ANGRY BIRDS 3

Maze Runner



DAT602

Milestone THREE

Dean Stanley-Hunt | 21/06/2017

TABLE OF CONTENTS

[EXECUTIVE SUMMARY - GAME DESCRIPTION 2](#_Toc485850244)

[STORY BOARDS & NARRATIVES 3](#_Toc485850245)

[LOGIN SCREEN 3](#_Toc485850246)

[MAIN MENU 4](#_Toc485850247)

[CHARACTER SELECTION 5](#_Toc485850248)

[HOST GAME 6](#_Toc485850249)

[ADMINISTRATION 7](#_Toc485850250)

[GAME SETTINGS 8](#_Toc485850251)

[MAZE CREATOR 9](#_Toc485850252)

[IN GAME SCREEN 10](#_Toc485850253)

[CRUD TABLE 11](#_Toc485850254)

[DESCRIPTION 11](#_Toc485850255)

[LOGICAL ERD 12](#_Toc485850256)

[DESCRIPTION 12](#_Toc485850257)

[MILESTONE 2 DESIGN MODIFICATIONS 13](#_Toc485850258)

[CHANGES MADE TO THE GUI: 13](#_Toc485850259)

[CHANGES MADE TO THE DATABASE: 13](#_Toc485850260)

[ACID REVIEW 13](#_Toc485850261)

[ACID DEFINITIONS 13](#_Toc485850262)

[ATOMICITY 14](#_Toc485850263)

[CONSISTENCY 15](#_Toc485850264)

[ISOLATION 15](#_Toc485850265)

[DURABILITY 15](#_Toc485850266)

[MILESTONE 3 DESIGN MODIFICATIONS 15](#_Toc485850267)

[CHANGES MADE TO THE GUI: 15](#_Toc485850268)

[CHANGES MADE TO THE DATABASE: 16](#_Toc485850269)

[FOLLOW UP SUMMARY 16](#_Toc485850270)

[FINAL GAME IMAGES & VIDEO 17](#_Toc485850271)

# EXECUTIVE SUMMARY - GAME DESCRIPTION

So after giving it some thought its decided to make a Maze game, that multiple players can join and walk around in a Maze level, with a goal point to reach, and generates new goal position each time reached so they keep going around collecting as many as possible in the given time of 4 minutes.

It will use Angry Birds characters and the goal can be the Angry Birds eggs as per the actual Angry Birds cartoons, where the pigs are always trying to get the eggs and birds need to get them back… player collects eggs and gets a score based on time taken, or incrementing on how many goals reached. As with the Angry Birds games which commonly have TNT explosive boxes, It will include these to smash into and it will Blast the player to a new XY point in the maze.

There is going to be multiple characters skins to choose, sound FX, and background music for different parts of the game. Game settings for adjusting volumes, and turning of music and sound FX. The player will be able to decide if they want it windowed or full screen resolution read from their screen settings.

When games end, players can start again or leave game, each player gets a set time frame to get the Eggs, these times are individual for each player to make it fair, so even if a player will join the game when other players are 2 minutes into getting eggs, they are not limited to whatever time other players have left for getting Eggs, the new player gets the full amount of time, and everyone having the same amount of time on the maze will make it fair to see who can get the best score in the given time frame.

If the host who started the public game, is to leave the game, it will remain active and not effect players. Players can keep playing and leave when they like, but once the last person leaves a game, it will then be removed from active games list and no longer exist. However if the original host wants to host a new game again, then when they start hosting a new game, the old game they created will be forcefully removed kicking all players out.

Players will be able to create 1 Custom Maze for their account, and everyone will have access to using it for games. This feature of the game will help to see the Available Maze maps grow with efforts from Players. The brief said we need to make it so players select a player online who is not in game to start a new game with, that makes sense for games like battleship where you need another player to start a game at all, and are limited to how many players perhaps, this maze game isn’t like that by choice of concept, players can play alone, and others can join whenever they like.

# STORY BOARDS & NARRATIVES

## LOGIN SCREEN

|  |
| --- |
| 1. Narrative: |
| The Login screen will provide the user with the ability to input a username then password for the username if username existed in database to login or they will be notified they are about to register it if doesn’t exist yet, and to input password for new account, it will add new user to the database with other stats as defaults, and login the user, and on successful login move to a game menu screen. If login failed, or account to register exists, pop up message to notify the user accordingly. If login failed 5 times, account gets locked. |
| 1. Storyboard: Login |
| |  | | --- | | Angry Birds 3 Maze Runner | | 5) Label for Status/Error/Locked/Etc  3) Password  2) Username  1) ANGRY BIRDS 3 LOGO | |
| 1. Action List: |
| 1) Animation game logo  2) User name input  3) Password input  4) Login button that when clicked checks account exists against database and shows input for password if does exist, if password correct then goes to game menu. Otherwise label saying failed login message, or locked if failed to login 5 times. If username didn’t exist in first place it will tell them they are about to create this account, and show password input, add username and password to Database after that.  5)Label Showing different stats according to actions happening! Wrong password/Locked/Etc |

## MAIN MENU

|  |
| --- |
| 2. Narrative: |
| This is the game menu reached after successful login, it will populate all currently hosted games, to provide the option to join a current game with a double click action, also showing a leaderboard list of the top 20 players scores this is where players aim to get their name and own the fame. One last list to show all players online right now! There are 3 buttons along the bottom to provide other options like host game which will take the user to a host game screen where they can select a maze for their hosted game, admins only will be able to see the admin button taking them to a admin screen to moderate the games back end, and lastly a button taking the user to a screen where they will be able to design and build their very own maze level which will be saved in database 1 maze per 1 user, and become visible to all users and a selectable maze map for hosted games. |
| 2. Storyboard: Game Menu |
| |  | | --- | | Angry Birds 3 Maze Runner | | 2) Leaderboard (Player, Highscore)  6) Players Online Now  1) Active Games (Host, Map) | |
| 2. Action List: |
| 1) Active Games List showing the Host, and Map, Double clicking a game will show character store the game you want to join and show character selection screen before actually joining game.  2) Leaderboard showing the top 20 players highscores.  3) Button that leads to the Host game screen where player can select Maze Map and start hosting.  4) Button only visible to Admins to goto Admin screen.  5) Button that takes user to a Maze Builder screen to make create there own map.  6) List of the players online now and there highscores.  7) Button to Kill running game that only appears to Admins, select game from list and kill. |

## CHARACTER SELECTION

|  |
| --- |
| 3. Narrative: |
| This is the Character selection screen and the user can see 6 characters to choose from for game player model. Upon clicking and selecting a character, it gets a border around it to illustrate it has been selected, and user can then click the Start Game button to now join the previously selected or newly hosting game. |
| 3. Storyboard: Character Select |
| |  | | --- | | Angry Birds 3 Maze Runner | | 2) Characters  ( Picture Box showing 6 selectable Characters)  1) Label – “Character Selection” | |
| 3. Action List: |
| 1) Label saying ‘Character Selection’.  2) Picture Box showing 6 characters to select from as images, showing a border around the currently selected character, if character clicked then border changes to new selected character.  3) Button to confirm character selection and officially join a game, goes to Game Screen.  4) Button to go back to game lobby an not host/join a game |

## HOST GAME

|  |
| --- |
| 4. Narrative: |
| This is the Host Game screen that allows the user to select from an Original Maze map or Player created Maze Map in the database, and then by a click of a button adds their newly hosted game to the list of games being hosted, and the game is then essentially live to everyone to see, and the user is then taken to the Character select screen to select character before entering actual game screen. |
| 4. Storyboard: Host Game |
| |  | | --- | | Angry Birds 3 Maze Runner | | 1) Maze Maps List | |
| 4. Action List: |
| 1) A List of Original Maze Maps, and Players Custom Maps to select from for hosting a new game. By default the first in list is pre-selected to avoid errors.  2) Button to Start hosting once Map is chosen, communicates with database to add game to database for all to see it and then takes user to character selection screen.  3) Button to go back to games screen and cancel hosting a game. |

## ADMINISTRATION

|  |
| --- |
| 5. Narrative: |
| The Admin screen will show a list of players and when a player name is clicked, details/edit panel to right side will update things related to that player which an admin can edit, password, highscore, admin, locked, and current in game score if any. then once happy with single or multiple changes, click the apply changes button to update database instantly, label will show status of query or updates. Ability to add and remove player accounts or change password. |
| 5. Storyboard: Admin Screen |
| |  | | --- | | Angry Birds 3 Maze Runner | | 3) Add / Delete / Change Password / Apply Changes  4) Player Details shown in text boxes and Check boxes and changeable. Click Update to commit changes!  2) Label - Status: Idle  5) Close  1) List of all players names | |
| 5. Action List: |
| 1) Datagrid showing the Players table in database containing all things a moderator admin should need to access, and the admin can interact with the datagrid to make changes on the visible datagrid, the changes will not be implemented immediately, only once update is clicked.  2) Label showing Status of table loading and updating.  3) 4 Buttons as Follows ‘Apply Changes’ that once clicked will update the selected player in lists changeable details from Player details panel into the database.  ‘Add’ that once clicked will prompt with inputboxs asking for new username and password to add to database, then after adding other fields will be defaults for the database and Refresh the Datagrid.  ‘Delete’ once clicked will delete the selected players row in the datagrid, or possibly might prompt with inputbox for admin to input username to remove.  ‘Change Password’ once clicked prompts a input box asking for the new password for selected player, then prompts if your sure you want to change the password for said player, Yes or No.  4) Player details panel with text boxes, check boxes showing currently selected players changeable info, admin can make changes here then click the apply changes button!.  5) Close Admin panel button, hides the admin groupbox back to game lobby. |

## GAME SETTINGS

|  |
| --- |
| 6. Narrative: |
| This Game settings screen will allow the user to adjust and customize some graphical and audio capabilities for the game such as windowed or full screen, music and soundfx volumes, all the most common things you expect from any game these days. Once the user has setup the settings they like, they are held locally as generally are best suited for the machine the game is running on at the time, not always suitable across multiple machines if they login the same account and it updated to game settings held in a database for example, it might not suit the different machine, so that is why these settings will be locally stored and loaded each time game is opened, this settings window will always pop up as the first thing gamers see. |
| 6. Storyboard: Settings Screen |
| |  | | --- | | Angry Birds 3 Maze Runner | | 5) Launch Game  4) Game Music / Menu Music / Sound FX – Three Separate CheckBoxes  2) Windowed? CheckBox  1) Resolution Label | |
| 6. Action List: |
| 1) Label for Resolution Setting, Loads screens current resolution by default and will only change if windowed mode is selected  2) Check box saying ‘Windowed’ which if checked will load game in windowed fashion on screen instead of full screen like unticked does.  3) 3 separate Volume level slider bars to the left of their check boxes counterparts that turn ON these sounds and their volumes if checked. Adjusting these 3 volume levels will effect how the game projects its audio, after some testing I will identify the best default settings for these.  4) These are 3 separate check boxes that align right of their slider bar counterparts, if checked then audio music or FX can be heard in game, unchecked is like muting the audio type.  5) Button to Launch game with the selected settings, hides the form, and calls the Main game form to load and initialize according to the settings. |

## MAZE CREATOR

|  |
| --- |
| 7. Narrative: |
| The Create Maze Screen will provide the player with an interactive board of coloured labels, and a maximum number of blocks they have left to place in the maze parameters. For each block placed it will change the colour of the label they clicked pretty much, and show them essentially an outline for how their custom maze is going to look for its layout. Once they are happy with the maze they created they can then publish their new or updated maze for it to become visible to all players who want to host or start a new game. They user can also cancel building this new or updated maze. But users are not able to delete a Maze, once first create a Maze, there will always be one associated to the account and adding to an ever growing list of Mazes available to play. They can only update their stored maze by publishing a new one. Always storing only one per player. |
| 7. Storyboard: Create Maze Screen |
| |  | | --- | | Angry Birds 3 Maze Runner | | 3) Publish Maze  4) Cancel  1) Maze Designer Panel Box- GroupBox / Label’s  2) Blocks Left:400 | |
| 7. Action List: |
| 1) Groupbox & Labels outlining the Maze parameters and providing interchangeable label colouring at a click, all effecting the Label showing how many blocks are left to change colour for to the Maze walling colour, if they remove a maze wall colour from the display the block count goes up, and once user is happy with the maze they have designed, then interactions with this part will be finished.  2) Label showing Blocks Left: # , showing the amount of blocks they have which really effects how many label colours they can click to make into maze walling to design the walls for their own custom maze pattern, if they click a maze wall to remove it, this count goes up again.  3) Button to publish the Maze and make it saved in their account and public to all gamers to now play on and enjoy! Publishing will replace any existing maze associated with the user account if any.  4) Button to Close this Maze Creator screen, and cancel creating one, making no change to DB. |

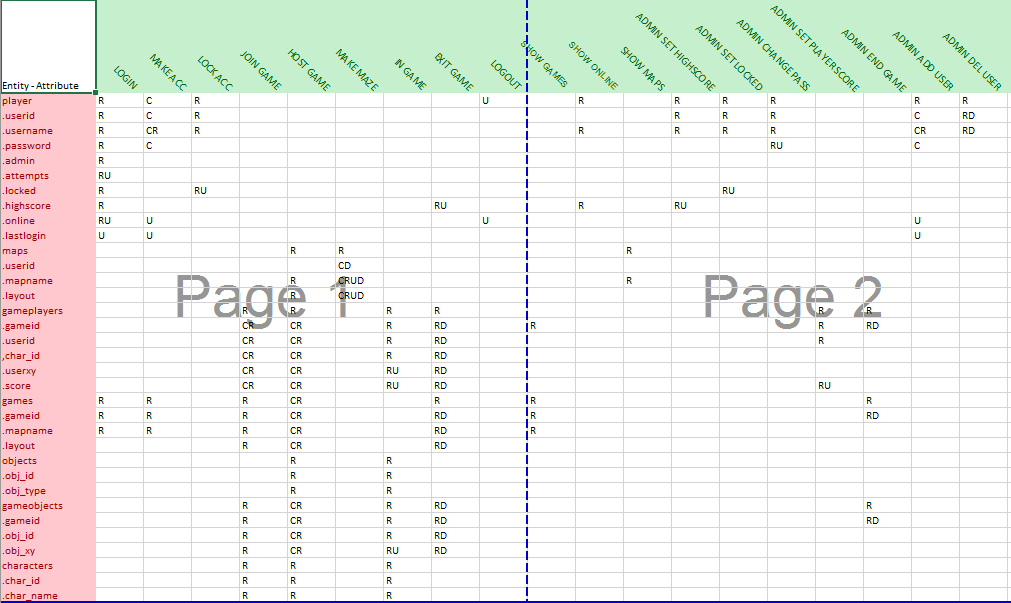
## IN GAME SCREEN

|  |
| --- |
| 8. Narrative: |
| The in Game Screen is where all the real fun happens, players get submerged into a well presented graphical fast paced game, where they control their character running around the maze with other players, trying to get the eggs first, smashing into TNT explosive boxes which blast them to a random location in the maze. They could end up closer to the eggs and get them first to get the points, or it can backfire and put them further away. The player will see the time count down until reaching 0, and then the game finishes for that player showing them the score reached and asking if they want to play again or go back to main menu. Players can see the score for every player in the game. |
| 8. Storyboard: In Game Screen |
| |  | | --- | | Angry Birds 3 Maze Runner | | 4) Game Play Graphics Display  ( Maze Walling, Player Characters, Game Objects like Eggs and TNT )  1) Exit  3) Time: 012  2)  P1: 3 P2: 5 P3: 1 | |
| 8. Action List: |
| 1) Exit game button, player can click to leave game at any time and go back to main menu.  2) Datagrid showing all players names, characters, position and scores!  3) Label showing how much time left to play before game finishes for an individual player.  4) Game play graphics are being constantly drawn in this area. |

SO WHATS NEXT?

That concludes the storyboard and narratives for my game! Next lets take a look at what the database needs for my game are, I have done a CRUD table and Logical ERD to see below followed by my Design decisions and ACID review.

# CRUD TABLE



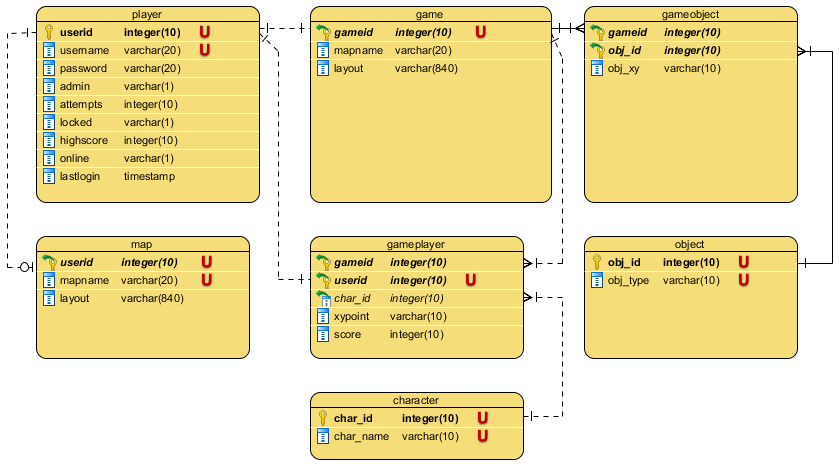
### DESCRIPTION

The above is a CRUD Table designed to show all Table Entities with their Attributes going down, while across shows different stages of using the game, then you can see various letters corresponding in each row and column which represent the interaction with database required for each stage, right down to with attributes needed to be created, retrieved, updated or deleted!

EG. Under LOGOUT we see 2 ‘U’ letters for Update action with database where it needs to Update on the player table, the variable called ‘online’. This helps show what queries are needed!

# LOGICAL ERD

ENTITY RELATIONSHIP DIAGRAM

****

### DESCRIPTION

The Logical ERD shows all Tables, Attributes, Data types, and some constraints such as Unique, Primary key and Foreign Key. There is 7 Tables, and a variety of relationship links ranging from one to zero or one, one to one and one to many. The Main Table is the player entity, relationships are:

* One Player can have None or One Maze Map.
* One Player can Host One live Game.
* One Game session can have Many Game Objects
* Many Game Objects each use One Object
* One Game session can have Many Game Players
* Many Game Players each use One Character
* One Game Players Username belong to One Player

Now the games Database is ready to be created, tested, iterated if needed then implemented.

# MILESTONE 2 DESIGN MODIFICATIONS

## CHANGES MADE TO THE GUI:

So the changes made for the game since Milestone 1 are visible in the storyboard, added a few buttons to various game screens to support going back to game lobby from character selection for example. And more importantly changed the admin section to more how the brief asks.

As you know milestone 1 was planning to showcase another method of working with the database to show details to a data grid, alter them directly and click a button to update database to match any changes made without procedures or writing SQL but instead using s SQL Command Builder. But this has been changed due to the requirement to show all admin interaction via a Console Application for testing the database. Since all the procedures for the admin interactions now exist, the actual game can now just use those, hence the design changes.

## CHANGES MADE TO THE DATABASE:

So the changes made for the database since Milestone 1 are visible in the Logical ERD, Only one small change for the gameplayer table where it was username referencing username from player table a non primary key, now changed to be user id on gameplayer table referencing the user id from player table. Test data was already a satisfactory representation of an active game state.

One other change was suggested by Todd which would allow the same one player to host multiple games at once, but for the design and way game works for this game it is highly undesirable to have one person able to just start hosting multiple games.

They could add 300 games and just fill out the active games list showing to other players with empty games being all hosted by one person, not the kind of mess desired, it is strictly 1 to 1 relationship for 1 player can only host 1 game at a time.

The gameid is equal to the userid for who is hosting for cognitive ease and because it is already unique enough to be the primary key, same as it is unique enough to be the map id and primary key for the maps table.

# ACID REVIEW

## ACID DEFINITIONS

Quoted from Web Source: <https://en.wikipedia.org/wiki/ACID>

*“Atomicity*

*Atomicity requires that each transaction be "all or nothing": if one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged. An atomic system must guarantee atomicity in each and every situation, including power failures, errors and crashes. To the outside world, a committed transaction appears (by its effects on the database) to be indivisible ("atomic"), and an aborted transaction does not happen.*

*Consistency*

*The consistency property ensures that any transaction will bring the database from one valid state to another. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof. This does not guarantee correctness of the transaction in all ways the application programmer might have wanted (that is the responsibility of application-level code), but merely that any programming errors cannot result in the violation of any defined rules. The sum of A and B data is unchanged by the execution of transaction. In general consistency required specified integrity such as primary key and foreign key.*

*Isolation*

*The isolation property ensures that the concurrent execution of transactions results in a system state that would be obtained if transactions were executed sequentially, i.e., one after the other. Providing isolation is the main goal of concurrency control. Depending on the concurrency control method (i.e., if it uses strict - as opposed to relaxed - serializability), the effects of an incomplete transaction might not even be visible to another transaction.*

*Durability*

*The durability property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors. In a relational database, for instance, once a group of SQL statements execute, the results need to be stored permanently (even if the database crashes immediately thereafter). To defend against power loss, transactions (or their effects) must be recorded in a non-volatile memory.”*

## ATOMICITY

-In this Game ATOMICITY is being implemented for every procedure, because the game is a MMO consisting of multiple users possible hundreds or thousands, who would all need to interact with the games database at the same time in a lot of cases, it is required for accuracy of the state of the database at any given moment to any user.

To handle this the database is using transactions with auto commit turned off, so in the procedures any and all interactions inside a procedure are stored into a transactional state, and don’t execute until told to commit, making sure all interactions happen all at once.

With the added benefit of error handling to catch errors, the transaction can be told to roll back any or all changes made to act as if they never happened preserving the database state to be unaffected and prevents possible state corruption. This database is also using avoidance as error prevention by checking if things exist or not before doing any interaction to alter the state of the database.

This is extremely useful and important towards any MMO game!

## CONSISTENCY

-Fortunately because the game is using INNO DB for its database schema it gets the benefit of its internal features designed just for providing consistency by protecting data from crashes and providing crash recovery support. Protection of data from crashes of the database is essential for the game and its users, wouldn’t you hate to have been playing for 5 years, built up the number 1 top spot for the leader board and then lose it and lose your account perhaps too, due to the database crashing and corrupting all or some data, it is highly undesirable.

## ISOLATION

-The ISOLATION part of the game refers to the isolation level of transactions and the locking of access to elements inside the database schema. This has not been implemented in this games database design for milestone two, but can be in future to restrict access to tables for modifying them while a transaction is being processed, so that only that transaction can access it until said transaction in a procedure is completed where the isolation level can be lifted again and better work with the games ATOMICITY.

## DURABILITY

-For the games database durability MySQL has a range of configuration options available which have not been needed for milestone two, but could be implemented before distribution and thinking about the system hardware requirements to handle the expected load appropriately.

One important thing that would definitely be implemented outside of configuration settings would be running the database on a system powered by a UPS, so in the event there is a power outage it remains ‘ON’ and data is not effected, even the live accessibility can remain active in such an event, while at the same time helping prevent power surges to the system.

Another strategy to maintain durability through any situation the game will implement shall be regular scheduled backups, locally and cloned to an offsite location in another country for protection of data loss in extreme circumstance such as a fire, earthquake or flooding.

# MILESTONE 3 DESIGN MODIFICATIONS

## CHANGES MADE TO THE GUI:

So the changes made for the game since Milestone 2 are as follows, I have decided to implement admin area control to be a data grid style after all, and click to edit from tables directly, then once done editing information, the administrator can click a button to update all possible changes.

Other than that not a lot changed about my GUI design, I have implemented everything I set out to achieve such as Windowed or Full screen modes, audio sounds, maze building feature, and made it graphically pleasing on the eye I feel as a bunch of added extra’s to the game, probably far above the level other learners are capable of doing, but I have had a blast developing this.

The game was first tested locally hosted and running very smoothly and great, but I did run into a bunch of problems when I migrated the database to be hosted and running from the MariaDB server setup somewhere in Singapore as I was told by Todd. The game became slow.

I then had to redesign the what the game was running game play, and make changes to the database which I will cover below in a moment. But firstly its important to know that with this report and the game source code files, is a notepad text file marked important that you should read before going ahead with using the game, and testing out the game.

Basically saying you need to run the SSH tunnel to connect to the game server, and to use the login details provided for the SSH Tunnel. And some other useful information about where the Database interactions are all held inside the source code for the game to make every one’s life easier so be sure to read that, as the SQL text file does not need to be executed onto the server, it is already setup, but can be tested locally too, but mostly the same as Milestone 2, but with commented out Error Diagnostics because it didn’t want to work on the Maria DB.

## CHANGES MADE TO THE DATABASE:

So the changes made for the games database since milestone 2 and that I am most proud of in milestone 3 is the fact that my game now runs online via the server Todd hosts located somewhere in Singapore. Migrating the server was easy enough but did come without its quirks.

The MariaDB did not want to except some error diagnostic commands, and other little things, so I had to comment them out for the SQL file. Then everything worked as normal. But the next problem was how slow the game was running with all the latency between the server, so to solve this and get it running smoothly, I had to take what was 3 database interactions and make it all run at the speed of one database interaction.

I accomplished this by taking the get game objects procedure call and putting the request to update it, in to processing thread of its own on loop until game ends. That was one interaction that now could run seamlessly on its own thread uninterrupted. Next I decide to join together my procedure for getting the game players data and the procedure for updating the players own character position.

So now when it updates the players movement position, the database will return the table of current game players data, so that the game can redraw all the players and update players position all in one smooth database interaction, and thanks to the threaded task for object, it will draw all the objects too.

The change to the Update Player Position procedure was the only change I had to make for what was already a good database design, that I knew was going to serve its purpose for the game I set out to create here for the project. Now go enjoy have a play on Angry Birds 3 Maze Runner!

# FOLLOW UP SUMMARY

**Well that’s all there is to it, from Milestone 1 to Milestone 3, I have shown everything I can from design process, to testing and implementation. The game is robust, graphical, some error handling, very interactive with tool tip bubbles showing for each button to clarify what button does what. All the database interaction required for the game to run and fully function have been completed. The game is operation running online via the server in Singapore, and bonus extra included like sounds, full screen mode, maze level creator, administration that can alter the persons in game character, position, score, remove games, alter every aspect of a game players data and more.**

**Incase you didn’t know by now, you’re a bird or a pig, running around in custom player made maze levels, to rescue as many eggs as you can if you’re a bird, or if a pig its thought of to steal as many eggs as you can, as it goes with Angry Birds Cartoons.**

**TNTs as seen in the Angry Birds games, exist on levels, and if walked into, will explode, blasting your character to another part of the maze level, and then respawning the TNT somewhere else on the level again. You could end up closer to the eggs and get it before another player… or end up worse of and more further away! To risk it or not? Your choice!**

**Thanks for Reading.**

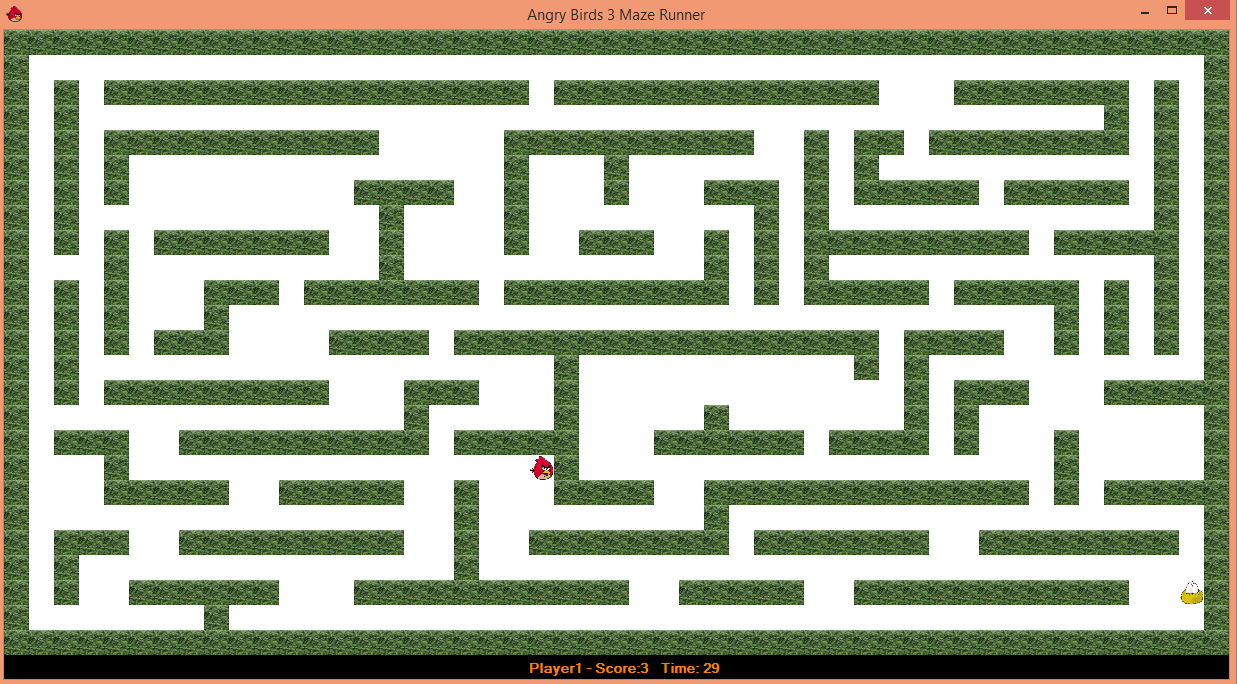
**Dean Stanley-Hunt**

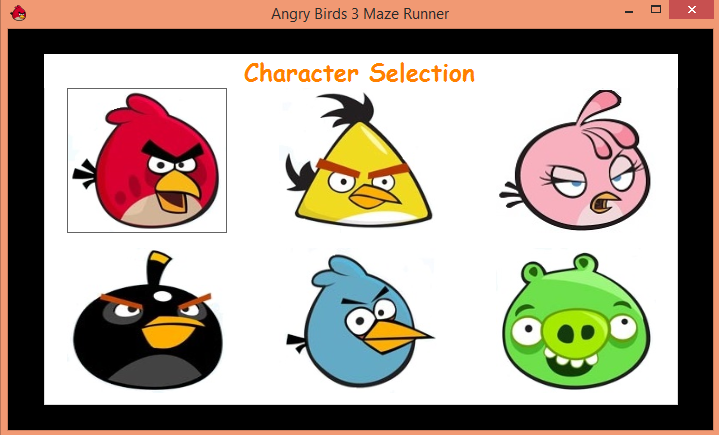
**DAT602 Milestone 3 – 21/06/2017**

# FINAL GAME IMAGES & VIDEO

<https://www.youtube.com/watch?v=4Tx_YuFoccY>

**From This….**





**TO THIS……………….!!!!!!!!!!!!!**

