**Design Document:**

**Card game of War**

**by Wes Brown**

**8/4/2017**

**TABLE OF CONTENTS**

1 Identification 3

1.1 Document overview 3

1.3 References 3

2 Software Development Activities 3

2.1 Software development process 3

2.1.1 Overview of process phases 4

2.1.2 Lessons Learned section 4

2.1.3 Technical documentation 4

2.2 Software development tools 5

2.2.1 Workstation 5

2.2.2 Requirements management and documentation 5

2.2.3 Software Design 5

2.2.4 Coding and automated tests 5

2.2.5 Configuration management 5

2.2.6 Obsolescence management 5

3 Development Process Phases 5

3.1 Software Specifications 5

3.1.1 Input data 5

3.1.2 Content 5

3.1.3 Output data 6

3.1.4 Review and acceptation criteria 6

4 Responsibilities 6

4.1 Activities and responsibilities 6

4.2 Documentation of Activities and responsibilities 6

# Identification

There is a need for a digital game of war card game. In collaboration with all the various teams here at our company, our task is to design and create a functioning version of the card game war

## Document overview

This document will outline the design, development, testing, implementation and on going support of the game of war software solution. The below sections will detail out an overview of how to bring this product to market.

# Software Development Activities

The section list and describes the software development activities of the card game of war project. The below section will provide all the team members a master document to work from that can be updated by them as needed, to make this a living document of the project

## Software development process

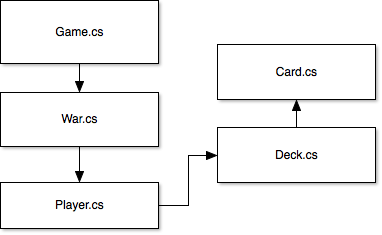
The software development process will use the SCRUM process. The development team, in conjunction with the project managers, business analysts, QA, deployment and other management groups will build sprints for rapid design, development and deployment of this card game solution

### Overview of process phases

Below is a detail overview of the SDLC process for this project. The below will detail out the specification process, a detailed design section, a coding section and unit test section, an implementation section and finally a verification section.

* Software specification and design review:
  + The technical team decided to use an Object oriented architecture written in the C# computer language to build this card game solution. This solution will be able to be executed on multiple platforms as well as make it easy for users to run the distributed version of the game.
  + This solution will be a command line based game to begin with. Perhaps a phase 2 project will include a GUI front end to this solution.
  + We are going to use the Mono complier to compile our code into the final executable.
* Software detailed design:
  + Order of operation for the game of war:
    1. Take deck and shuffle.
    2. Deal 26 cards to each player 1 and player 2.
    3. Player 1 flips over and exposes a card value.
    4. Player 2 flips over and exposes a card value.
    5. Who ever as the higher card value wins. Example: 5 of clubs, beats a 4 or better.
    6. A face card (jack, queen, king) is worth 10 and an ace is worth 11.
    7. Suit does not matter in this game.
    8. If the cards tie in value, then there is a war.
    9. Each player flips over 2 cards and on the 3rd card the winner is determined.
    10. Who ever has the largest value wins, if there is a tie then another repeat step 9.
    11. Game continues as above until a player no longer possesses any cards.
    12. If war is declared and one player does not have enough cards to go to war the other player wins.

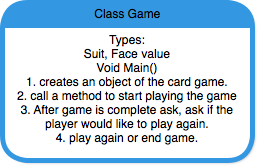
Below is a flow chart of the objects that are to entail the source code for the card game of war. The arrows indicate the inheritance of each object an it’s relation to the parent class. Because the game is simple at this point the developers have left a lot of the classes public to assist in easy access to attributes and methods of those classes. Future development may require encapsulating the classes and privatizing the elements and methods to avoid inappropriate access.



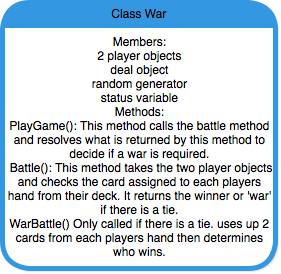
* Software coding:

Below is another diagram listing the technical layout of each class, The attributes and methods as well as a brief description of the objects function.

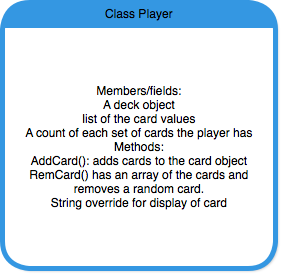
* + Below is the class Game. It is part of the namespace WarGame. This name was left generic to facilitate other games to be added to this object later.



* + Below is the class war. It is part of the namespace WarGame. This class is the heavy lifter of the program. This class creates 2 players and a deck of cards. Then assigns 26 cards to each player and proceeds to compare hands of cards and declares winners until one player runs out of cards and a game winner is declared.



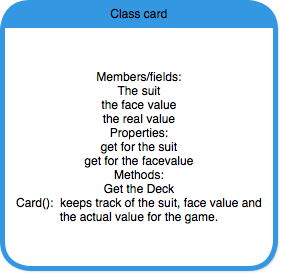
* + Below is the class Player. This class will deal with player related objects such as the deck of each player, the cards and string displays.



* + Below is the class is the deck object. It will be responsible for assigning the cards to the players’ decks and shuffling the cards so they are random to each player.



* + Below is the class is the card object. Assigns the suits, face values and real value of each card to help the other objects make decisions about player winning cards.



* Unit Test plan.

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Test Plan | |  |  |
| Test # | Description | Outcome | passed/failed |
| 1 | Deck is shuffled correctly | Each shuffle is random | Passed |
| 2 | Each player is assigned 26 cards | Each player object has 26 cards | Passed |
| 3 | each player pulls a card from their deck and they are compared and one player either wins or there is a war | each player card is value is compared correctly or war is declared | Passed |
| 4 | War plays out with 2 cards set down then use the 3rd card to determine winner. If tied, war again | Correctly determines winner for war or if there is another need for war | Passed |
| 5 | Once a player is out of cards the other player wins. | Yes | Passed |
| 6 | Logic to ask for a new game works | Yes | Passed |
| 7 | If one player is does not have enough cards to play through a war cycle then the other player wins | Stack overflow. Program bombs | Failed |

* Software integration: Determine a date and time to deploy this solution first to the test environment for the QA team to run tests. Then, determine a good time to deploy to a location for people to purchase the software package. Have managers determine timelines for this final deployment.

### Lessons Learned section

This section of the design document is to document post go-live evaluation of the project. This section will also capture ways to improve the process and issues that arose during the project life cycle.

After then end of the project here are some items of review:

* Software specification: Did we gather the requirements in an adequate way?
* Software detailed design: Was the documentation provided well enough to help the developers build the solution?
* Software coding and unit tests: Was the development cycle quick enough and was the solution delivered on time and budget? Were there enough tests performed to reduce bugs in the release?
* Software integration: Was the solution deployed on time and during a time that did not impact users or players?
* Software verification tests: Did the test run though enough cycles to work out bugs?
* Final verification test review: Did everyone feel the project was completed well?

The planning of phases and reviews is given below OR is in the project management plan.

### Technical documentation

The following documentation is produced during the design phases:

* See the design portion of this document. Some attachments may be included if required.

### Deliverables:

The following items are delivered at the end of the process:

* Design documentation,
* Technical documents
* Project timelines
* Management expectations
* Code and final package
* Test results
* Deployment plan
* Support plan

### Workstation

Multiple platforms.

### Coding and automated tests

Used various tools for development. Testing used packaged software to run tests.

### Configuration management

Describe tools used for configuration management and bugs management:

* Used CVS as code repository.
* Used Bugzilla to track bugs during project and after project

# Development Process Phases

See section 2.

Overview of process phases

## Software Specifications

### Input data

Spec numbers can be listed and linked to here in the document repository.

### Content

This section is to link all the documentation on various drives to this one document:

* Functional requirements documents are listed here.
* Developer notes.
* Project manager notes
* Business analyst notes.
* User documentation.
* end user notes.

# Responsibilities

Here is a brief responsibilities section of the SCRUM cycle so that ownership can be established.

## Activities and responsibilities

Each activity has someone responsible.

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Responsibility** | **Service/Department** | **Comment** |
| Project management | Project Manager | Build out the timeline | Document provided and updated as needed |
| Configuration tools management | QA, testing and deployment of solution | QA and configuration team | Need input from management |
| Setting up the Development tools | Configuration Manager | Config department | Set up timelines and deployment times |
| Software specifications | Development team | Dev team | None |
|  |  |  |  |

## Documentation of Activities and responsibilities

The person responsible for writing a document is not necessarily the person responsible for the activity. Generally, the last approves what the first did. For small teams, tech control and approval may be the same person.

| **Activities** | **Document name** | **Elaboration** | **Technical Control** | **Approval** |
| --- | --- | --- | --- | --- |
| Gather requirements | Design doc | none | Developer | team |
| Flow charts | Design doc | none | Business analyst | team |
| System design | Design doc | none | Business Analyst | team |
| Project plan | Design doc | none | Project manager | team |
| Development plan | Design doc | none | Systems Engineer | team |
| Cost and timeline summary | Design doc | none | Management Team | Managers |
| Development plan and Development | Design doc | None | Development team | IT manager |
| Testing | Design Doc | None | QA and Development team | QA |
| Implementation | Design Doc | None | Deployment and QA | Managers |
| Monitor system | Design Doc | None | Development and Support Team | Support team and management |