Group C

Project Plan

Snake Game for the Android Smartphone

Awais Nisar, James Gregory, Michal Huzevka, Welly Mulyadi
9/22/2013

Table of Contents

1. Overview

Value proposition

In order to demonstrate the effective use of the Waterfall software development model, we will create a simple Smartphone game. The game, a remake of Snake, will be developed in 4 weeks.

2. Assumptions

User based assumptions

1. The user is familiar with swipe gestures
2. The user is familiar with how snakes works

Project based assumptions

1. As a team we have working knowledge of java
2. We have enough time to complete our functional, strategic and technological goals.
3. We will have weekly meetings to discuss our milestones

3. Success Criteria

Project management criteria

- Meeting weekly milestones
- Working collaboratively in a team
- Splitting tasks accordingly
- Assigning roles
- Getting user feedback

Snake game criteria

- Game is well tested
- Functioning Menu
- Smooth Gameplay
- Responsive to swipe gestures

4.Goals and Scope

4.1 Project Goals

Functional goals

Our ultimate goal is to create a remake of the snake game that will allow users to use swipe gestures to control the snakes movement.

Strategic goals

Over the next couple of weeks as a team we wish to achieve our set out milestones and complete the game before the due date so we can perform some usability testing.

Technological goals

Our goal is to create the application for android based devices this includes android tablets and android smartphones.

4.2 Constraints

Time: As we only have a couple of weeks to complete this, time is our main constraint. We have a time constraint of 200 man hours for this project, for all phases of the project. Each developer should spend up to 50 hours on this project.

Technological: Platform must run Android OS and have a touch screen.

Environmental: Due to time constraints , we are limiting ourselves to working with android environment.

4.3 Project Scope

4.3.1 Included

- A remake of snake that will run on android smartphones with touch screens
- App features:
    - Menu on launch (Play, Tutorial, Options, Highs cores)
    - Tutorial submenu
    - Options submenu (Toggle Sound, Clear High-scores)
    - High scores submenu
- Game play Features
    - Movement Controlled by swipe gestures
    - Score tracking (in game and high scores)
    - Random power-up/down drops (Snake's Size Enhancement/Reduction)
    - Increased difficulty over time
    - Random obstacles
    - Faster snake animation

4.3.2 Excluded

- Portability to iOS or Windows Phone.
- Portability to older phone without touch screen.
- Use of keys to control snake.
- Touch screen 'virtual keys' to control snake.
- Sharing of high-scores

5.Detailed Project Requirements

5.1 Project requirements

The project requires us to build an android Smartphone game (snakes) using the waterfall structure. The project requires us to work collaboratively as a team to set milestones and aim to achieve them. Since we are using a waterfall structure the project requires us to plan/break down tasks using a Gantt Chart (included in appendix A) which is packaged with project Libre. In addition to tasks breakdown, our project requires us to take on multiple roles due to the fact that our team consists of 4 members and the numerous tasks to be completed.

5.2 Product requirements

The core requirements of the application are an android operating system and a compatible touch phone. The product also requires a capacitive touch screen to operate properly as it will require the user to swipe up and down to move the snake.

5.3 User requirement

The user requires us to build a simple snake game for the android touch phone which is easy to learn and play. The user also requires us to provide a simple tutorial if they are confused about some aspect of the game.

5.4 Management requirements

Management requires us to set milestones and achieve them.

5.5 Functional Requirements

Our project requires functions to be implemented and these include calculating scores, determining the location of power-up/ down fruits, determining difficulty levels and resizing the snake.

5.6 Maintenance Requirements

Our application will require constant maintenance after the initial release so its important for the team to release patches and bug fixes to prevent the application from being not maintained well.

6. Project Deliverables

6.1 Internal receivers (Project team members)

1. Project Plan
2. Functionality Prototype
3. Game play Prototype
4. Final Prototype
5. Final Product

6.2 External receivers (Management and users)

1. Management Presentation
2. First Product Presentation
3. Second Product Presentation
4. Product Demonstration

7. Resource Requirements

7.1 Project resource requirements

Hardware:

Android Smartphones/tablets

Software:

Java, XML, web tutorials, github.

7.2 Application Requirements

Hardware

Smartphone/tablets phone with a touch screen

Software

Android 4.0+ OS

8. Implementation, Plan, Schedule

Milestones	Description	Criteria	Planned Due Date	Hours (~)
M0	**Start Project**		**Week 6**	**8**
	Project goals and scope define		5/09/2013	
M1	**Start Planning**		**Week 7**	**8**
	Work on project plan	M0	14/09/2013	
M2	**Finalise Project Plan**		**Week 8**	**8**
	Finalise project plan for assessment, work on presentation	M0, M1	19/09/2013	
M3	**Start Execution**		**Week 9**	**16**
	Paper prototypes for menu, tutorial, high-scores and gameplay. Set up work environment (java, eclipse, github ..)		25/09/2013	
M4	**Gesture Prototype**		**Week 9**	**8**
	Work on simple program that registers swipe gestures (gesture prototype)	M3	29/09/2013	
M5	**Menu Prototype**		**Week 9**	**8**
	Develop simple phone app with a menu at launch and desired submenus	M3	29/09/2013	
M6	**Gameplay Framework**		**Break**	**10**
	Develop framework for gameplay, gesture controls for snake like object on screen	M3, M4	06/10/2013	
M7	**Functioning Menu**		**Break**	**14**
	Implement all menu features by end of mid semester break.	M3	6/10/2013	
M8	**Functioning Gameplay**		**Week 10**	**22**
	Work on implementing all gameplay features by the end of week 11	M6	13/10/2013	
M9	**App Prototype**		**Week 10**	**18**
	Combine Functioning Menu and Gameplay Framework into one complete app prototype	M9, M6	13/10/2013	

M10	**Polished App**		**Week 11**	**20**
	Modify the game app based on feedbacks from peers and run Tests	M10	20/10/2013	
M11	**Completed App**		**Week 12**	**20**
	Ensure App is ready for submission. Hours in week 12 are unallocated as a buffer for any unexpected problems.	M11	27/10/2013	

9. Task breakdown

Team 1 - Functionality of App

Team member	Role	Responsibilities
Awais Nisar	Project manager /developer	Overseeing the whole project. Developing the functionality of the application.
Welly Mulyadi	Developer/designer	Developing the functionality of the application. Creating paper based prototypes. Creating UML diagram

Team 2 -Gameplay Mechanics

Team member	Role	Responsibilities
James Gregory	Developer	Overseeing the work. Assisting in developing the gameplay mechanics
Michal Huzevka	Developer	Developing the gameplay of the application. Analyzing the risks involved in making critical decisions.

Work allocation

Team member	Work allocated	Related Milestones
Awais Nisar	Functionality of the Game	M0-3, M5, M7, M9, M10, M11
Welly Mulyadi	Functionality of the Game	M0-3, M5, M7, M9, M10, M11

James Gregory	Gameplay mechanics	M0-3, M4, M6, M8, M10, M11
Michal Huzevka	Gameplay mechanics	M0-3, M4, M6, M8, M10, M11

10. Testing Plan

Type of testing	**Testing period**
Unit testing - This will test the functionality of the application	Throughout development
User acceptance testing - Get feedback from users (Usability, Accessibility)	Week 10
Alpha testing (Functional testing) -	Week 11
Beta testing (Non - functional testing) -	Week 12

11. Risk Management

Our project might run into unexpected problems. An event might occur that can have a negative impact on the status of the project. These events are deemed risk events and the risk manager is responsible for determining whether a risk event has occurred and for informing the rest of the group. The risk manager will report any risk events during the weekly meeting. A solution will then be proposed during the weekly meeting that should conform to the solution in the risk management plan.

The risk management plan shows possible risk events and solutions for them.

Risk Management Plan

Risk Event	Severity	Probability	Possible solution
Insufficient time to complete project	Medium	Low	We will need to make sure that we're meeting the milestones on time so that we don't get too far behind. If we do fall behind, then we may have to work past our time constraints
Requirements for a milestone not met	Medium	Medium	We will need to do all the proposed features for the unsuccessful milestone in the next milestone. We will also try to understand why we missed the milestone's requirements.
Source code unintentionally deleted	Low	Low	We will store the source code on Github and commit frequently so that means we won't lose too much code.
Failure to meet product requirements	High	Low	All product requirements must be met for the product to be a success. During Milestone 10 and Milestone 11 we will formally go through all the requirements and scope and make sure our product meets those requirements.
Client doesn't accept the	High	Low	We will need to get new acceptance criteria from the client

product even though the product meets our requirements			and rewrite the product requirements and our goals. See 13. Change Management
Product deviates from original requirements	Medium	Low	We will need to either change the product to match the product requirements, or alter the product requirements. If we alter the project requirements, we must submit a change request. See 13. Change Management
Ambiguous requirements	Medium	Low	Some requirements may be ambiguous and hard to measure (for example – the smooth gameplay requirement). To solve any ambiguous requirements, we will write more precise criteria for those requirements and confirm it with upper management if needed.

12. Version Management

We will use the Git version control system and all our source code will be versioned on Github. To prevent merge conflicts, we will commit to the repository frequently. Git is a distributed version control system which means we can store a local copy of the entire repository on our own computers. We will store this local copy as a backup in case the Github website goes down.

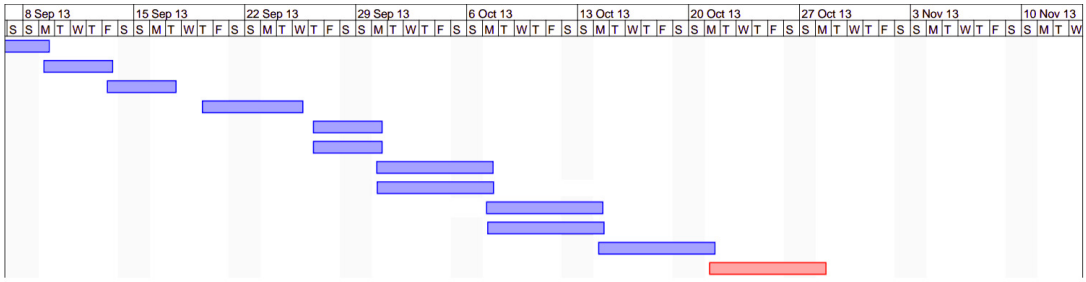
By having version management we can prevent the loss of critical milestones and also have a fallback for when the new version is not stable.

13. Change Management

In the case of change of plan for example re-defining project scope the team will need to hold a meeting to discuss the best solution to reach the desired state. A change request will also have to be written and then submitted to upper management (Brad/Wayne). If the change request is approved then we may change the project requirements and/or project scope. A change notification will describe all the changes in detail and will be given to upper management after the change request is approved.

14. Appendices

Appendix A:



		Name	Duration	Start	Finish
1		Start Project	2 days	6/09/13 8:00 AM	9/09/13 5:00 PM
2		Start Planning	5 days	9/09/13 8:00 AM	13/09/13 5:00 PM
3		Finalise Project Plan	3 days	13/09/13 8:00 AM	17/09/13 5:00 PM
4		Start Execution	5 days	19/09/13 8:00 AM	25/09/13 5:00 PM
5		Gesture Prototype	3 days	26/09/13 8:00 AM	30/09/13 5:00 PM
6		Menu Prototype	3 days	26/09/13 8:00 AM	30/09/13 5:00 PM
7		Gameplay Framework	6 days	30/09/13 8:00 AM	7/10/13 5:00 PM
8		Functioning Menu	6 days	30/09/13 8:00 AM	7/10/13 5:00 PM
9		Functioning Gameplay	6 days	7/10/13 7:00 AM	14/10/13 5:00 PM
10		App Prototype	6 days	7/10/13 7:00 AM	14/10/13 5:00 PM
11		Polished App	6 days	14/10/13 7:00 AM	21/10/13 5:00 PM
12		Completed App	6 days	21/10/13 8:00 AM	28/10/13 5:00 PM