  - 4.4. The game shall have an end screen.



# Non-Functional Requirements

## Specific Non-Functional Requirements

- 1) The system shall have a failure rate of no more than 3%.
- 2) The obstacles should fill between 20-30% of the screen in a normal state (eg. no power-ups or power-downs).
- 3) The system should pass 98% of all tests in order to be considered fully 'playable'.
- 4) The game shall start within 5 second of pressing the start button (on average).
- 5) The game shall receive an average overall feedback score of 7/10 from beta testers.
- 6) The game shall have a maximum input response time of 25ms.
- 7) The game shall run at least 30 frames per second.

## 3.2 Acceptance tests

### Acceptance Tests

Test identifier	<b>AT_001 - Character movements</b>
Test items	This test verifies that the player can move the character using the keyboard
Input required	The user pressing the keys: <ul style="list-style-type: none"><li>• W to move forward</li><li>• A to move left</li><li>• S to move backwards</li><li>• D to move right</li></ul>
Output required	The character moving in the correct direction on the screen

Test identifier	<b>AT_002 - Character interactions</b>
Test items	This test verifies that the player can make the character interact with objects
Input required	The user pressing the E key when near a door
Output required	The character opening the door

## Accessibility tests:

These accessibility tests are detailed analysis of the function of single tasks, and will be completed and analysed until all of the tests we laid out passed. This will make up a large part of our testing strategy.

Id	Given	When	Then	Passed
1	The program	They start the program	The window should be displayed	
2	The program	They start the program and a window appears	A Splash screen will be shown for 2 seconds	
3	The Main menu	The credits button is pressed	The credits screen is displayed	
4	The Main menu	The instructions button is pressed	The instructions screen is displayed	
5	The Main menu	The Play button is pressed	The user is taken to the start of the game	
6	The main menu	The exit button is presses	The user is displayed a prompt	
6.1		And the user presses the yes button	The program exits	
6.2		Or the user presses the No button	The prompt is closed	
7	The instruction screen	The back button is pressed	The user is taken to the previous state	
8	The End screen	5 seconds has passed	The user is taken to the credits	
9	The credits screen	The user presses a button	The user is taken to the main menu	

## 4. *Implementation Plans* [15% marks]

- 4.1 Task list throughout the term (for weeks 3-8)
- 4.2 Project plan for implementation of the project in term 2, including milestones and deliverables from weeks 5-12

# Task List

## Week 3

- i. Introduction to the group and the task presented.
- ii. Establish a group leader and then have the group leader assign appropriate group members to roles.
- iii. All group members to help decide on the type of game we are looking to implement.

## Week 4

- iv. Decide on a plot for the game
- v. Discuss the general idea of the game and how the game's levels will be implemented
- vi. The group leader to assign group members to be responsible for different sections of the Design Report.
- vii. The group leader must create an attendance sheet to track the attendance of group members for record.
- viii. The group leader to decide on a name for the sprite and name of the game being created.

# Contingency Planning

## Contingency Planning

It is crucial that a contingency plan is put into place for this project for the following reasons and more:

- 1) One of the group members become unavailable for any reason
- 2) The need to share data outside of meetings
- 3) A group member who has been assigned a task has failed to complete the task
- 4) Hardware or software becomes damaged for any reasons ranging from theft or natural disaster
- 5) Time constraints which limit the amount of work the group has planned to implement

These five reasons listed above, if occurred, could have a dramatic impact on our project and outcomes. There may be more issues we possibly encounter during our work together, and if so the team leader should call a meeting to discuss them. The listed reasons however are the most likely and most important to this project.

### Option One - Likelihood: Low

In the event of option one, where a group member becomes unavailable, an emergency meeting must be called to discuss distribution of said group member's work load. A member being 'unavailable' refers to the member leaving university, falling seriously ill, changing major programmes, etc. This does not simply mean the member has failed to show up to meetings with no reasoning, or without sending apologies in advance.



#### 4.1.1 Risk Assessment

With a project such as this that spans over a relatively long period it is necessary to produce a risk analysis. Considering these factors will allow for a more precise contingency plan.

The risk analysis shown below, describes the possible risks that we may encounter as a group and how they might be compensated.

Nº	Risk	Description	Likelihood	Impact	Contingency Plan
1	Group member is unable to participate	A group member is effectively no longer 'part of the group', for the rest of the project.	Medium	Low	With the group being large for this size project the impact will be relatively minimal. However, the task delegation will need to be updated.
2	All work is corrupted	All programming work is no longer accessible, and will have to be remade from the beginning.	Low	High	As the work will be delegated in terms of classes, there will be multiple backups on member's machines locally and on the shared environment

## RISK ASSESSMENT & CONTINGENCY

Risk	Probability	Effects	Solution	How	Who	Result
Members of the group are unavailable (ill, busy, drop out)	Moderate.	Serious.	Spread out workload equally to remaining members.	Give each member 1-2 additional tasks.	Group project members.	Project ongoing smoothly.
The time required to finish a task is underestimated.	High.	Serious.	Speed up production.	Members allocate more time into the project.	Group project members.	Project finishes as scheduled.
Members of the group fall behind with coursework for other modules.	Moderate.	Serious.	Provide assistance and give useful advice.	Use Facebook to contact members.	Group project members.	Member catches up with other modules and progresses on group project
Important files get lost/wiped.	Low.	Catastrophic.	Back-up data in multiple locations.	We will utilize collaborative cloud services i.e. Dropbox + Github.	Group project members.	Proceed group project with back-up files.



# Milestones

## List of tasks (milestones)

#	Description	Dependencies	Acceptance test	Start Date	Duration ( days )	End Date
Task 1	Design of the interface.			12-Jan	3	15-Jan
Task 2	Sound track.			25-Jan	5	30-Jan
Task 3	Sprites ( the design of walls, characters, weapons, etc. )			15-Jan	6	21-Jan
Task 4	Implementation of menu interface as well as actual game interface, layout canvas on which the game will be developed essentially.	T1	AT_005	15-Jan	6	21-Jan
Task 5	Mount sprites onto walls, characters, etc. Make them animated.	T4		21-Jan	5	26-Jan
Task 6	Take input from the user and map it onto controls.	T4	AT_005	21-Jan	2	23-Jan
Task 7	Monsters on the map.	T4		21-Jan	3	24-Jan
Task 8	Maze design, walls implementation.	T4		21-Jan	7	28-Jan
Task 9	Switch between weapons in the inventory.	T6	AT_004	23-Jan	1	24-Jan
Task 10	Switch between items/attack modes.	T6	AT_004	23-Jan	1	24-Jan
Task 11	Save game / load game.	T6		24-Jan	2	26-Jan
Task 12	Movement of character.	T6	AT_001	26-Jan	10	5-Feb
Task 13	Level select.	T6	AT_010, AT_011	24-Jan	2	26-Jan
Task 14	Collision detection for walls, monsters, etc.	T8, T7		28-Jan	7	4-Feb
Task 15	Lighting - light does not go through walls.	T9		4-Feb	7	11-Feb

# Deliverables

## Deliverables

<b>Deliverable description</b>	<b>Prerequisites</b>	<b>Deadline</b>
Usable menu and game interface	T1, T4	21-Jan
Movement of character and ability to learn how to play the game	UT_003, T12	5-Feb
Fighting against monsters	T7, T19, UT_004	15-Feb
Advancement through levels by killing boss	T19, T6, UT_001	19-Feb
Stats upgrade between levels	T23, UT_005	19-Feb
Storyline displayed between levels with ability to skip it	T18	31-Jan

# MILESTONES & DELIVERABLES

Milestones						
#	Description	Internal Dependencies	Acceptance Test	Start Date	End Date	Duration
1	Placing graphics/sprites on the screen.	-	Visual	21/01/2016	28/01/2016	7 days
2	Grid map implementation	-	Visual	11/01/2016	15/11/2016	4 days
3	Character movement	-	AT_004	21/01/2016	28/01/2016	7 days
4	Collision detection	2, 3	AT_004	18/01/2016	22/01/2016	4 days
5	Graphic map/level generator	1	Visual	11/01/2016	25/01/2016	14 days
6	Implemented the designed levels	1-5	AT_006	11/01/2016	25/01/2016	14 days
7	Implemented enemy guard AI paths	1-6	AT_007	22/01/2016	04/02/2016	14 days
8	Design/implement interface	-	AT_001, AT_004	11/01/2016	18/01/2016	7 days
9	Menu screens	-	AT_001	11/01/2016	18/01/2016	7 days
10	Character dialogue	-	AT_003	11/01/2016	25/01/2016	14 days
11	Game events - score, pickups	1-4	AT_005	23/01/2016	04/02/2016	14 days
12	Game events – Level progression, lives	1-6	-	11/01/2016	25/01/2016	14 days
13	Achievements	1-8	Visual [too specific for general test]	23/01/2016	04/02/2016	14 days
14	Saving and loading	1-6	AT_002	17/01/2016	24/01/2016	14 days
15	Fully playable	1-14	ALL	19/02/2016	26/02/2016	7 days

## Deliverables

Description	Deadline
Graphic/Sprite level generator displaying maps	25/01/2016
Movement of the character	28/01/2016
Guards following AI paths and interactions	04/02/2016
Useable menu screens and interface	29/01/2016
Level and story progression	25/01/2016
Final product	26/02/2016

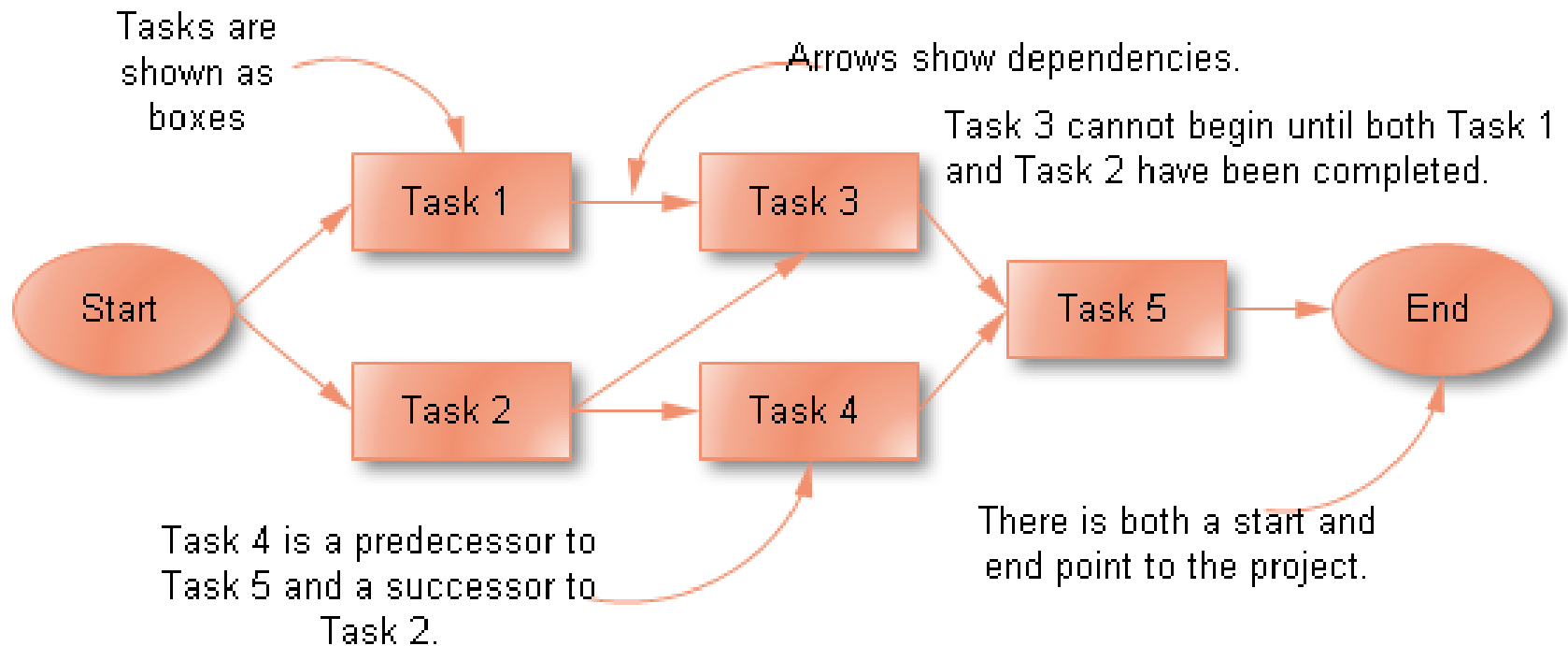
## 4.3 Activity network with critical path, and Gantt chart

- Activity network (activity graph): A graphical method for showing dependencies between tasks (activities) in a project.

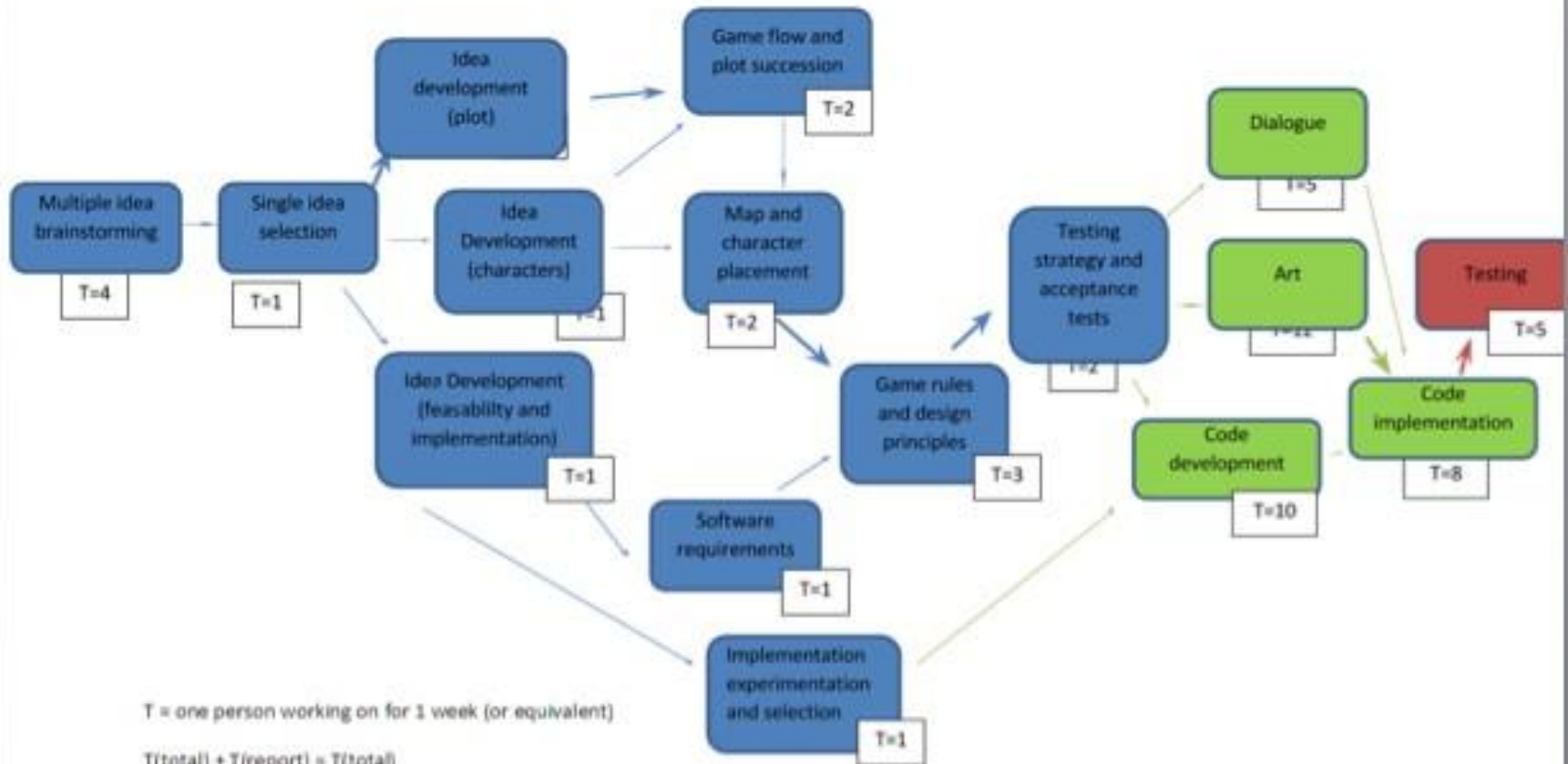
## 4.3 Activity network with critical path, and Gantt chart

- The critical path is the longest distance between the start and the finish of your project, including all the tasks, their duration, which gives you a clear picture of the project's actual schedule. Another term in the critical path method is earliest start date.
- A Gantt chart is a type of bar chart that illustrates a project schedule, named after its inventor, Henry Gantt (1861–1919), who designed such a chart around the years 1910–1915.

# Activity Network



# Activity Network



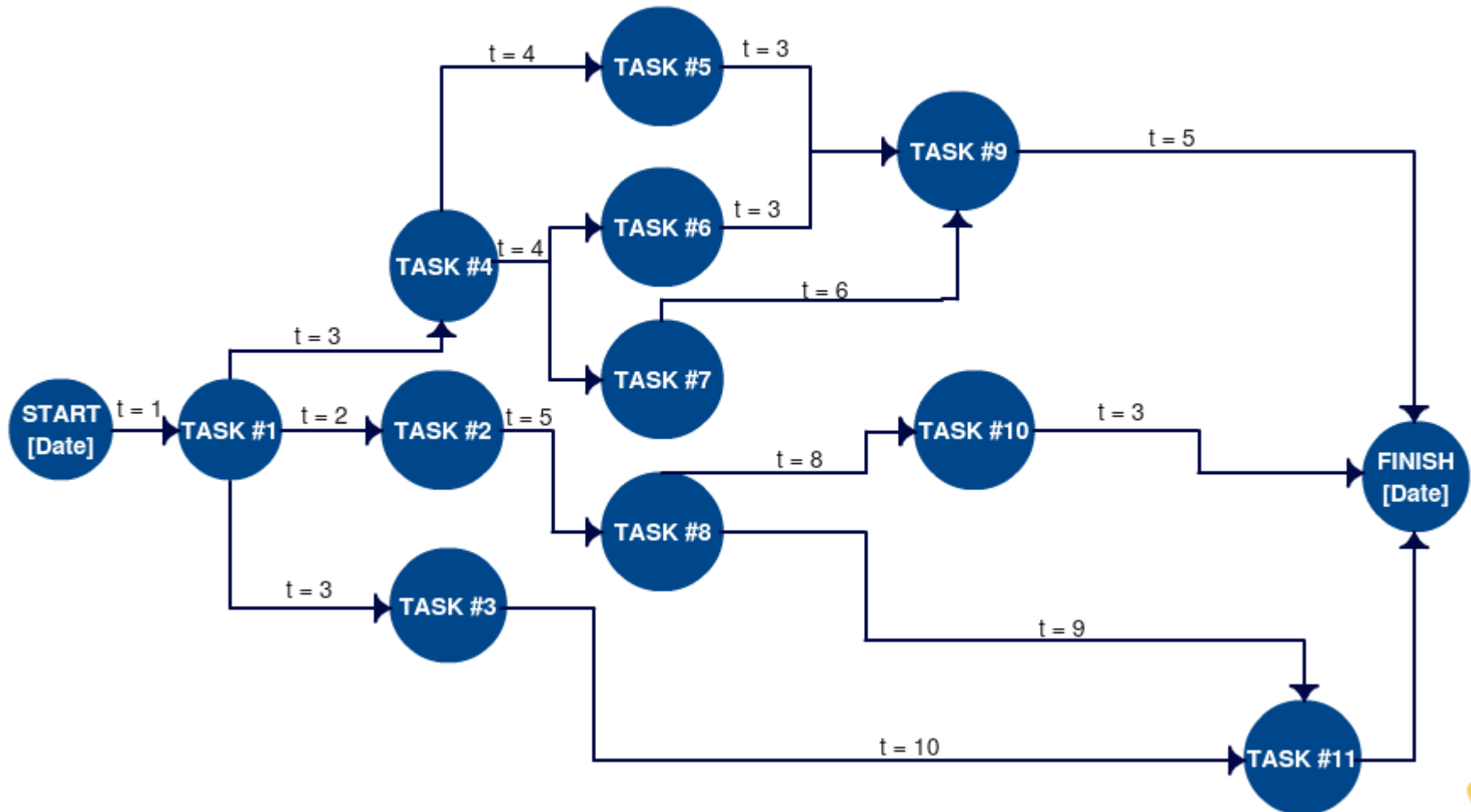
T = one person working on for 1 week (or equivalent)

$T(\text{total}) + T(\text{report}) = T(\text{total})$

$61 + 14 = 75$  (15 weeks x 5 people)

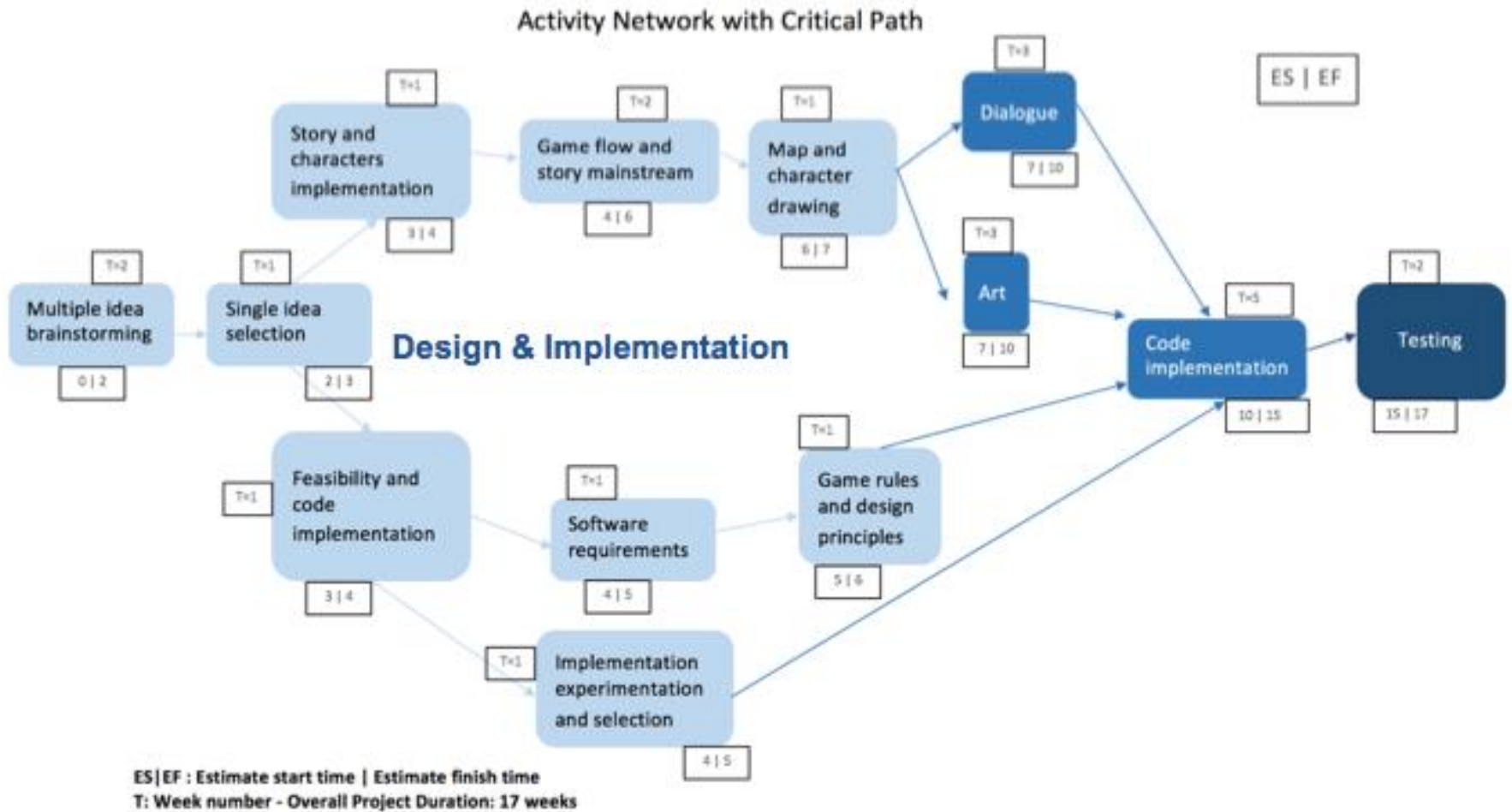
## Example of PERT Chart

\* t in days





# ACTIVITY NETWORK & CRITICAL PATH



# GANTT Chart

Gantt chart for tasks



This is a gantt chart of all of the tasks we allocated over the whole two weeks, with the tasks in week 10-13 simplified. Appendix article 3) is an extended chart for week 11-16. Important to note is our plan to test all the way through the process and that all dialogue, sprite sheets and character images need to be completed by week 14 for our plan to be successful.



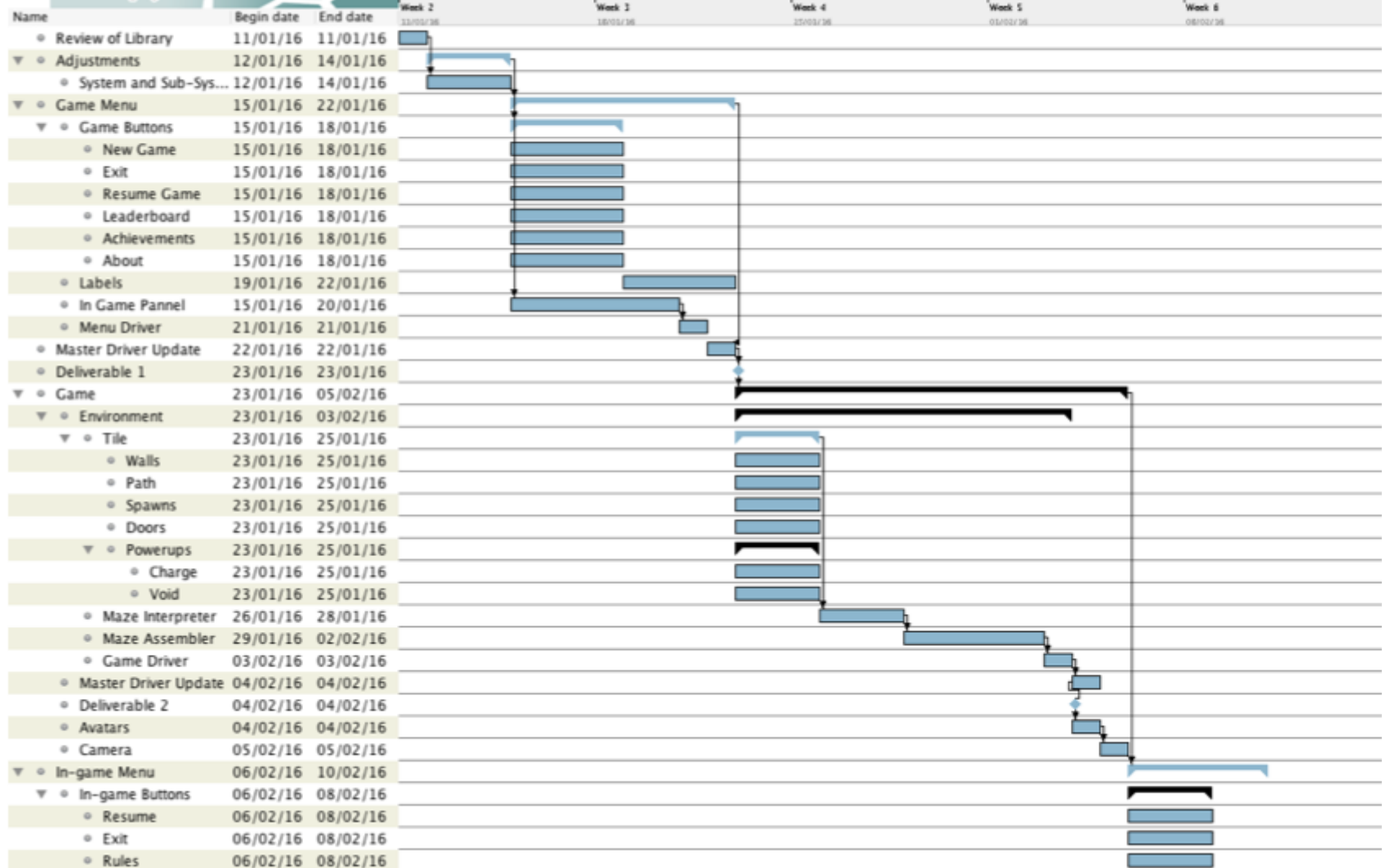
January 2016

Deliverable 1

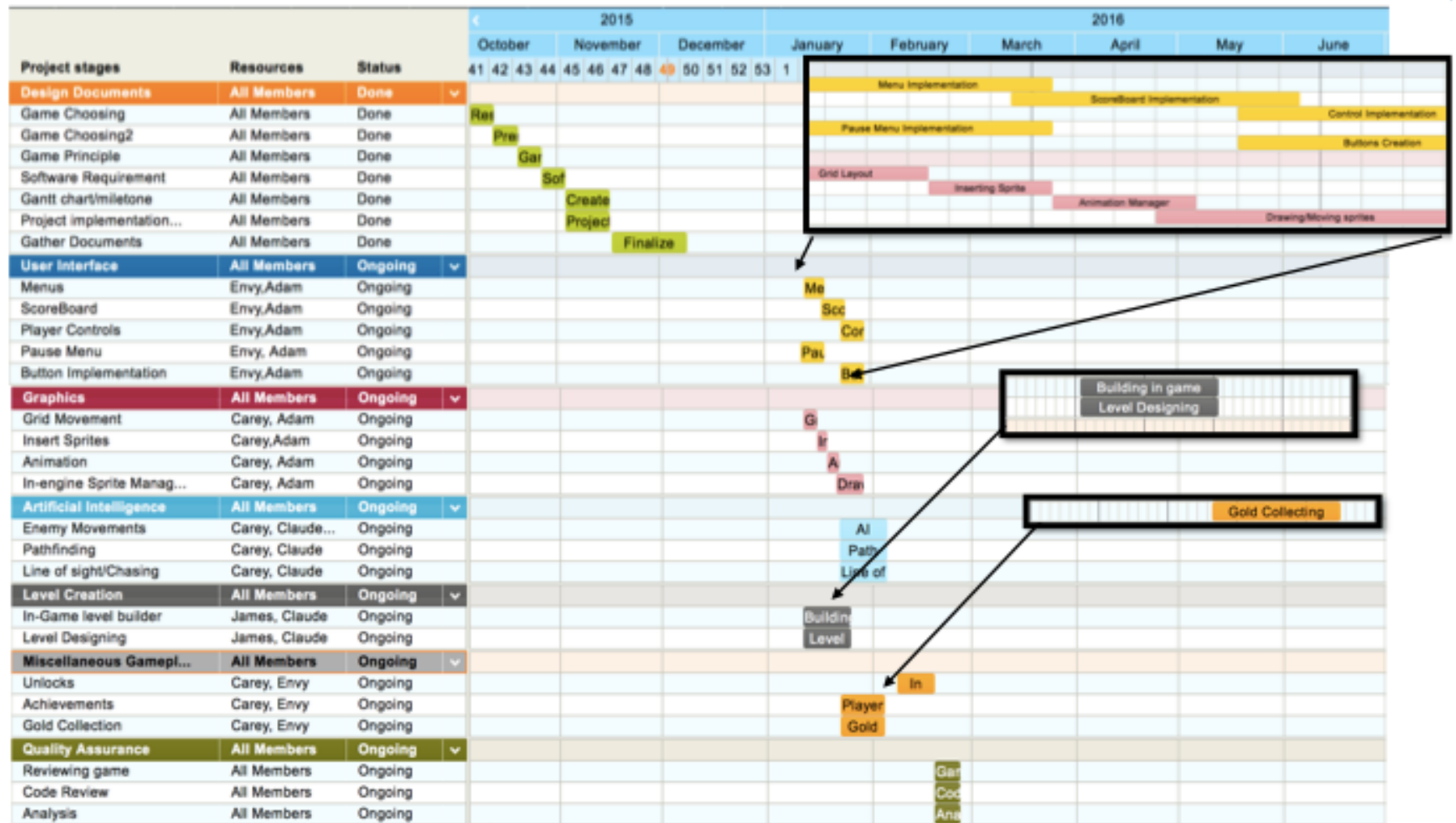
February 2016

Deliverable 2

Deliverable 3



# GANTT CHART



# 5. Structure and presentation of the report [10% marks]

- Quality aspects such as structure, language, coherence, formal issues (such as page numbers, sections with headings and introductions, table of contents, etc.), etc. will contribute 10% to your overall mark.
- When writing up your report, keep a few points in mind, i.e.:
  - The reader is the most important person
  - Keep the report within the 20 page limit (excluding title pages, table of contents, appendices etc.)
  - Organise the document for the convenience of the report user
  - All references should be correct in all details
  - The writing should be accurate, concise and unobtrusive
  - The right diagram with the right caption should be in the right place for the user
  - Summaries give the whole picture, in miniature
  - Reports should be checked for technical errors, typing errors and inconsistency
  - The reader is the most important person

# Contents



This is a short tutorial showing you how to create an automated table of contents in Microsoft Word.

## Contents

Chapter 1 – Creating a document .....	1
Chapter 2 – Formatting .....	1
Section A: Heading 1 types .....	1
Section B: Heading 2 Types .....	1
Chapter 3 – Table Styles .....	1
Chapter 4 - Summary .....	1

# Contents Sheet

## Contents

<b>Introduction and Project Ideas</b>	<b>3</b>
Introduction to the project including tasks to do	3
Games Ideas	3
Advantages in Game Choice	5
Reasons for Our Choice	5
Potential Difficulties	5
Our Decision as a Group	5
<b>Desired Game Features and Gameplay</b>	<b>6</b>
Character and plot	6
Levels and gameplay	6
Power-Ups	8
Game design principles that influenced the design of the game	8
Game Rules	9
<b>Software Engineering</b>	<b>10</b>
Software Requirements	10
Acceptance Tests	11
<b>Implementation Plans</b>	<b>13</b>
Task List	13
Contingency Planning	15
Project Plans	16
Splitting up programming tasks	17
Milestones and deliverables	19
Critical Path, Activity Path & Gantt Chart	20
<b>Appendix A: A Further Insight on Group Meetings</b>	<b>23</b>
<b>Appendix B: User Interface Storyboard</b>	<b>25</b>
<b>Appendix C: Logging an Interruption</b>	<b>26</b>
<b>Appendix D: Approval Documentation</b>	<b>27</b>

<b>Appendix E: Disapproval Documentation</b>	<b>28</b>
<b>Appendix F: Attendance Sheet</b>	<b>29</b>
<b>Appendix E: Potential Graphics</b>	<b>30</b>



# CONTENTS

## > Introduction

- > **Introduction**
- > **Initial Ideas**
- > **Final Design**

Page 3

Page 3

Page 4

Page ?

## > Game Design

- > **Backstory**
- > **Influences**
- > **Key Features & Gameplay**
- > **Level Structure**
- > **Game Principles**
- > **Game Rules**

Page 8

Page 8

Page 8

Page 9

Page 10

Page 11

Page 12

## > Software Engineering

- > **Software Requirements**
- > **Acceptance Tests**

Page 13

Page 13

Page 14

## > Implementation Plans

- > **(Design) Task List**
- > **Risk Assessment**
- > **(Development) Task List**
- > **Milestones & Deliverables**
- > **Gantt Chart**
- > **Activity Chart & Critical Path**

Page 1ti

Page 1ti

Page 16

Page 17

Page 19

Page 20

Page 21

## > Appendices (Visual Material)

- > **Appendix A: Game Desirables Mind Map**
- > **Appendix B: Level Designs**
- > **Appendix C: Prototype User Interfaces**
- > **Appendix O: Game Sprites**

Page 22

Page 22

Page 23

Page 24

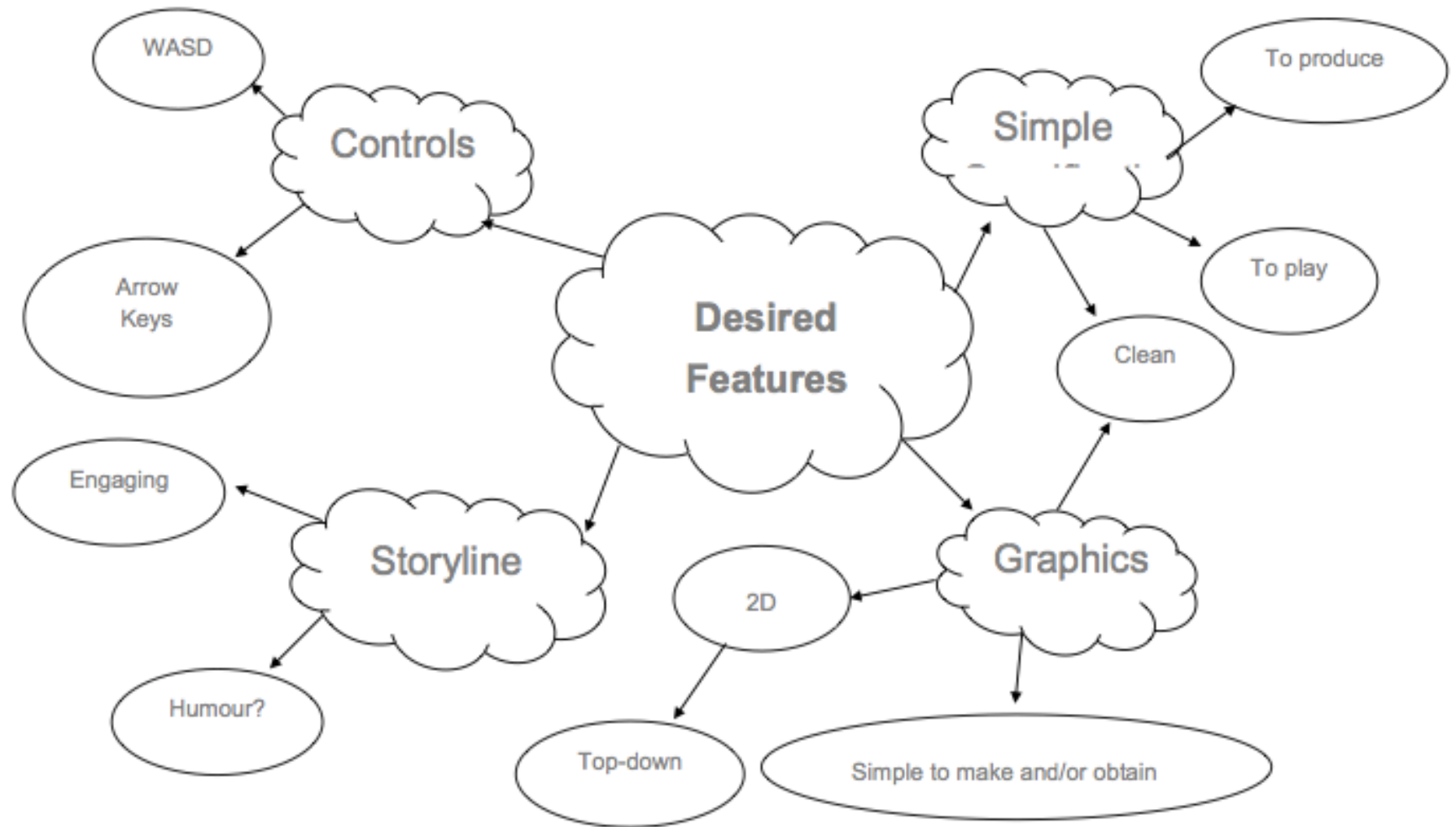
Page 2ti



## *6. Appendices*

- Working documents
- Milestone versions of task list, project plans, requirements study, architectures, etc.

## APPENDIX A – GAME DESIRABLES MIND-MAP



## Article 2: Detailed Gantt chart for weeks 11-17



## Desirability Questionnaire

Select 0 – 9 for each of the following questions:

### **Controls and Gameplay**

Fluidity of movement

Poor

Excellent

0 1 2 3 4 5 6 7 8 9

Control system

Poor

Excellent

0 1 2 3 4 5 6 7 8 9

Dialog interaction

Poor

Excellent

0 1 2 3 4 5 6 7 8 9

Fluidity of gameplay

Poor

Excellent

0 1 2 3 4 5 6 7 8 9

Select ONE per question

Question	Strongly Disagree	Disagree	Slightly Disagree	Slightly Agree	Agree	Strongly Agree
I was immersed in the story						
I found the quests interesting						
I felt like this could be a real location.						
I found the ending satisfying						
I would recommend to a friend						

5 words based on your experience with the game:

Accessible	Desirable	Gets in the way	Patronizing	Stressful
Appealing	Easy to use	Hard to use	Personal	Time-consuming
Attractive	Efficient	High quality	Predictable	Immersive
Busy	Empowering	Inconsistent	Relevant	Too technical
Collaborative	Exciting	Intimidating	Reliable	Interesting
Complex	Familiar	Inviting	Rigid	Uncontrollable
Comprehensive	Fast	Motivating	Simplistic	Unconventional
Confusing	Engaging	Not valuable	Slow	Unpredictable
Connected	Fresh	Organized	Dull	Usable
Consistent	Frustrating	Overbearing	Stimulating	Satisfying
Customizable	Fun	Overwhelming	Straight Forward	Valuable

which face you shown while playing the game:

5 words based on your experience with the game:

Accessible	Desirable	Gets in the way	Patronizing	Stressful
Appealing	Easy to use	Hard to use	Personal	Time-consuming
Attractive	Efficient	High quality	Predictable	Immersive
Busy	Empowering	Inconsistent	Relevant	Too technical
Collaborative	Exciting	Intimidating	Reliable	Interesting
Complex	Familiar	Inviting	Rigid	Uncontrollable
Comprehensive	Fast	Motivating	Simplistic	Unconventional
Confusing	Engaging	Not valuable	Slow	Unpredictable
Connected	Fresh	Organized	Dull	Usable
Consistent	Frustrating	Overbearing	Stimulating	Satisfying
Customizable	Fun	Overwhelming	Straight Forward	Valuable

which face you shown while playing the game:

## Appendix E: *Potential Graphics*

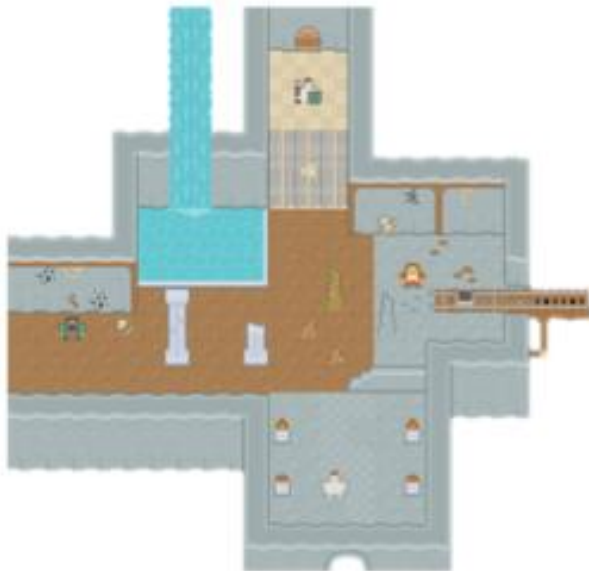
Here is a design of our main sprite: Hip Hop. We decided to keep the design of the sprite rather simple so it is easy to redraw to implement different movements and to code. At the same time, we wanted to maintain the character's personality with the drawing, giving Hip Hop a backwards cap, shirt and trainers.





## APPENDIX D – GAME SPRITES

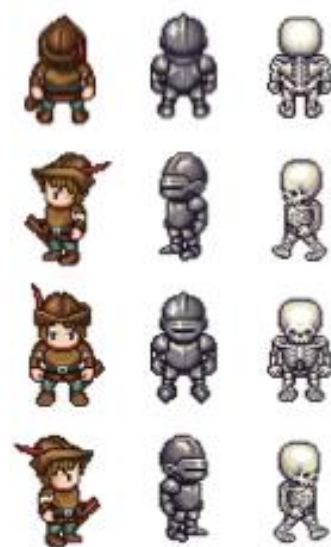
Sample of Selected Tile-sets



Player Sprite



Example Enemy Sprites



Thank you kindly...any questions?

