Project Plan for Team Broken Lightbulb

Version: 1.0
Author: Connor Schenck
Date: 8 Sep 2010

TASKS, RESOURCES, SCHEDULE, AND DELIVERABLES

Task	Resources	Deliverable	Date	Risks & Mitigations
Team name and roles	Connor	Team Composition Form	9/3/2010	
Develop Project Plan	Connor (with aid from entire team	Project Plan	9/10/2010	Risk1, Risk2
Develop Economic Model	Connor	Economic Model	9/10/2010	Risk3
Create Use Cases	Titus, Ryan T	Use Cases	9/10/2010	Risk1
Create Screenshots	Ryan T	Preliminary Screenshots	9/10/2010	
Defined Project Measures	Connor	Project Measures	9/17/2010	
Develop commonality/variability analysis	Titus, Ryan T	Commonality/Variability analysis	9/17/2010	Risk1
Develop system generation and verification plan	Taylor, Ezra, Ryan F, John	System Generation and Verification Plan	9/24/2010	
Create decision model table	Titus, Ryan T	Decision Model Table	9/24/2010	
Phase 1				
Develop Modules	Ezra, Taylor	Module	9/27/2010	Risk5
Develop Uses Structures	Ezra, Taylor	Uses Structures	9/27/2010	Risk5
Develop Process Structure	Ezra, Taylor	Process Structure	10/1/2010	Risk5
Develop Interface Specifications	Ezra, Taylor	Interface Specifications	10/1/2010	Risk5
Code the modules	Titus, Ryan T, Ryan F, Connor, John	Module Implementation	10/5/2010	Risk5
Create the module tests	Ezra, Taylor, Ryan F, John	Module Tests	10/5/2010	
Test the modules	Ezra, Taylor, Ryan F, John	Test Results Report	10/11/2010	
Phase 1 Complete	**entire team**		**10/11/2010**	
Develop Mid Project Plan	Connor	Mid Project Plan	10/15/2010	
Develop Midterm Progress Report	Connor, Ezra, Titus, Taylor, Ryan T, Ryan F, John	Midterm Progress Report	10/19/2010	
Phase 2				
Revise Modules	Ezra, Taylor	Revised Module	10/22/2010	Risk4, Risk5
Revise Uses Structures	Ezra, Taylor	Revised Uses Structures	10/22/2010	Risk4, Risk5
Revise Process Structure	Ezra, Taylor	Revised Process Structure	10/26/2010	Risk4, Risk5
Revise Interface Specifications	Ezra, Taylor	Revised Interface Specifications	10/26/2010	Risk4, Risk5
Code the modules	Titus, Ryan T, Ryan F, Connor, John	Module Implementation	11/1/2010	Risk4, Risk5
Create the module tests	Ezra, Taylor, Ryan F, John	Module Tests	11/1/2010	
Test the modules	Ezra, Taylor, Ryan F, John	Test Results Report	11/5/2010	
Phase 2 Complete	**entire team**		**11/5/2010**	
Phase 3				
Revise modules, uses structures, process structure, and interface specifications to add a new feature	Ezra, Taylor	modules, uses structures, process structure, interface specifications	unknown	Risk4
Implement the code to add the new feature	Titus, Ryan T, Ryan F, Connor, John	Module Implementation	unknown	Risk4
Develop tests for the new feature code	Ezra, Taylor, Ryan F, John	Module Tests	unknown	
Test the new feature	Ezra, Taylor, Ryan F, John	Test Results Report	unknown	
Phase 3 Complete	**entire team**		**12/10/2010**	
Develop Retrospective Report	Connor	Retrospective Report	12/10/2010	

RISKS

Risk ID	Risk	Mitigation
Risk1	The project is too ambitious	We've planned for 3 phases of development. The third phase is designed to be cut short if needed (or not done at all)
Risk2	Project Schedule is too aggressive	All of our due dates for artifacts during the phases are well before the dates set by the professor leaving us a buffer if needed
Risk3	Estimates are inaccurate	As this is a prototype project, accurate estimates are not required
Risk4	Each phase has its own architecture design	Designs and code from previous phases will be reused to save time and keep from reinventing the wheel
Risk5	Unexpected complications arrise	The due dates within the phases are designed to be able to be pushed back if needed, sacrificing development time in phase 3

REFERENCES

1. Our schedule is planned out in Google Calendar format: TBD

NOTES

Our project plan is laid out in three phases:

- **Phase 1**: This is the initial design phase. During this phase only the most basic and necessary features will be implemented. This phase will implement the commonalities of the family. The result will essentially be only as much as is required to make a runnable game.
- **Phase 2**: This is the product-line phase. During this phase, the variabilities for each family member will be added and the necessary code to make it possible to generate new family members will be added. The result will be a family of games based off the game created in phase 1 with each having its own variabilities.
- **Phase 3**: This is the feature phase. This phase will consist of choosing a feature, designing it into the game, coding it in, and testing it. This will be repeated until there is no time remaining until the project deadline. This phase is designed to allow flexibility in scheduling deadlines and room to push them back. The result of this phase will be a more polished version of the family of games created in phase 2.

Economic Model

Market:

The target market is the PC gaming market. The market is estimated to be worth $11 billion. In recent years, the demand for graphics intensive PC games has declined while the popularity of simple yet entertaining games has increased (a good example of this are Flash based games). Our game will appeal to this demographic because it's simple graphics engine will not act as a barrier to people with weaker systems; our game mechanics are simple and easy to understand but still fun; and most importantly because we are using Java, our game will be system independent which will allow Apple and Linux users to play right alongside Windows users.

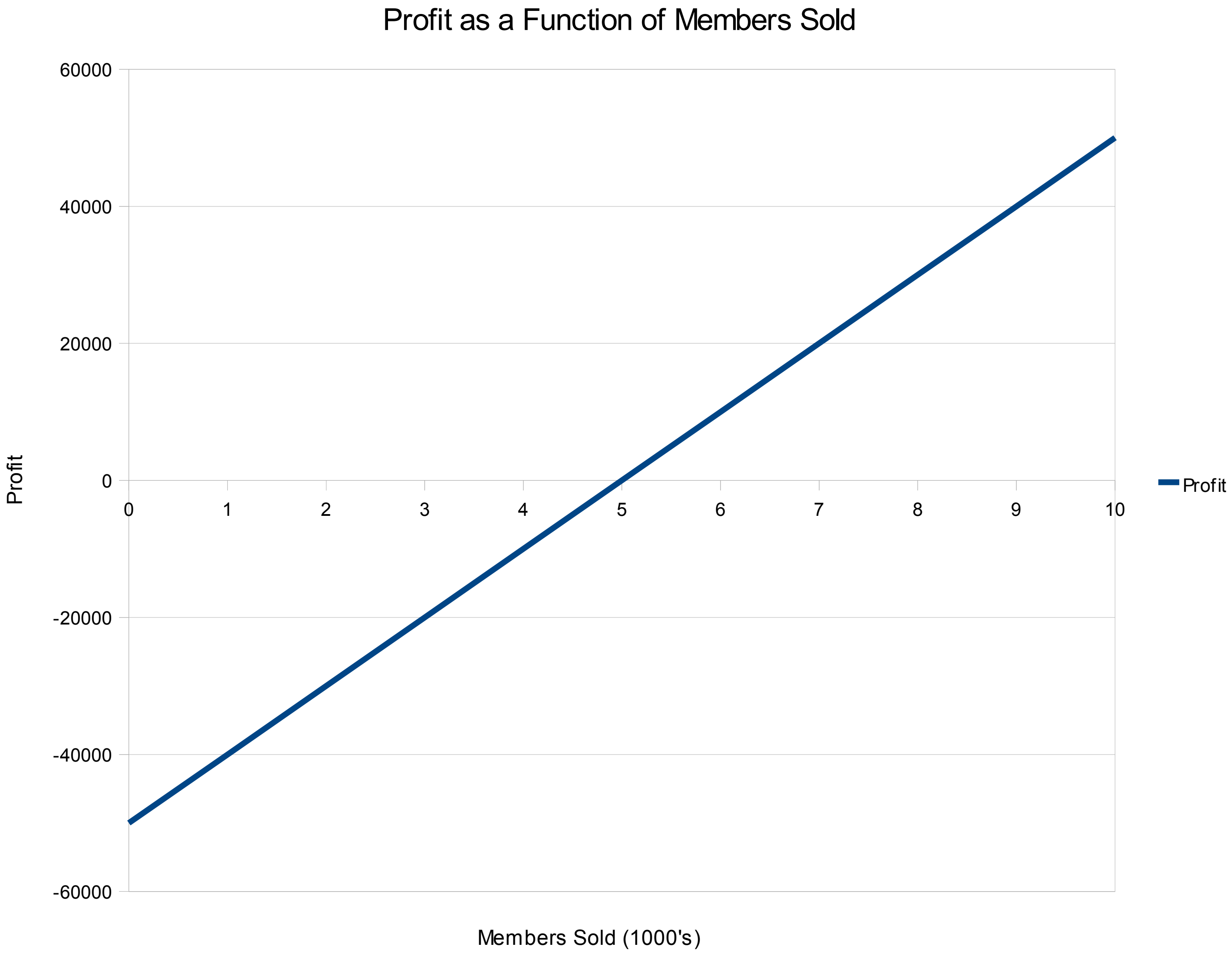
Our game will most strongly target strategy game players disillusioned with modern strategy games. Strategy games released in recent years have focused more on tactics and less on strategy leaving many gamers yearning for more. Our game is pure strategy based (no tactics), which will appeal to this market strongly. We plan to start out small by only targeting 5% of the total PC gamer market, but as the game gains reputation, we hope to increase our target. In ten years we plan to be targeting 15% of the market.

Number of family members:

Our game will have the ability to incorporate a vast number of units, tech trees, and maps. This will only depend on which version of our software a customer purchases. Any combination of units, tech trees, and maps is a potential family member.

The marginal cost of adding a new family member will be very small, as the game will be designed so that creating an additional family member only requires editing some parameters on a generator. Some of these parameters may be:

- Creating a race (a race is a collection of units and a tech tree)
- Adding or removing a race from the game
- Creating units
- Adding or removing units from a race
- Altering units
- Creating tech trees
- Adding or removing a tech tree for a race
- Altering tech trees
- Creating maps
- Altering maps



We will reach the break even point after selling 5000 members.

R = N * (P – C) – I

R: Profit

N: number of members sold

P: price to purchase one member ($10)

C: cost of selling one additional member ($0.01)

I: initial investment ($50,000)