Dice Hero

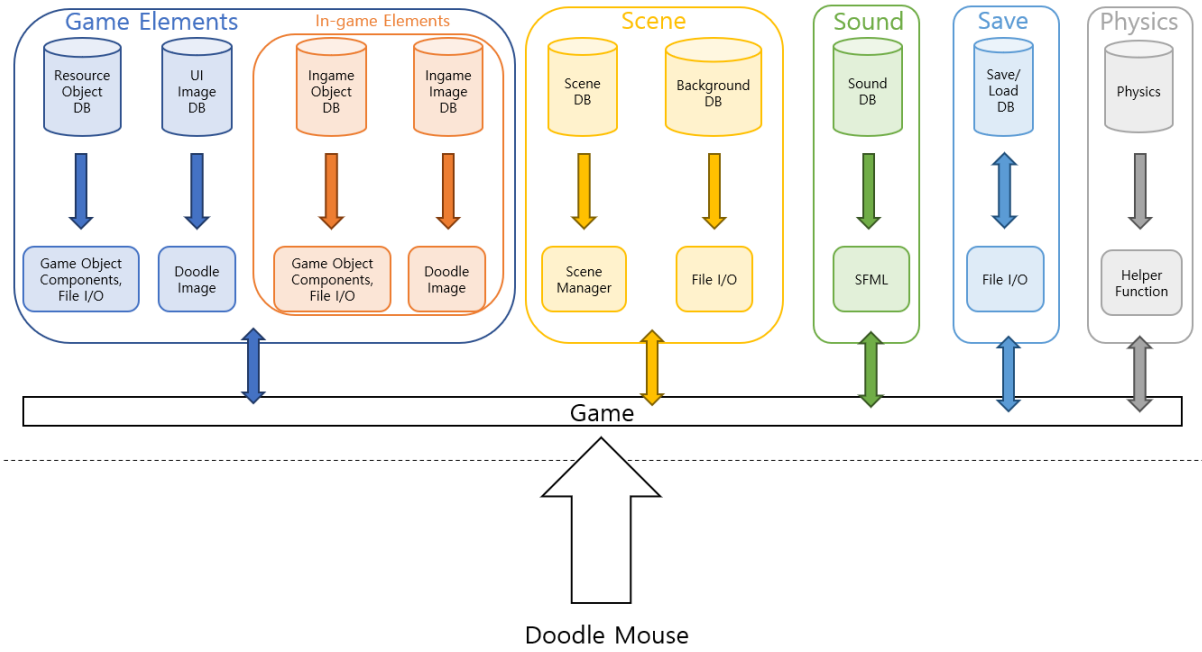
Technical Specification

GAM150S21KR
Spring, 2021
MuYaHo

Producer: Daehyeon Kim: Main
Technical Director: Junsu Jang: In-game
Lead Designer: Taeju Kwon: Support
Test Manager: Jihyeon Song: Support

Table of Contents

Architecture Overview



Graphics Implementation

❑ **Image Load**

Since the number of button images is so many, loading and drawing them using Doodle takes a delay due to the time to recall the image. To preload the images and take them out when needed, Save image in <map> with string (name of button), button image is preloaded via image set function using inline in top_header.

❑ **Button status**

In order to draw a different image when a button is pressed or not pressed, a condition was added in the draw_function of the Button class to separate when the button was pressed and when it was not pressed.

❑ **Draw each chapter's image**

It's hard to draw images one by one because there are many chapters. After unifying the file name, the image was called using to_string (now chapter).

Physics Implementation

- ❏ **Movement for the player: character**

  In the in-game, each turn's character automatically moves on the board according to the eyes of the dice rolled by the player. Each character has a **tile number variable** that is currently located, so it moves by adding or subtracting that number. Using the characteristics of the board game, the character visually moves in **one tile at a time**. And when the character reaches a tile that calculated the number of dice eyes, it finally stops.

- ❏ **Movement for interactable objects**
  - ❏ Dice: A pair of dice shows a rolling animation and randomly displays the results when the player clicks on them with the mouse.
  - ❏ Token: Tokens can be freely moved by the player dragging with the mouse.

- ❏ **Collision of in-game objects**

  When the player chooses one token, each area was designated to prevent conflicts so that tokens do not overlap.

Player Controls Implementation

- ❏ This is a **single-player game**, the player can control with only the mouse left button.
  There is no chance to control without the mouse left button.

Behavior Implementation

- **In-game object behavior techniques**
  - The character and tile has a number currently located and interacts with the objects that the number of tile has.
  - Character classes have dice and tokens, and interacting dice and tokens gives an action effect to the character. Depending on the effect, the switch does the following: Move, Attack, Get mana, Magic Attacks, …
    - Move: Add the number of dice to the number the character has.
    - Mana: Give mana to Mage. If Mage has
    - Attacks: The player can select tiles to the extent of the character's attack range. An attack object according to the selected number of the tile has.
  - At the end of the player's turn, a common monster checks for a character in the same tile as him, calculates his physical strength, and the character's defense to cause damage.
  - Boss Monster has a pattern state, draws, updates according to the state, waits for a set amount of time when the pattern is finished, and sets the next pattern.
  - The peddler and the magic tower have number, and the player can buy magic if the character has the same number as the peddler's or the magic tower's number.
  - A dice has a three-dimensional vector and uses a matrix to implement motion in which the die rolls.
  - A token can recognize the mouse's click state. And it can be dragged depending on the click state.
  - Common monsters in later chapters use the state system of the Boss Monster to obtain a small pattern.

- **Distinct types of behavior**
  - Components have the number of tiles they are located in and compare the numbers to interact.
  - Describe the techniques used for these distinct behaviors.
    - Put similar elements in one array and compare the number in the entire array.

- **How objects operate together**
  - Allows interacting with each other through globally generated quests or Player Status.

Coding Methods

- **Coding conventions**
  - **File naming conventions**
    - Image: no naming convention, but it must be in the same folder by type.
    - Data: It should say data at the end of the name.
  - **File locations**
    - Image: At similar type folder in assets folder.
    - Data: At dataFile folder
  - **Code formatting**
    - When making a function, It must be in the namespace that shows the type of work.
    - The names of class and struct must begin with a capital letter.
    - New things, such as classes, structures, functions, variables, etc., must be explained by name alone. (If it can't, some comments must be needed)
  - **Code documentation (commenting)**
    - At the top of the file there is some comment to explain the file and related Person.
    - If new things cannot explain themselves, it needs some comments to show itself. (That can be omitted if the name is sufficient.)
  - **Variable naming conventions**
    - Global variable: lowerCamelCase
    - Local variable: lowerCamelCase
    - Class and Struct member variable: lowerCamelCase
  - **Function naming conventions**
    - Global functions: UpperCamelCase
    - Class and Struct member functions: all_lower

- **Source control system**
  - We use GitHub, and Sourcetree.
  - **Any rules team has.**
    - When someone finishes and pushes it on GitHub, they upload it to the Team Discord Push chat channel to let others know.
    - Materials that teammates didn't create themself leave the source in the source channel.
    - Team members quickly ask questions about other tasks so that other tasks do not interfere with the project.
    - Team members do not arbitrarily touch the code created by other team members.

Debugging

- ❏ There is console debugging that allows players to see what items they have acquired.
- ❏ The FPS can be found in the Setting Scene.
- ❏ There is no in-game console or additional visual drawing, and debug on/off is not possible.

Tools

- ❏ Visual Studio Compiler
- ❏ Doodle
- ❏ SFML

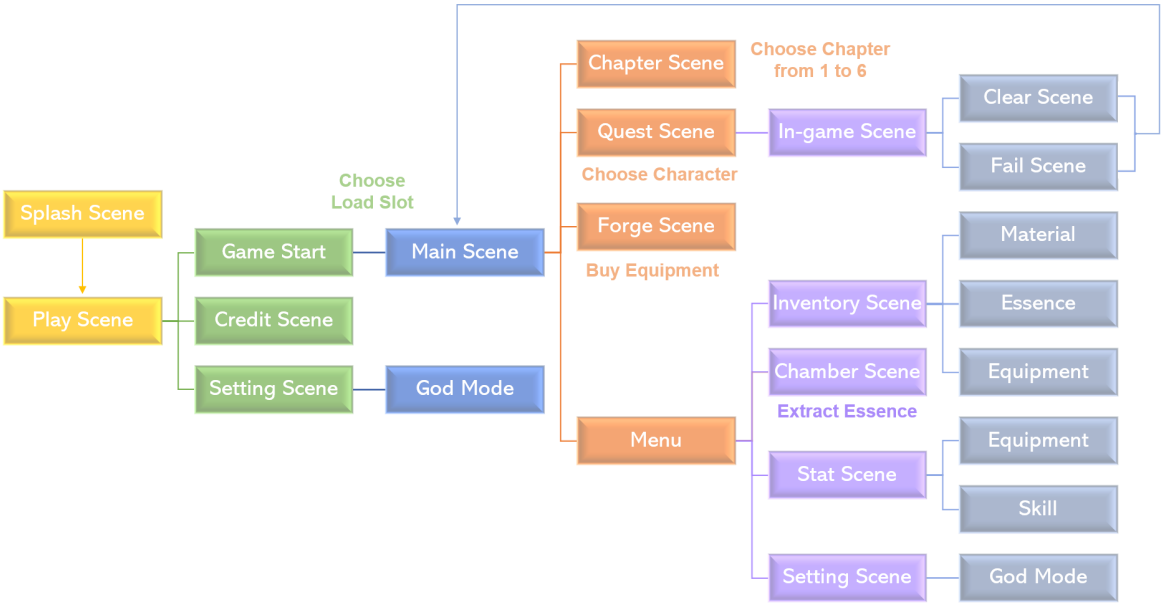
Scripting Languages

- ❏ C++
- ❏ txt

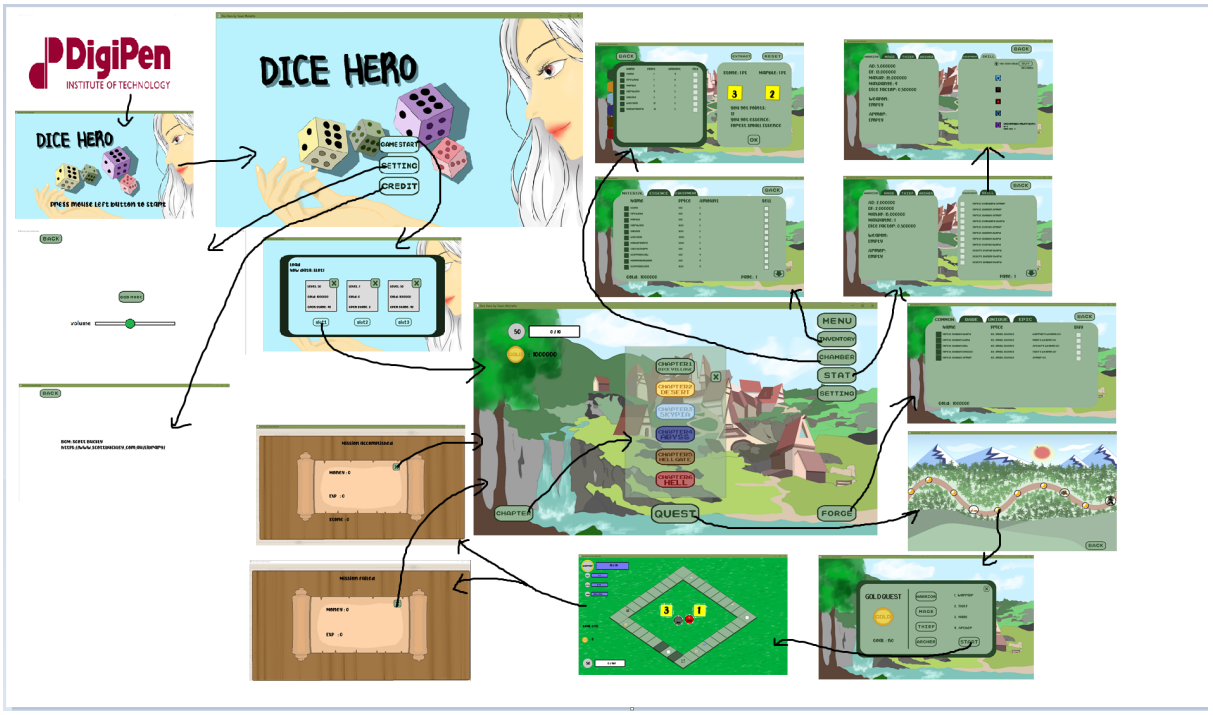
Technical Risks

- ❏ **Technical risks to the project**
  - ❏ As the number of elements increases, the number of elements must be compared, so the more elements, the longer the action can take.
    - ❏ Makes it possible to compare and finish only what is needed with a faster comparison algorithm.
  - ❏ If created after a comparison, there may be bugs caused by the inability to compare newly created objects.
    - ❏ Sequences generation and comparative behavior, and implements functions in the order given.
  - ❏ If all the character's behavior effects are handled with switches in In-game scene, the code of In-game Scene will be too long, making it difficult to maintain and change.
    - ❏ Rewrites the code so that what the class can do on its own can be done inside the class.

Flowchart



Mockups



Art Requirements

- ❏ File is .png file.

- ❏ **Background**

  6 chapters for 3 scenes(18) and the Start scene and Clear/Fail scene.
  Total 21 background images.
  File name is (Role Name)_Background.png
  All made by Taeju

- ❏ **UI Frame**

  Inventory - 6 chapters for 3 different information(18)
  Chamber - 6 chapters for 6 different information(36)
  Forge - 6 chapters for 4 different information(24)
  Stat - 6 chapters and 8 different information(48)
  Announcement Board - 6 chapters
  Quest Sign Board - 6 chapters
  File name is (Scene Name)-(Chapter Number).(Information Number).png
  For example, Chapter 4 UI for Forge scene, first information will be forge-4.1.png
  All made by Jihyeon

- ❏ **Buttons**

  6 chapter for 37 kinds of button for pressed/released(444)
  File name is (Role Name)(Chapter Number).(Pressed = 1, Released = 2).png
  All made by Jihyeon

- ❏ **Icons**

  Material - Will be assets.

  Essence - Jihyeon Made. File name is essence-(Chapter Number).(Value).png

  Equipment - Jihyeon Made. File name is (Equipment Name)-(Chapter Number).(Value).png

  Skill - Assets https://opengameart.org/content/random-rpg-icons-part-1, CC-BY 3.0

  File name is (Character)(Skill Tier).png

  For example, Warrior's first skill icon is Warrior1.png

Audio Requirements

- ❏ Sounds are .ogg file.
- ❏ BGM file name is (Scene or Chapter Name)_BGM.ogg
- ❏ Effect Sound file name is (Behavior or Situation Name)_ES.ogg
- ❏ BGM from https://www.scottbuckley.com.au/library/
  must credit this
  [Music by] or [Additional Music by] or [Track Title] by Scott Buckley – www.scottbuckley.com.au
- ❏ **Effect Sounds**
  - ❏ Own SFX Click Sound
  - ❏ Male Character Dying Sound - CC 0.0
    https://freesound.org/people/Under7dude/sounds/163442/
  - ❏ Female Character Dying Sound - CC-BY 3.0
    https://freesound.org/people/AmeAngelofSin/sounds/345049/