

Project Management Plan

for

<Uno>

Version 1.0 draft 1

Prepared by <Sean Coady, Mark Sowles>

<CS 492>

<26 Feb 2014>

Table of Contents

Overview	1
Project Purpose, Objectives, and Success Criteria	1
Project Deliverables	1
Assumptions, Dependencies, and Constraints	1
References	ii
Definitions and Acronyms	ii
Evolution of the Plan	ii
Project Organization	ii
External Interfaces	ii
Internal Structure	ii
Roles and Responsibilities	ii
Managerial Process Plans	iii
Start-Up Plans	iii
Estimation Plan	iii
Staffing Plan	iv
Staff Training Plan	iv
Resource Acquisition Plan	iv
Project Commitments	iv
Work Plan	iv
Control Plan	v
Data Control Plan	v
Requirements Control Plan	v
Schedule Control Plan	v
Budget Control Plan	v
Communication, Tracking, and Reporting Plan	v
Metrics Collection Plan	v
Risk Management Plan	v
Issue Resolution Plan	vi
Project Close-Out Plan	vi
Technical Process Plans	vi
Process Model	vi
Methods, Tools, and Techniques	vi
Configuration Management Plan	vii
Quality Assurance Plan	vii
Documentation Plan	vii
Process Improvement Plan	vii

1. Overview

< This project's objective is to create a fun, interactive game that has little to no errors. The success of this project will be evaluated on several criteria; such as difficulty, errors in code, and effective implementation of the real-world game's logic and rule set. The schedule will be defined in later sections. >

1.1. Project Purpose, Objectives, and Success Criteria

<The purpose of this project is to create a game that is fun to play, logically accurate, and easy to understand. The needs that are to be satisfied are to basically follow the rules of the actual card game, and to ensure that we can effectively implement them into our game. End users will judge the effectiveness on our game on how closely our software emulates the real game. No other integration with other outside product.

- *JDK/JRE (current version)*
- *No shared code – all will be original*
- *No dependencies on hardware other than having JRE installed*
- *Schedule dependencies: Product launch by the end of the semester >*

1.2. Project Deliverables

<List the major items to be delivered to the customers, subcontractors, integrators, or other parties. As appropriate, list the deliverables, their recipients, interim and final delivery dates, and delivery method. A table like the one below is a good way to show this information.>

Deliverable	Recipients	Delivery Date	Delivery Method	Comments
SRS Document	Dr. Zhao	09/13	Electronic	--
Object Oriented Analysis	Dr. Zhao	10/13	Electronic	
Prototype	Dr. Zhao	12/13	Electronic	--
Project Management Plan	Dr. Zhao	3/2/14	Electronic	--
Finished Code	Dr. Zhao	4/1/14	Electronic	
Finished Testing	Dr. Zhao	4/15/14	Electronic	
Deadline	Dr. Zhao	5/7/14	Electronic	
Presentation	Dr. Zhao	5/7/14	In Person	

1.3. Assumptions, Dependencies, and Constraints

Assumptions are as follows:

- *A1 User knows how to basically operate and control a Graphical User Interface (GUI)*
- *A2 User has the latest JRE installed on their machine*

- *A3 User has a working knowledge of the rules of our game>*
- *D1 Both members will contribute equally to the success of our project*
- *D2 Both members will be versed well enough in Java to ensure close to error-free code*

1.4. References

-Will add as references become necessary

1.5. Definitions and Acronyms

GUI – graphical user interface

JRE – Java Runtime Environment

JDK – Java Development Kit

1.6. Evolution of the Plan

< We will have bi-weekly meetings to discuss the scheduling of our project's progress. We will schedule updates during these meetings, and will incorporate these updates into our plan as needed. We will make changes to our requirements document, should the needs of our project change at any time. Project reviews will be held 2x / month, to evaluate the progress of the current state of our project. After source code has been completed, we will change from bi-weekly meetings, to weekly meetings, to discuss how we will proceed from that point on.>

2. Project Organization

<This section describes interfaces to entities outside of the project, identifies the internal project structure, and defines roles and responsibilities for the project.>

2.1. External Interfaces

<Describe the organizational boundaries between the project and external entities. Define and describe communication with senior management, customers, subcontractors, purchasing, sales, marketing, legal, finance, procurement, installation and support organizations, standards or certification bodies, auditors, manufacturing, and the like.>

2.2. Internal Structure

<Describe the internal structure of the project organization, including interfaces between the units of the software team. It might be helpful to include organization charts or matrix diagrams to illustrate lines of authority, responsibility, and communication. Identify representatives of key units, such as senior management, engineering support functions (configuration management, quality assurance, verification and validation), and process improvement.>

2.3. Roles and Responsibilities

<List the major project team roles and the individuals who will fill these roles, along with the specific responsibilities those individuals will have. Identify the organizational units or project team roles that are responsible for all major work activities and supporting processes. Consider the following list of potential project roles, adapted to your organization's local terminology:

- *Project Manager*
- *Product Manager*

- Technical Lead
- Software Lead
- Hardware Lead
- Architect
- Systems Engineer
- Requirements Analyst
- Software Engineer
- Hardware Engineer
- Test Engineer
- Configuration Control Board
- Configuration Management Manager or Coordinator
- Quality Assurance Manager, Coordinator, or Engineer
- Technical Applications Support
- Subject Matter Expert

Identify other internal and external project stakeholders who are not specifically members of the project team. Describe their relevance to the project and their degree of interaction for specific project activities. Potential topics to address regarding stakeholders include:

- *List of all relevant stakeholders and the rationale for each stakeholder's involvement*
- *Project roles and responsibilities of stakeholders during each life-cycle phase*
- *Relationships between stakeholders*
- *Relative importance of each stakeholder to project success by project phase*
- *Resources (such as training, materials, time, or funding) needed to ensure adequate stakeholder participation*
- *Schedule for phasing of stakeholder participation*

3. Managerial Process Plans

<This section defines the various project management plans and activities for the project. >

3.1. Start-Up Plans

<We have established a timeline of deadlines, and will try to strictly abide by these deadlines as time permits. There are many factors that could possibly affect the viability of these deadlines, and will make changes to them as needed.>

3.1.1 Estimation Plan

Using an empirical estimation method as an organic product, with 4kLOC, our estimation is as follows:

(rounded)

$$\text{Effort} = 2.4 * 4^{1.05} == 10$$

$$\text{DevTime} = 2.5 * 10^{0.35} == 5$$

$$\text{Person} = 10/5 == 2$$

Translating to -- 10 person months, with a development time of 5 months, and requires 2 people.

3.1.2 Staffing Plan

< The entirety of our staffing on this project will be the sole responsibility of two team members (Mark / Sean). We will be responsible for completing everything without aid.

3.1.3 Staff Training Plan

<Mark / Sean have been trained in Java extensively , as per the requirements of the Western Illinois University Computer Science curriculum, and other independent projects. Both members have adequate experience in the Netbeans, jGRASP, and Eclipse IDEs which will be used for our code.>

3.1.4 Resource Acquisition Plan

<Resources required are few, but as follows:





























- *Development resources: Personal computer / lab computer with JRE/JDK, and either Netbeans, jGRASP, or Eclipse installed. Microsoft Office required for documentation creation. We will use Assembla for source control of our code.*
- *Test resources: A Java IDE is required for testing and debugging, which can be accessed on either of our home computers or at a lab CRC.*
- *Product resources: Minimal amount of harddrive space (<100MB), JRE installed, and either Mac OS X, Windows, or a Linux operating environment.>*

3.1.5 Project Commitments

<Each team member is responsible for doing half of the work in every aspect. For code, we will do an equal amount of writing / debugging. For documentation, we will work cooperatively. >

3.2. Work Plan

Our work plan can best be assumed by our Gantt chart:

		Task Mode 	Task Name 	Duration 	Start 	Finish 	Predecessors 
			 Concept Development	37 days	Sun 9/1/13	Tue 10/22/13	
			game concept	30 days	Sun 9/1/13	Thu 10/10/13	
			UI Concept	16 days	Tue 10/1/13	Tue 10/22/13	
			 UI Demo Development	44 days	Tue 10/15/13	Fri 12/13/13	
			Develop UI Graphics	20 days	Tue 10/15/13	Mon 11/11/13	
			Prepare UI Demo	31 days	Fri 11/1/13	Fri 12/13/13	4
			First Demo	0 days	Fri 12/13/13	Fri 12/13/13	
			 Core Development	145 days	Fri 10/11/13	Thu 5/1/14	
0			AI Development	145 days	Fri 10/11/13	Thu 5/1/14	3
1			Game Mechanics	144 days	Fri 10/11/13	Wed 4/30/14	3
2			Information Menu	46 days	Wed 10/23/13	Wed 12/25/13	3,4
3			Internal Testing	20 days	Fri 4/4/14	Thu 5/1/14	
4			Game Demo	0 days	Mon 5/5/14	Mon 5/5/14	
5			Prepare Game Demo	11 days	Mon 4/21/14	Mon 5/5/14	

3.3. Control Plan

<This section describes how the project will control and report on the project status and activities. Specify the frequency at which the various project status indicators are to be monitored and specific events that could trigger a status evaluation.>

3.3.1 Data Control Plan

<Project files will be accessible to all team members using Assembla. No private data is at risk.>

3.3.2

<Requirements and deadlines to be set by the customer, and confirmed after review by the team members.>

3.3.3 Schedule Control Plan

<Schedule will be dictated by the work plan outlined in the Gantt Chart. All team members will share the workload for each task, and for completing documentation. Team members will meet and communicate several times weekly.>

3.3.4 Budget Control Plan

<As this is for academic purpose, we neither have a budget, or a plan to make/manage one.>

3.3.5 Communication, Tracking, and Reporting Plan

Type of Communication	Communication Schedule	Typical Communication Mechanism	Who Initiates	Recipient
Status Report	every Friday	team meeting	Project Manager	Mark/Sean
Schedule and Effort Tracking Report	weekly	phone	Project Manager	Mark/Sean
Project Review	bi-weekly	face to face	Project Manager	Mark/Sean
Requirement Changes	as needed	face to face or e-mail	Project Manager	Mark/Sean

3.3.6 Metrics Collection Plan

N/A

3.4. Risk Management Plan

< - Adhering to the schedule: We will make every effort to be on time with our scheduled deadlines, but if we do happen to miss one, we will have a plan to re-schedule this section of the project before falling too far behind.>

- **Communication** - *We will ensure that the proper communication channels are open, so that we are both on the same page as far as the project status.*
- **Equal contributions** - *It is each member's responsibility to equally contribute to the success of our project. If any doubt of equal work, we will have a meeting discussing what needs to change.*

3.5. Issue Resolution Plan

<Any problems that will arise will be discussed by team members during their weekly meetings. Resolution progress will be maintained with clear lines of communication between the team members via e-mail as well as weekly meetings.>

3.6. Project Close-Out Plan

<At the end of the development process the project will be presented to the customer and a group of peers. The software will then be released for use by anyone who wishes to do so. There currently no plans to continue to support or update the product after release.>

4. Technical Process Plans

4.1. Process Model

<We will be utilizing a waterfall style process model, with phases and milestones outlined in the attached Gantt Chart.

The customer will review at the specified milestones.>

4.2. Methods, Tools, and Techniques

The software is being developed for use on Windows 7 capable machines with minimal hardware requirements.

NetBeans IDE was used for software development

GIMP for image/asset creation and editing

Microsoft Project for project charts

Microsoft Word and Google Docs for documentation.

The software will be written in Java, with standard coding and documentation conventions.

Project phases will be reviewed by the developers and the customer, with some peer reviewing.

4.3. Configuration Management Plan

<Due to the small size and team of this project all configuration management will be discussed within the team and done as necessary. Documentation will be available via Assembla for all members to log in and access at any time. Any updates are communicated within the team via email, or during the weekly meetings.>

4.4. Quality Assurance Plan

<We plan to deploy the highest quality product possible, and as such, will require an extensive QA process. Before releasing a public version of our game, we will top-down test each individual section and sub section of our game, ensuring that it adheres to the strict guidelines described in our SRS document, and in this document. During our bi-weekly meetings we will determine what sections of our game have been completed, so that we may begin testing of these sections. As we test our game, we will be using both white-box and black-box methods of testing so that we may minimize errors as much as possible. We will continue to use SVN for our source control, which will provide us with fallback versions of our game, should something be discovered in a later version during testing. After we have determined that we have finalized version of our game, we will request the assistance of other students, professors, and our peers to further test our game in a black-box setting. Our main focus of QA will be to be certain that our game follows the real-world rules of Uno 100 percent. The number of hours which we will spend on QA is subject to change, but at this time we are estimating about 20 hours of debugging, re-coding, and evaluating feedback. The QA process will be split 50 percent between Mark and myself.>

4.5. Documentation Plan

Document	Template or Standard	Created By	Reviewed By	Target Date	Distribution
User Manual	Custom	Mark/Sean	Peer Reviewed	Delivered on release date.	Electronic
List of Rules	Mattel	Mark/Sean	Peer Reviewed	Delivered on release date.	Electronic
Patch Notes	Custom	Mark/Sean	Peer Reviewed	Delivered with each revision	Electronic
SRS	Provided Template	Mark/Sean	Peer Reviewed	FL/13	Electronic
PMP	Provided Template	Mark/Sean	Peer Reviewed	3/4/14	Electronic

4.6. Process Improvement Plan

<Improvements will be added as needed. We will be focusing on mainly implementing the basic version of the game, but if time permits, we will be adding additional features such as multi-player capabilities, playing over a network, and other non-essential elements. If other elements are needed to perform essential functions, we will add them accordingly.>

Revision History

Name	Date	Reason for Changes	Version
sean	3/3	initial	1.0
mark	3/3	adding additional info	1.0