



# UNIVERSITY OF KWAZULU-NATAL

---

## COMPUTER METHODS 3

---

Super Mario San Andreas

213508238 Keshav Jeewanlall

214512918 Kahil Thiakdhari

214502393 Devashnee Bhagawandin

214551467 Nivaan Krishundutt

214529934 Priyen Munien

## Contents

<b>1) The story line of Super Mario San Andreas .....</b>	<b>2</b>
<b>2) Software development methodology and lifecycle .....</b>	<b>3</b>
<b>3) Software architecture .....</b>	<b>3</b>
<b>3.1 Basic declarations .....</b>	<b>3</b>
<b>3.2 The main game loop .....</b>	<b>4</b>
3.2.1 The display window.....	4
3.2.2 keyboard events.....	4
3.2.3 Timer events .....	5
<b>3.3 Game implementation methods.....</b>	<b>5</b>
3.3.1 Level 1 .....	5
3.3.2 Level 2 .....	5
3.3.3 Level 3 .....	5
<b>4) References .....</b>	<b>6</b>
<b>5) Github links.....</b>	<b>6</b>
5.1. Main game repo.....	6
5.2. Contributors .....	6

## 1) The story line of Super Mario San Andreas

Mario and Princess Peach's life was finally perfect, in any case, why wouldn't it be? Bowser who had previously kidnapped the Princess from the Mushroom Kingdom was dead. Mario's brick breaking and mushroom mashing days were far behind him.

Luigi, Mario's brother had moved away from their village a few years ago, in search of job in the big city, after helping his brother rescue Princess Peach from the evil clasp of Bowser.

Mario, previously a plumber, had chosen a new profession just so he could always be near his love Peach. Both he and Peach had become the local home decorators in the village. It may not be as exciting as how life was but at least he knew that every day of the rest of his life will be spent with Peach, or so he thought. . .

One morning, Mario was startled to find his Princess not in bed beside him. Frantically he rushed throughout the village in search of her, only to return home unsuccessful. Could it be happening again? Is his precious Princess once again kidnapped?

After some time, Mario hears a knock on the door and swiftly goes to open it only to find no one but a letter on the doormat. He opens the letter and it reads:

“Brother, after all these years of living in your shadow I've finally got what I truly deserve, I have my Princess and she's all mine now. Don't waste your time coming after me. The city is a big place and without friends you might find yourself in places you'd rather not be in. I have my sources and the moment anyone sees you anywhere near the city or my princess, YOU WILL PAY

It's time for my happily ever after, brother  
yours truly  
LUIGI  
“

Furiously, Mario races to the city of San Andreas in hopes of winning back his Princess.

## **2) Software development methodology and lifecycle**

The objective of this project is to display students' knowledge of C++ which is an object oriented programming language and ability to work as a team to develop software. The software being developed is a 2D graphical game. Microsoft Visual Studio together with Allegro 5 are the integrated development environment used in the development process. A public GitHub repository was created and used as source control, allowing for the latest version of software to be available to all contributors of the project. The well know games of Super-Mario and GTA San Andreas are incorporated to develop the game called Super Mario San AnDreas .

A combination of an agile and plan driven process was chosen in implementing the software development lifecycle for Super-Mario San-Andreas. This methodology best suited the development team of this game due to the arising challenges. Allowing the team to change the process to reflect and achieve the desired specifications. The agile method allowed the team to have flexibility where as the plan driven methods allowed the team to have an overview of the end goal.

Plan driven method allowed for establishment of classes and relationships between classes to be recognised. The initial architecture of the game was jotted down and incremental development to start.

Incremental development took an agile approach where members of the development team where assigned tasks and sub-sections of the game to develop and test. A GitHub repository was set up, allowing the development team access to the current version of the game. The team checked the plan to see if initial specifications where met and then went into the intermediate version for development. Validation took place after the intermediate development version was completed and resulted in the final version of the software being development and ready for release.

## **3) Software architecture**

The code was split up into classes allowing for the best deployment of object orientated programming. Concepts such as polymorphism, inheritance, templates and generic programming are the basic concepts of object-orientated programming used in the development of the game. Allegro 5 is used in order to incorporate sound and graphics.

### **3.1 Basic declarations**

In every class and file the appropriate header files and .cpp files were included to assist the compiler when compiling the code. Multiple iostream, namespace and allegro files where used and theses needed to be included at the top of each file.

Multiple .cpp files are used to break up the code into classes keeping the code more organised. When a development team separates the code into different cpp files it becomes easier to work as a team. When errors arise it becomes easier to detect by the compiler and developer because code is broken down. Multiple cpp files makes the code more complicated but it is good programming practise to break down code allowing for changing specifications and further development in the future.

In the declaration of bitmaps and other variables for multiple functions were declared and later on in the program destroyed to account for memory leaks that may occur.

### 3.2 The main game loop

#### 3.2.1 The display window

Allegro objects were created and used to initialised the display game window. Functions such as ALLEGRO\_DISPLAY and al\_set\_window\_position() were used to display the window on the computer screen. The screen was also made resizable for user preference. The world graphics are drawn to the window using allegro functions. The picture moves using the camera position function.

#### 3.2.2 keyboard events

Keyboard events		
<u>Keys pressed</u>	Game is being played	Game is on pause screen
up	If Mario is on ground he jumps	Goes to the game
down	If on ground Mario ducks ;if in the air he falls	
right	Mario moves to the right	
left	Mario moves to the left	
enter		
space	Mario attacks	Start game
escape	Exit game window	
<u>Keys are released</u>		
Up	If Mario in the air he falls	
Down		
Right	Stop moving right	
Left	Stop moving left	
Space	Stop attacking	
Enter		

### 3.2.3 Timer events

the timers are used in order to keep the game updated with the chain of events in the game. This helps to achieve smoothness in animations by refreshing different aspects achieving the desired results.

## 3.3 Game implementation methods

The game consists of 3 levels. The game world obstacles which is the base class for all of the obstacles used in the for the game. The classes which inherit from the base class are the pillar, manhole and spike class. The pillar (blocks) class are used to help Mario over world obstacles. Manhole class services as breaks in the world that causes death to Mario. Spikes are also obstacles that cause death to Mario. The world also has enemies, which causes death to Mario. Mario builds strength and experience over the level making him faster and stronger. The basic strategy of this game is to get through to the end of the stage while trying to avoid getting killed by enemies and obstacles such as spikes and manholes. The bricks can be used to navigate over obstacles by jumping onto them.

### 3.3.1 Level 1

Block and enemies are positioned to help display the story line. A manhole is placed at the end to tie in with the next level.

### 3.3.2 Level 2

The same world obstacles are used as level one. The difference is enemies are now placed onto the blocks to increase the difficulty and manholes appear frequently.

### 3.3.3 Level 3

World obstacles from worlds 1 and 2 are present in world 3 with a introduction to spikes that causes death to Mario. The spikes increase difficulty. Once world 3 reaches the end Mario has to fight his brother Luigi in order to mark the safe return of his princess.

#### 4) References

<http://iconizer.net/en/search/No-license-filtering/0-128/1/car>  
<http://bestfreeclipart.tk/clipart/7358/>  
<http://www.clipshrine.com/Tour-bus-17682-medium.html>  
<https://openclipart.org/detail/210007/ilmenskie-interior-gr-manhole>  
[spikeshttp://scribblenauts.wikia.com/wiki/File:Long\\_Spike\\_Row.png](spikeshttp://scribblenauts.wikia.com/wiki/File:Long_Spike_Row.png)  
<https://www.youtube.com/user/CodingMadeEasy>  
<http://www.codingmadeeasy.ca/>  
<https://www.github.com/codingmadeeasy>

#### 5) Github links

##### 5.1. Main game repo

<https://github.com/Kahilt/SuperMarioSanAndreas.git>

##### 5.2. Contributors

<https://github.com/PriyenMunien>

<https://github.com/Devashnee>

<https://github.com/KeshavJ>

<https://github.com/Kahilt>

<https://github.com/Nivaan214551467>