CS 4366: Senior Capstone Project

Backyard Garden

Educational Android Game

**Version 1.0:** February 1, 2015

**Team:** Wendy Castillo

Gregory Harrison

Travis Morrison

Table of Contents

Introduction 2

1.1 Executive Summary 2

1.2 Deliverables 2

2 Management Structure 3

2.1 Project Lifecycle 3

2.2 Roles and Responsibilities 3

2.3 Communication 4

3 Planning and Control 4

3.1 Milestones 4

3.2 Work Breakdown Structure 5

3.3 Schedule 5

4 Costs and Resources 5

5 Technologies to be used 5

# Introduction

Educational Android Game Project Abstract

## Executive Summary

## Backyard Garden will be an educational android game application that targets young children before and during kindergarten. In this game children will be able to interact with a backyard garden setting and have fun growing plants while learning basic math and science. There will be no “ending” to our game but it will have set up achievements to complete. After completing certain achievements, the game will introduce new educational concepts so that the child playing will be able to keep learning throughout the game at their own pace. We believe that this method will allow a child to desire learning rather than feel discouraged about what they are doing.

The game will begin with a friendly backyard setting with a plot of land designated for planting. The user will be prompted to place one seed in the dirt, water it, and give it some sunshine. They will then be able to see the seed grow into a seedling, and then the user will repeat the water and sunshine process. After watering the plant and giving it sunshine twice, the plant will bloom into a flower and the user will have a basic understanding of plant growth. The user will be able to count how many seeds they need with a visual representation. This will begin teaching the user how to count from 1 to 10.

Currently, there is no replica game of our project idea and we are excited to offer it to parents. We believe that they will appreciate and support our aim of teaching basic math and science to children in a fun new way. Children are becoming familiar with trending technology at a higher rate and we believe that utilizing the android market to showcase our project will help us gain popularity and profit.

For this semester, we will focus on the most basic concepts as described above. However, we envision new educational topics added to a series of this game application in the future. This will expand our consumer base and allow our current consumers to grow with us. The success of these applications could eventually lead our team to try selling our product to private and public schools and will be great to present to homeschooling families.

## Deliverables

|  |  |  |
| --- | --- | --- |
| Project Management-Related Deliverables | |  |
| Product-Related Deliverables: | | |
| SPRINT 1 | Basic Features | Basic GUI; Basic environment; |
| SPRINT 2 | Improved & Added Features | Basic Player Model; Player movement; Player controls |
| SPRINT 3 | Improved & Added Features | Planting Function; Water function, Sun light function |
| SPRINT 4 | Improved & Added Features | Advanced Player model; Advanced GUI |
| SPRINT 5 | Improved & Added Features | Advanced environment; |

# Management Structure

Management structure for project aspects includes project lifecycle, roles and responsibilities, and communication.

## Project Lifecycle

During the course of this semester our team is implementing an Agile methodology to accomplish the project. The agile methodology allows for opportunity to assess the direction of work throughout the development lifecycle. We will follow regular cadences of work, known as sprints, in which our team will try to accomplish a successful product. We will introduce this methodology by following Scrum. This ensures that our team emphasizes empirical feedback, team self-management, and striving to build properly tested product increments within short sprints. Below is a list of what each sprint should include:

1. Requirement Specifications
2. Analysis
3. Design
4. Implementation
5. Testing

Following these procedures will ensure that our project is successful and no important features are left short of time to implement. There will be a total of 5 sprints as well as preparation for our final submission and presentation. Sprints will include the following actions: requirement specifications, analysis, design, implementation, and testing. Each sprint will reveal any pivots, changes, and will ensure that we have a better assessment of our progress and performance. An assessment report will be done after each sprint and the report will consist of: assessing the project progress, team performance, pivots, changes, and other important findings.

## Roles and Responsibilities

This table includes the roles and responsibilities following a SCRUM approach to implement software development tasks for this project.

| Role | Assigned Individual | Responsibility |
| --- | --- | --- |
| Product Owner | Castillo (Sprint 1 & 2) | * Responsible for Return on Investment (ROI) * Final arbiter of requirements questions * Focused more on the what than on the how |
| Harrison (Sprint 3 & 4) |
| Morrison (Sprint 5 & Final) |
| SCRUM Development Team  (Cross-functional group; Attempts to build a “potentially shippable product increment” every Sprint; Collaborates; Self-organizing) | Castillo, Harrison, & Morrison | * Write code * Unit Testing * Fix & test defects * Code inspection * Write test plans * Execute test plans with test cases * Develop/interpret user documentation * Participate in meetings about the project |
| ScrumMaster  (Has no management authority; Doesn’t have a project manager role; Facilitator) | Morrison (Sprint 1 & 2) | * Protects team from distractions and interruptions * Facilitates the process * Promotes improved engineering practices * Follows ScrumMaster checklist |
| Castillo (Sprint 3 & 4) |
| Harrison (Sprint 5 & Final) |

## Communication

For our project the dedicated ScrumMaster will be responsible for initiating team meetings on Wednesdays and Saturdays at a location that works for all members. ScrumMaster will ensure that our team have a location to meet without distractions and will run through a ScrumMaster’s checklist to ensure other members are performing their duties. Technology used for communication include: GroupMe (for instant messaging), Dropbox (for documents and images), and GitHub (for code sharing).

# Planning and Control

Planning and control project aspects related to planning, and controlling the work.

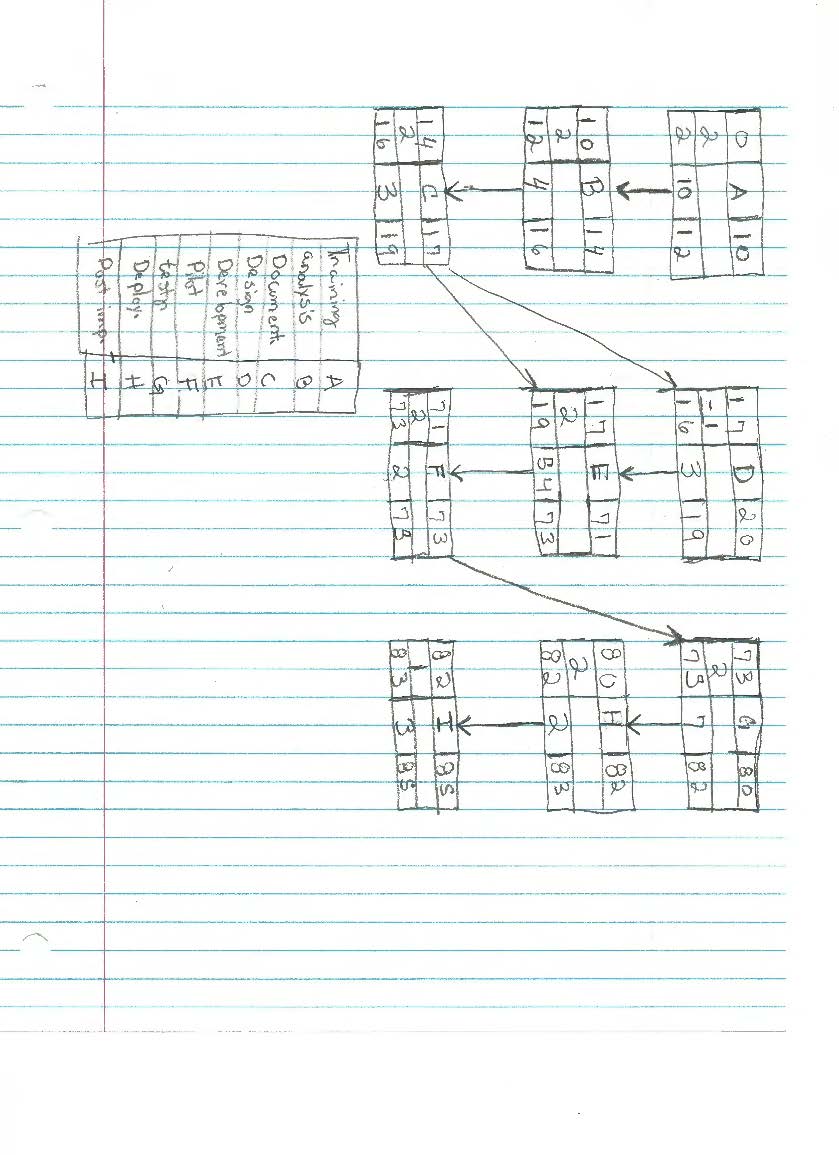
## Milestones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Due Date** | **Status** | **Responsible** | **Comments** |
| Project Start Up Meeting | Week 2  1/31/2015 | Completed | Castillo, Harrison, Morrison |  |
| Project Proposal | Week 3  2/1/2015 | Completed | Castillo, Harrison, Morrison |  |
| Sprint 1 Features | Week 4 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 1 Artifact & Assessment Report | Week 5 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 2 Features | Week 6 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 2 Artifact & Assessment Report | Week 7 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 3 Features | Week 8 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 3 Artifacts & Assessment Report | Week 9 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 4 Features | Week 10 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 4 Artifact & Assessment Report | Week 11 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 5 Features | Week 12 | Pending | Castillo, Harrison, Morrison |  |
| Sprint 5 Artifact & Assessment Reports | Week 13 | Pending | Castillo, Harrison, Morrison |  |
| Lessons Learned | Week 14 | Pending | Castillo, Harrison, Morrison |  |
| Develop Final Project Report | Week 15 | Pending | Castillo, Harrison, Morrison |  |
| Final Project Presentation | Week 16 | Pending | Castillo, Harrison, Morrison |  |

## Work Breakdown Structure



## Schedule



|  |  |  |  |
| --- | --- | --- | --- |
| Task Name | Duration | Start | Finish |
| **Software Development** | **81 days** | **Wed 1/14/1** | **Tue 5/5/15** |
| **Post Implementation Review** | **3 days** | **Fri 5/1/15** | **Tue 5/5/15** |
| Document lessons learned | 2 days | Fri 5/1/15 | Sun 5/3/15 |
| Distribute to team members | 1 day | Mon 5/4/15 | Mon 5/4/15 |
| Create software maintenance team | 2 days | Mon 5/4/15 | Tue 5/5/15 |
| Post implementation review complete | 0 days | Tue 5/5/15 | Tue 5/5/15 |
| **Deployment** | **2 days** | **Wed 4/29/1** | **Thu 4/30/15** |
| Determine final deployment strategy | 1 day | Wed 4/29/15 | Wed 4/29/15 |
| Develop deployment methodology | 1 day | Wed 4/29/15 | Wed 4/29/15 |
| Secure deployment resources | 1 day | Wed 4/29/15 | Wed 4/29/15 |
| Train support staff | 1 day | Wed 4/29/15 | Wed 4/29/15 |
| Deploy software | 1 day | Thu 4/30/15 | Thu 4/30/15 |
| Deployment complete | 0 days | Thu 4/30/15 | Thu 4/30/15 |
| **Documentation** | **3 days** | **Thu 1/29/15** | **Mon 2/2/15** |
|  | | | |
|  | | | |
| Task Name | Duration | Start | Finish |
| Develop Help specification | 2 days | Thu 1/29/15 | Fri 1/30/15 |
| Develop Help system | 2 days | Thu 1/29/15 | Fri 1/30/15 |
| Review Help documentation | 2 days | Thu 1/29/15 | Fri 1/30/15 |
| Incorporate Help documentation feedback | 2 days | Thu 1/29/15 | Fri 1/30/15 |
| Develop user manuals specifications | 2 days | Sat 1/31/15 | Sun 2/1/15 |
| Develop user manuals | 2 days | Sat 1/31/15 | Sun 2/1/15 |
| Review all user documentation | 1 day | Mon 2/2/15 | Mon 2/2/15 |
| Incorporate user documentation feedback | 1 day | Mon 2/2/15 | Mon 2/2/15 |
| Documentation complete | 0 days | Mon 2/2/15 | Mon 2/2/15 |
| **Training** | **10 days** | **Wed 1/14/1** | **Tue 1/27/15** |
| Develop training specifications for end users | 4 days | Wed 1/14/15 | Sat 1/17/15 |
| Develop training specifications for helpdesk support staff | 4 days | Wed 1/14/15 | Sat 1/17/15 |
| Identify training delivery methodology (computer based training, classroom, etc.) | 4 days | Wed 1/14/15 | Sat 1/17/15 |
| Task Name | Duration | Start | Finish |
| Develop training materials | 5 days | Sun 1/18/15 | Thu 1/22/15 |
| Conduct training usability study | 5 days | Sun 1/18/15 | Thu 1/22/15 |
| Finalize training materials | 4 days | Thu 1/22/15 | Tue 1/27/15 |
| Develop training delivery mechanism | 4 days | Thu 1/22/15 | Tue 1/27/15 |
| Training materials complete | 0 days | Tue 1/27/15 | Tue 1/27/15 |
| **Testing** | **7 days** | **Mon 4/20/1** | **Tue 4/28/15** |
| Develop unit test plans using product specifications | 6 days | Mon 4/20/15 | Sat 4/25/15 |
| Develop integration test plans using product specifications | 3 days | Sat 4/25/15 | Tue 4/28/15 |
| **Development** | **54 days** | **Wed 2/4/15** | **Fri 4/17/15** |
| Review functional specifications | 5 days | Wed 2/4/15 | Tue 2/10/15 |
| Identify modular/tiered design parameters | 5 days | Tue 2/10/15 | Mon 2/16/15 |
| Assign development staff | 4 days | Mon 2/16/15 | Thu 2/19/15 |
| Develop code | 32 days | Thu 2/19/15 | Thu 4/2/15 |
|  | | | |
|  | | | |
| Task Name | Duration | Start | Finish |
| Developer testing (primary debugging) | 10 days | Thu 4/2/15 | Wed 4/15/15 |
| Development complete | 0 days | Fri 4/17/15 | Fri 4/17/15 |
| **Design** | **3 days** | **Sun 2/1/15** | **Tue 2/3/15** |
| Review preliminary software specifications | 1 day | Sun 2/1/15 | Sun 2/1/15 |
| Develop functional specifications | 1 day | Sun 2/1/15 | Sun 2/1/15 |
| Develop prototype based on functional specifications | 1 day | Sun 2/1/15 | Sun 2/1/15 |
| Review functional specifications | 1 day | Mon 2/2/15 | Mon 2/2/15 |
| Incorporate feedback into functional specifications | 1 day | Mon 2/2/15 | Mon 2/2/15 |
| Obtain approval to proceed | 1 day | Tue 2/3/15 | Tue 2/3/15 |
| Design complete | 0 days | Tue 2/3/15 | Tue 2/3/15 |
| **Analysis/Software Requirements** | **4 days** | **Wed 1/28/1** | **Sun 2/1/15** |
| Conduct needs analysis | 1 day | Wed 1/28/15 | Wed 1/28/15 |
| Draft preliminary software specifications | 1 day | Wed 1/28/15 | Wed 1/28/15 |
|  | | | |
|  | | | |
| Task Name | Duration | Start | Finish |
| Develop preliminary budget | 1 day | Thu 1/29/15 | Thu 1/29/15 |
| Review software specifications/budget with team | 1 day | Thu 1/29/15 | Thu 1/29/15 |
| Incorporate feedback on software specifications | 1 day | Fri 1/30/15 | Fri 1/30/15 |
| Develop delivery timeline | 1 day | Fri 1/30/15 | Fri 1/30/15 |
| Obtain approvals to proceed (concept, timeline, budget) | 1 day | Sat 1/31/15 | Sat 1/31/15 |
| Secure required resources | 1 day | Sun 2/1/15 | Sun 2/1/15 |
| Analysis complete | 0 days | Sun 2/1/15 | Sun 2/1/15 |
| **Pilot** | **2 days** | **Sat 4/18/15** | **Mon 4/20/1** |
| Identify test group | 1 day | Sat 4/18/15 | Sat 4/18/15 |
| Develop software delivery mechanism | 1 day | Sat 4/18/15 | Sat 4/18/15 |
| Install/deploy software | 1 day | Sun 4/19/15 | Sun 4/19/15 |
| Obtain user feedback | 1 day | Sun 4/19/15 | Sun 4/19/15 |

# Costs and Resources

Estimates for costs and resources.

The average salary for a video game designer is around $54,000 a year. That comes out to $27 an hour for a normal 40 hour week. The three of us working 10 hours each a week for 13 weeks comes out to $10,530.

# Technologies to be used

Hardware and software needed for this project.

**Hardware:**

Personal Computer(s)

Storage Device(s)

**Software:**

Unity

Dropbox

Github

Notepad++

TextWrangler

Photo Editor

**Revision History:**

Version 1.0: Last Modified February 1, 2015 – Castillo, Harrison, & Morrison

Produced Document.