Documentation

Table of Contents

[Introduction 1](#_Toc121790365)

[Purpose 1](#_Toc121790366)

[Game Description 1](#_Toc121790367)

[Process Model 1](#_Toc121790368)

[Use case 2](#_Toc121790369)

[UML Model 3](#_Toc121790370)

[Class Diagram 4](#_Toc121790371)

[State Diagram 4](#_Toc121790372)

[Activity Diagram 5](#_Toc121790373)

[Testing Strategy 5](#_Toc121790374)

[Unit testing 5](#_Toc121790375)

[Integration Testing 6](#_Toc121790376)

[Conclusion 6](#_Toc121790377)

[References 7](#_Toc121790378)

# Introduction

The report is being written to explain the software requirement specification of Wandering the woods game as the major process structure of the project is implement the complete simulation based game also the implications and analysis of the descriptive details must be developed with respect to the controlled application management and descriptive domain of implementation (Al-Aidaros, 2013). The analytical thinking of the application could be applied with the collection of understanding with the complete orientation and implications of the controlled utilization being planned and managed in order to show the validity of in depth modeling so that the complete information and managerial utilization of application development.

## Purpose

The major purpose of application is to develop the core level understanding to utilize the process of implementation based on which the controlled planning and management of the service level results based on which the managerial model and controlled specification could be planned in order to show the complete view of service based management as per the simulation game development to complete the entire process of understanding and management of final results.

## Game Description

The game is basically Wandering into the woods, and the game is based on the health and energy such as the user start the game , then the wander through the forest , in case if the user is stuck into the trees , the energy level will be down , and game will end by showing the ending message (Bjorvatn, 2018). It is necessary to show the complete functional implementation based on which the utilization strategies could be done in order to show the proper results based on which the properly managed instructions will be defined in order to complete the process of execution to be validated and analyzed in order to show the complete strategic results of functional and nonfunctional management of game.

# Process Model

The process based management is controlled with the flow of execution to show the complete integrated management as the Agile based functions for the application has been used with respect to the complete functional and nonfunctional systematic requirements being planned and overviewed with the strategic domain of utilization.

Agile is an incremental method of project and software building that helps teams offer value to their clients quicker and with fewer difficulties. Instead than putting everything in one "big bang" debut, an agile team provides work in modest, yet consumable, increments. Requirements, strategies, and results are examined continually so companies have a natural method for reacting to changes fast.

In agile, a product is produced in a number of fixed-length rounds called sprints, offering agile teams a structure for producing code on a regular basis to make sure. Learn how the scrum technique effects traditional project management.

Kanban is a prominent agile methodology that requires real-time information of group's capacity and complete disclosure of work. Learn how the Kanban method of agile software development might benefit for your team (Bowern, 2006).

Agile project management is an incremental strategy to managing software projects that focuses on frequent releases and client feedback. Start here now for your agile transition.

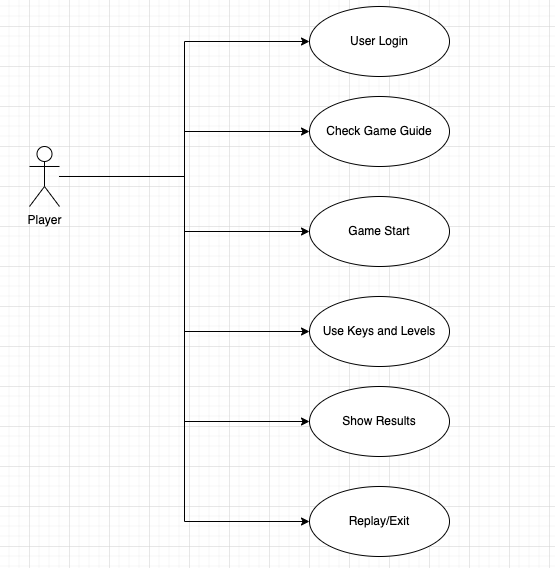
Instructions for joining an agile team as a brand manager or product owner. Discover the ins and outs of creating product roadmaps, setting feature priorities, writing product requirements papers, and making informed decisions with product analytics.

Scrum of scrums, also known as the Scaled Agile Framework, is a method for expanding the scope of an existing agile project (Ferrag, 2018). Both are excellent jumping-off locations as you expand agile throughout your company.

Both It and Agile represent cultural trends that inspire firms to achieve higher. Read here now to learn exactly what agile and DevOps interconnect.

Whereas the conventional "waterfall" model has one discipline input toward the task, then "throw it to the wall" the next participant, agile asks for collaborative inter teams. Open communication, cooperation, adaptation, & trust amongst members of the team are all heart of agile. But even though the team lead or product owner normally prioritizes the work to just be delivered, their team takes the lead on selecting how it work would get accomplished, self-organizing around specific tasks and responsibilities (Konsky, 2007).

# Use case



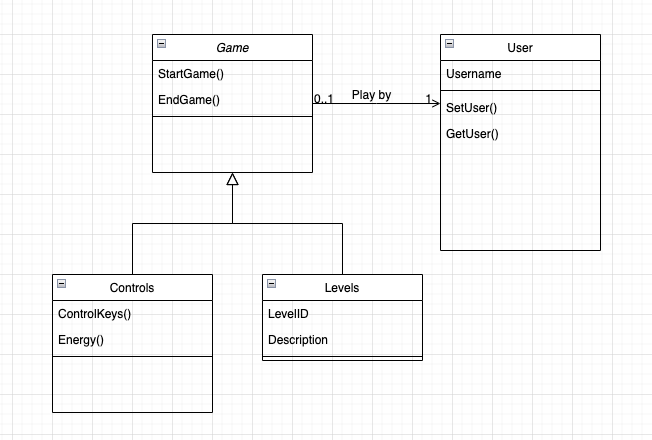
# UML Model

The Unified Modeling Language (UML) is a modelling language that can be used for various purposes. The basic purpose of UML is to provide a standard approach to visualize the way a systems has been designed. It's very close to the plans used in other branches of engineering. UML is not a software program; it is rather a symbolic representation. UML diagrams are used to depict a system's behavior and architecture. UML is useful for modelling, designing, and analyzing software and systems for business professionals, architects, and engineers (Kumar, 2017).A streamlined method of communication is essential for the successful development of complex applications requiring the input of several teams.

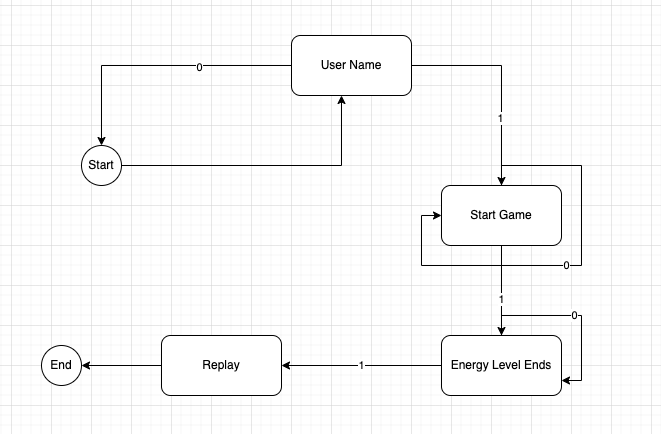
Merchants are unable of comprehending computer language. This means that UML is crucial for conveying the system's core requirements, features, and procedures to non-programmers. A huge amount of time is saved later on when organizations are able to visualize processes, interactive features and the system structure (Musa, 2016).

Object oriented analysis and design are inextricably tied to UML. UML makes usage of elements and develops connections among them to create diagrams.

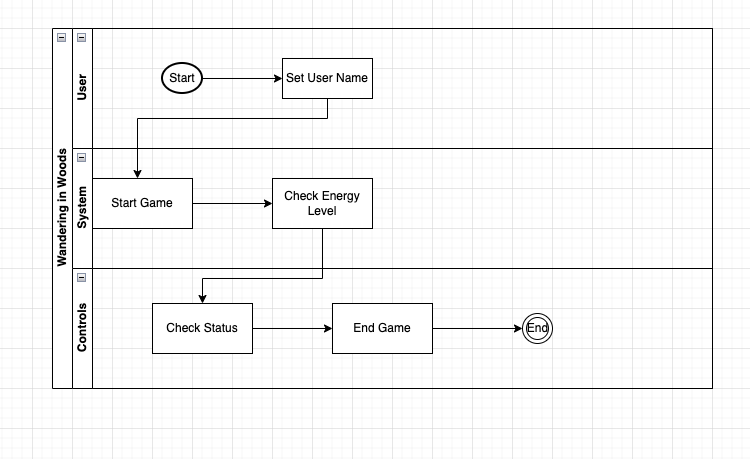
## Class Diagram



## State Diagram



## Activity Diagram



# Testing Strategy

## Unit testing

With jUnit we will test the single methods that represent the smallest testable pieces of the application. The methods will be evaluated as they are generated to validate the framework as it is developed. In object-oriented programming, the class is considered as the smallest unit, although the smallest testable portion remains the method of the classes. Hence, independent of your practices remain the target of unit tests. However, the tactics in testing must alter with OOP as it may be impossible to test a single action in solitude resulting in the necessity to test sequences or methods included in a class. Also, the team will have to employ scenario or lace testing if the class is the smallest tested unit (Schwarting, 2018).

A decent test case for a functional testing will contain a number and title to distinguish it, as well as brief explanation of the test. A record of both the actual and expected outcomes is kept. The test is considered successful if the observed results are in line with the predicted results. Alternatively, if the real and expected results do not equal then the test is reported failed & indicates a probable issue in the system. Last, the test table must include a column that shows the requirement or user narrative is linked with the testing ground.

To perform white box unit testing, a developer would construct test cases for each method created for the classes to ensure that their respective logic routes are followed. For this kind of unit testing, the focus of the test cases will be on the black box side, where it will be ensured that the pre and post- conditions for every method are met and that the actual way outputs fit the expected method outcomes for the set of inputs.

## Integration Testing

When evaluating an application's API, integration testing is performed to guarantee that not only do all API calls succeed, but also that the methods themselves function as intended. Often, test suites of system testing are done at a routine period as part of an automated testing process, which is how integration testing is performed. Because of the many tests and dependencies between the various pieces of software, it is often impossible for a single engineer to run an entire integration test suite manually (Ul Musawir, 2017).

If you want to find and fix problems quickly and completely test all of a project's components, integration testing should be done in small, regular increments. Top-down testing, bottom-up testing, and continuous testing are just a few of the incremental integration testing methods available. In this case, the team has decided to use a continuous integration strategy, wherein the components of the ever-changing software are merged into the latest daily build of the project in accordance with the team's evolutionary process model.

When performing integration testing, the use cases would be examined individually, and the interactions between the classes required to complete each use case would be verified. In order to demonstrate that the evolving model doesn't quite fail in unpredictable ways, it will be necessary to create chains of API calls that are necessary to finish each use case appropriately. QA testing will be conducted once each use case is incorporated into the prototype to make sure no new bugs have been introduced. Here is where the automation testing will really shine for the development team.

# Conclusion

It is concluded that complete details about simulation game development have been explained to measure the complete requirements of the system to develop the context of analysis with respect to manage the complete development of the application. A complete object oriented approach has been used to measure the integrated analysis with respect to the service level implementation of functional system requirements. All classes and UML modeling are well mapped to show the complete systematic requirements of project.

# References

Ferrag, M. A. (2018). Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes.

Konsky, v. (2007). Engaging undergraduates in discussions about ethics in computing.

Bowern, M. (2006). ICT Integrity: Bringing the ACS Code of Ethics up to date. Australasian Journal of Information Systems.

Al-Aidaros, A.-H. (2013). Ethics and ethical theories from an Islamic perspective. International Journal of Islamic Thought.

Schwarting, W. (2018). Planning and decision-making for autonomous vehicles. Annual Review of Control, Robotics, and Autonomous Systems.

Kumar, A. (2017). A review of multi criteria decision making (MCDM) towards sustainable renewable energy development.

Bjorvatn, T. a. (2018). Project complexity and team-level absorptive capacity as drivers of project management performance.

Ul Musawir, A. (2017). Project governance, benefit management, and project success: Towards a framework for supporting organizational strategy implementation.

Musa, A. (2016). A review of RFID in supply chain management: 2000–2015. Global Journal of Flexible Systems Management.