

**De La Salle University- Manila**

**Gokongwei College of Engineering**

LBYCPA1

Programming Logic and Design Laboratory

Project Proposal

Realm's Fate: The Ultimate Test of Choices

Erwin Adrian P. Alfonso

Ethan Vincent Ryan B. Beleno

Ralph Christian B. Tejada

**Project Description**

Give an overview of the project. Describe the problem to be solved. Enumerate the technical objectives. Show how the project will be done.

The "Realm's Fate: The Ultimate Test of Choices" game is a text-based adventure game that allows players to take on the role of a king and make decisions that affect the fate of their kingdom. The game is inspired by the popular game "Reigns," but implemented in Python as a command line interface.

Problem to be solved:

The game aims to provide players with an engaging and challenging experience that tests their decision-making skills and strategic thinking. The game also aims to provide an educational experience by allowing players to learn about the complexities of governing a kingdom, the impact of political and social decisions, and the importance of balancing competing interests.

Technical objectives:

The technical objectives of the project include:

Creating a text-based interface that allows players to interact with the game using simple commands.

Developing a game engine that generates scenarios and choices based on the player's previous decisions and the current state of the kingdom.

Implementing an artificial intelligence algorithm that simulates the behavior of the kingdom's various factions, including the nobility, the military, the clergy, and the common people.

Developing a scoring system that tracks the player's progress and awards points based on the length and success of their reign.

How the project will be done:

The project will be implemented using the Python programming language, which offers a variety of built-in modules and libraries that are well-suited for game development. The team will use an iterative and agile development approach, with regular testing and feedback sessions to ensure that the game is meeting its technical objectives and engaging the target audience.

**IPO**

Present and discuss the IPO model.

*Input:*

1. Player Choice: This is the input from the player, where they choose between two options presented by the game. The input will be a number, either 1 or 2, representing the option chosen by the player.
2. Player Name: This is an optional input from the player, where they can enter their name to be used throughout the game.

*Process:*

1. The game will read the player's decision and evaluate its impact on the kingdom.
2. It will check if the decision is valid and whether it aligns with the player's overall objectives.
3. It will update the game state based on the decision and calculate the consequences of the action taken by the player.
4. It will provide feedback to the player regarding the outcome of their decision.

*Output:*

1. Scenario: This is the output presented by the game to the player, where the game presents a scenario and two options for the player to choose from. The scenario will be presented as a string.
2. Kingdom State: This is the output presented by the game to the player, showing the current state of the kingdom based on the player's choices. The kingdom state will be presented as a dictionary, with keys representing the four key pillars of society - the military, the economy, the people, and the church - and values representing the current state of each pillar.
3. Game Over: This is the output presented by the game to the player when the player's reign ends. The game over message will be presented as a string.
4. Reign Length: This is the output presented by the game to the player, showing how long the player's reign lasted in years. The reign length will be presented as an integer.

Discussion:

The input to the game is the player's choice and their name, if they choose to enter it. The player's choice is a number representing the option chosen by the player, which will be used to determine the next scenario and the state of the kingdom. The player's name is optional and will be used to personalize the game experience for the player.

The game evaluates the player's decision and checks if it is valid and aligns with their objectives. It updates the game state and calculates consequences based on the decision. The player is provided with feedback on the outcome.

The game's output includes the scenario, the kingdom state, the game over message, and the reign length. The scenario is a string representing the scenario presented to the player, which will be generated based on the player's choices. The kingdom state is a dictionary representing the current state of the kingdom, which will be updated based on the player's choices and used to determine the game's next scenario. The game over message is a string presented to the player when the player's reign ends, which will signal the end of the game. The reign length is an integer representing how long the player's reign lasted, which will be calculated based on the number of choices made by the player.

**Methodology**

How are you going to do it? Present an overall system flowchart on how your project should work. Include a description of Python concepts that will be used to develop the project.

To implement the game, we will use a decision-tree structure and several programming concepts, including conditional statements, loops, and functions. The overall system flowchart for the game is as follows:

1. The game starts by presenting a scenario to the player and displaying two options for the player to choose from.
2. The player inputs their choice, either 1 or 2, which is evaluated using conditional statements.
3. The game updates the state of the kingdom based on the player's choice, which affects the balance between the four key pillars of society.
4. The game checks the balance between the four key pillars of society and adjusts them based on the player's choice. For example, if the player chooses to focus on the military, the game might increase military strength at the expense of the economy.
5. The game checks the length of the reign by counting the number of choices the player has made, with three choices equaling one year. If the player's reign ends, the game generates a new monarch and restarts the game.
6. The game generates a new scenario and repeats steps 1-5 until the player's reign ends.

To accomplish this, we will use several Python concepts:

1. Conditional statements: We will use if/else statements to evaluate the player's choice and update the state of the kingdom accordingly. For example, if the player chooses option 1, the game might increase military strength at the expense of the economy.
2. Loops: We will use a while loop to iterate through multiple scenarios and keep the game going until the player's reign ends.
3. Functions: We will use functions to modularize our code and make it more readable and maintainable. For example, we might have a function that generates a new scenario, or a function that checks the balance between the four key pillars of society.
4. Lists and Dictionaries: We will use lists and dictionaries to store data such as the scenarios and the balance between the four key pillars of society.
5. String Formatting: We will use string formatting to present the scenarios and other game information in a user-friendly way.

Finally, we will implement the unique gameplay feature of tracking the length of each reign based on the number of choices made by the player. This will involve counting the number of choices the player has made and converting it to years of reign, with each set of three choices representing one year.

**Schedule of Activities**

Provide a timetable or Gantt chart of your deliverables. Indicate who does what and when the deliverables will be accomplished.

| Task | Erwin | Ethan | Ralph | Duration: |
| --- | --- | --- | --- | --- |
| Planning and Design | 2 days | 2 days | 3 days | 1 week |
| Scenario Writing | 1 day | 1 day | 1 day | 3 days |
| Initial Coding | 5 days | 5 days | 5 days | 2.2 weeks |
| Testing and Debugging | 1 day | 1 day | 1 day | 3 days |
| Final Testing and Debug | 1 day | 1 day | 1 day | 3 days |
| Totale Duration: | 10 days | 10 days | 11 days | 1 month |

Planning and Design: In this phase, the team will conduct research to understand the requirements of the game, and will develop a design document outlining the game mechanics, user interface, and technical specifications.

Scenario Writing: In this phase, the team will develop a set of scenarios and choices that the player will encounter throughout the game. Each scenario will have multiple possible outcomes, and the choices made by the player will affect the state of the kingdom and the player's score. The team will also develop a scoring system that tracks the player's progress and awards points based on the length and success of their reign.

Initial Coding: In this phase, the team will begin coding the game, using the design document and scenarios developed in the previous phases as a guide. The team will create the game engine that generates scenarios and choices based on the player's previous decisions and the current state of the kingdom. The team will also develop an artificial intelligence algorithm that simulates the behavior of the kingdom's various factions, and create a text-based interface that allows players to interact with the game using simple commands.

Testing and Debugging: In this phase, the team will conduct regular testing and debugging sessions to ensure that the game is functioning as intended. The team will identify and fix any bugs or issues that arise, and will make adjustments to the game mechanics and user interface based on user feedback.

Final Testing and Debugging: In this phase, the team will conduct a final round of testing and debugging to ensure that the game is polished and is ready to be submitted as a final product.

**References**

Concept of the project:

<https://www.theguardian.com/technology/2016/sep/07/reigns-review-medieval-strategy-mobile-game-tinder>

<https://www.androidcentral.com/reigns-beginners-guide>

<https://www.tutorialspoint.com/python/python_functions.htm>