Alaska Kiley Independent Code Review – Text-based Roguelike Adventure Game Engine/UI

My project was my first attempt to use C# and Visual Studios to create a program using .NET. I chose a text based adventure game due to my interest in game development and the fact that the precedence of no games was broken by the opportunity provided by the group project suggestion. I know I would not have been able create a complete game, but I hoped to make a presentation and basis of the possibilities of a game.

To explain my process, I divided my work into stages. My first stage was educating myself on C# and learning how to deal with a new IDE. I made a basic logic puzzle in Java and then converted it to C# for my first exercise. After I completed that, I made Tic-Tac-Toe in GUI for C#, a basic game, but something I knew I could do, just not in C#. My next stage was constructing my models for my game and creating my UI. During this stage I made the Move-A-Square demo which will be included in my code. This stage was rough due to the fact I had to remake my entire project, something I will elaborate on later. My final stage was implementing commands, which are handled by the Player Class, which interacts with the Dungeon Class. This stage is also when I tried and failed to implement the database.

- What was the most frustrating aspect of the project?

There were certain aspects of this project that were not fun for me. C# has commonly been referred to as Microsoft's Java equivalent, but they are not equivalent. C# and Java have certain distinct differences that I do not enjoy. Minor differences being in the syntax (first letter capitalized for methods for example), but additionally the online documentation of Java provided by Oracle are much more extensive and effective for educating an individual about the mechanics and workings of the language. As much as I enjoy Stack Overflow, being able to read a clear documentation would have been appreciated. Different versions of Visual Studios had different tutorials and not everything was updated properly I feel. This lead me to being ill-prepared to create my database for my project. I realize that I could have spent much more time pushing forward and learning how to create the database, but I did need sleep. Additionally, another minor frustration was having to test every font to see which worked best for my project.

A new unique difficulty for me during this project was breaking the habits I established due to coding in Java. I made what I felt was a very good outline for my project, but then as I was doing my work, I realized my outline was not sufficient, so I was forced to restart my project from scratch except for my UI. I was retrieving many errors due to how I implemented my model. This was definitely a point where I was unlearning habits from Java to open my mind to learning new things.

- What was the most satisfying aspect of the project?

I really enjoyed watching my project when it worked. While certain aspects of Visual Studios are unintuitive and annoying, .NET is an effective language and solution for UI and programs of this sort. Creating visual solutions for my project was fun to do and was very satisfying for me. It was easy to make something basic and intriguing and challenging to make more. The way

buttons and events happened in the engine was neat and I regret not mapping Enter to enter the commands, but instead the button is the only option for inputting commands.

- What did you learn that you could apply to future projects?

I learned how to work in Visual Studio which is a useful set of skills in my eyes. I can add that to my resume. I am able to make basic GUIs for projects using .NET. It makes me feel more qualified for work and ready to learn new things as I go on. I definitely learned more about .NET and Visual Studios than C#. I can make new programs in these languages now and feel more able to self-educate.

- What grade do you think you deserve for the project, based on how much effort you put into it, how much you learned, and how much you accomplished?

I would first estimate that my grade would be somewhere between 70% and 100%. The best way to describe it is to list the feature I intended to implement.

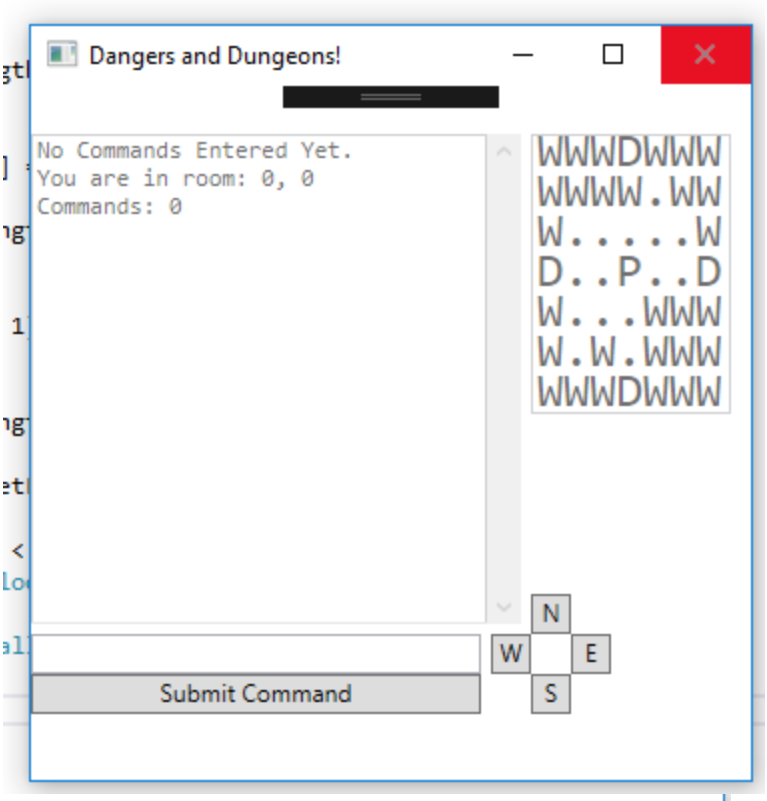
- Convert a small console program from Java to C#
- Convert my previous GUI version of Tic-Tac-Toe to C# and .NET
- Create a basic GUI with a text box updated by buttons
- Create a player model
- Create a dungeon model, containing a map of rooms which each contain a grid of tiles
- Implement multiple commands
- Save the map to a database
- Implement monsters who operate in a similar manner to the player, but with AI instead of commands
- Allow for the user to unlock doors

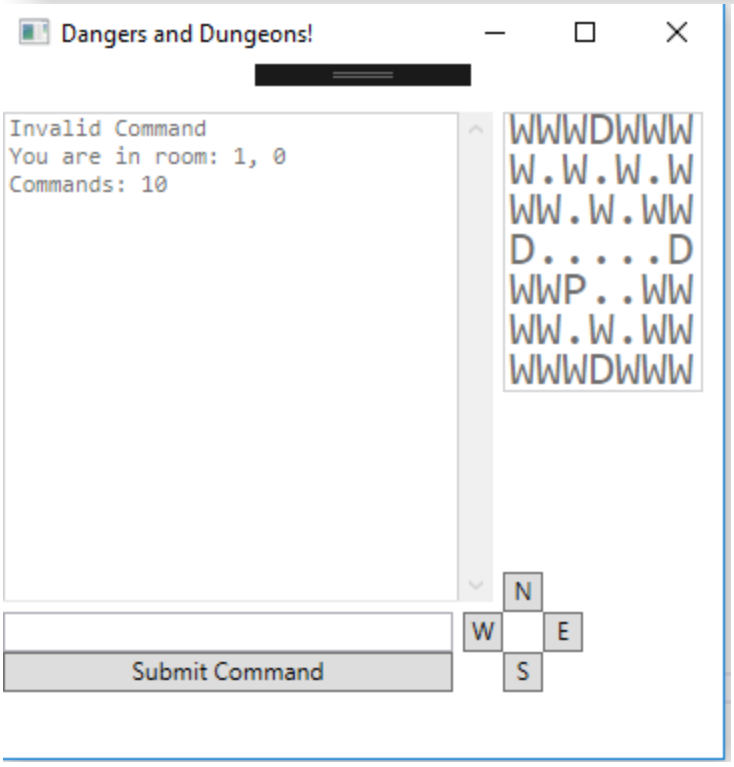
Not all of that got implemented obviously. Here are the reasons I have found why.

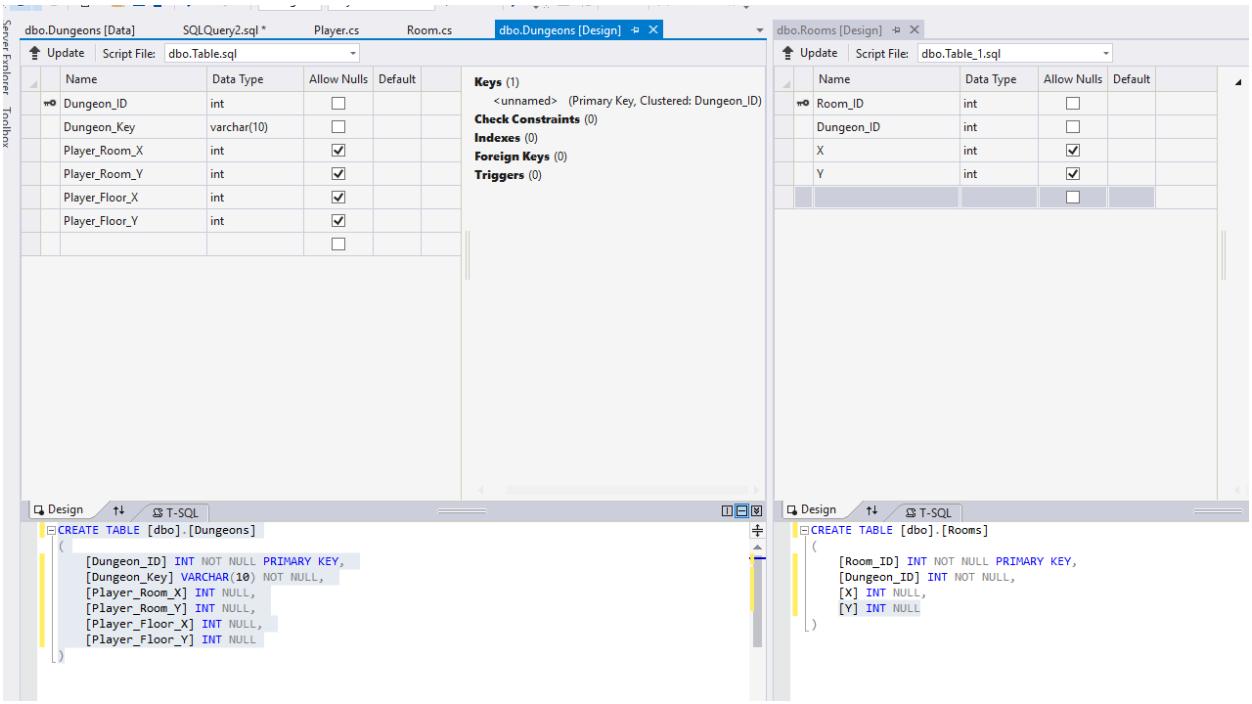
- Create a dungeon model, containing a map of rooms which each contain a grid of tiles
    - Not fully implemented, more details below at doors
- Save the map to a database
    - Did not have enough time to properly educate myself for the implementation and bring it together. I did not apply myself well enough to this problem. I feel I've failed this aspect of the project.
- Implement monsters who operate in a similar manner to the player, but with AI instead of commands
    - I felt I did not need to. I did so when I coded up a clone of Pac-Man so I already knew how to give commands to multiple AI in a game. I know that might be presumptuous, but I felt my time was better spent on other challenges.
- Allow for the user to unlock doors
    - Doors were sometimes blocked by walls, which I allow the user to remove, but originally, I planned for some doors to be locked and then allow for the user to unlock them, but I could not figure out how to cast a tile to a door, even though a door is obviously a tile. My understanding again was too rooted in Java

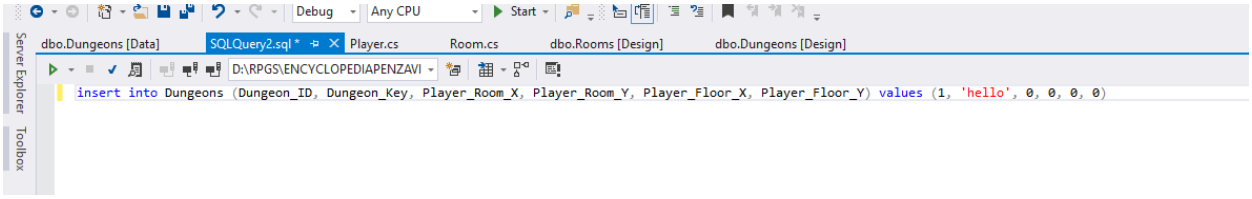
Overall, I wanted to exceed expectations, but I did not do that in my mind. I am unsure if I met expectations. I know I am below my own personal expectations, but I did feel I learned a lot during this project.

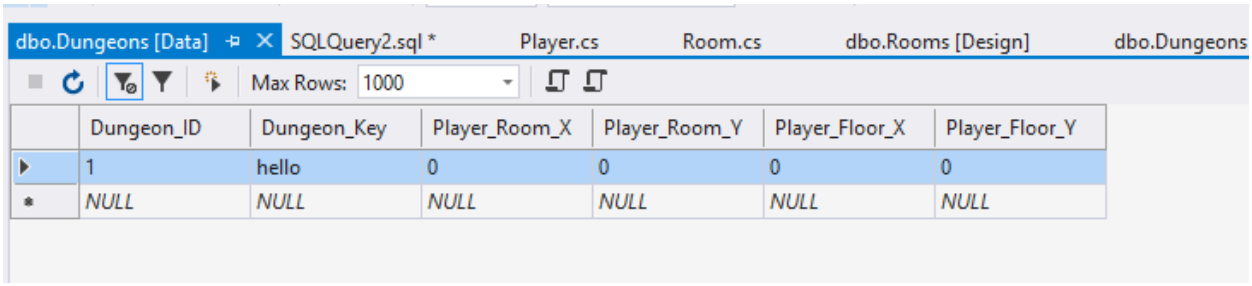
Here are some photos of my UI and my database progress


Dangers and Dungeons!
No Commands Entered Yet.
You are in room: 0, 0
Commands: 0
WWWDWWW
WWWW.WW
W.....W
D..P..D
W...WWW
W.W.WWW
WWWDWWW
N
W
E
S
Submit Command


Dangers and Dungeons!
Invalid Command
You are in room: 1, 0
Commands: 10
WWWDWWW
W.W.W.W
WW.W.WW
D.....D
WWP..WW
WW.W.WW
WWWDWWW
N
W
E
S
Submit Command


dbo.Dungeons [Data]
SQLQuery2.sql *
Player.cs
Room.cs
dbo.Dungeons [Design]
Update
Script File: dbo.Table.sql
Name
Data Type
Allow Nulls
Default
Dungeon_ID
int
Dungeon_Key
varchar(10)
Player_Room_X
int
Player_Room_Y
int
Player_Floor_X
int
Player_Floor_Y
int
Keys (1)
<unnamed> (Primary Key, Clustered: Dungeon_ID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)
Design
T-SQL
CREATE TABLE [dbo].[Dungeons]
(
[Dungeon_ID] INT NOT NULL PRIMARY KEY,
[Dungeon_Key] VARCHAR(10) NOT NULL,
[Player_Room_X] INT NULL,
[Player_Room_Y] INT NULL,
[Player_Floor_X] INT NULL,
[Player_Floor_Y] INT NULL
)
dbo.Rooms [Design]
Update
Script File: dbo.Table_1.sql
Name
Data Type
Allow Nulls
Default
Room_ID
int
Dungeon_ID
int
X
int
Y
int
Design
T-SQL
CREATE TABLE [dbo].[Rooms]
(
[Room_ID] INT NOT NULL PRIMARY KEY,
[Dungeon_ID] INT NOT NULL,
[X] INT NULL,
[Y] INT NULL
)


Debug
Any CPU
Start
dbo.Dungeons [Data]
SQLQuery2.sql *
Player.cs
Room.cs
dbo.Rooms [Design]
dbo.Dungeons [Design]
D:\RPGS\ENCYCLOPEDIAPENZAVI
insert into Dungeons (Dungeon_ID, Dungeon_Key, Player_Room_X, Player_Room_Y, Player_Floor_X, Player_Floor_Y) values (1, 'hello', 0, 0, 0, 0)


dbo.Dungeons [Data]
SQLQuery2.sql *
Player.cs
Room.cs
dbo.Rooms [Design]
dbo.Dungeons
Max Rows: 1000
Dungeon_ID
Dungeon_Key
Player_Room_X
Player_Room_Y
Player_Floor_X
Player_Floor_Y
1
hello
0
0
0
0
NULL
NULL
NULL
NULL
NULL
NULL