The Ballad of Gumption

Defining & Understanding

COVER SHEET

Table of Contents

Identification of the Problem

The game is called "The Ballad of Gumption", and is a point and click adventure game based on the Gumption Tomes from chapters 1 through 10. The gameplay is reminiscent of the original Monkey Island series, with a modern, intuitive inventory system. The project fills the market gap for entertainment, where the market has a lack of 2D, point and click adventure games done in a retro, 8bit pixel-art style.

The requirements and features for this project include the following:

- Graphical user interface
- Python 3.0
- Self-contained .app file
- Multiple chapters
- Pointing
- Clicking
- Save files
- Hint systems
- Two chapters
- Tutorial

- Targets system requirements:
  - Memory: 1GB
  - Processor: 2.6 GHz
  - Graphics: Intel Iris 1536MB
  - Storage: 200MB GB

Also would like to add some stretch goals to my project:

- The addition of chapters 3 to 10 and maybe think about chapters 1 to 3 from volume two.
- Voice acting
- Bonus TBA content

The game mechanics consist of solving puzzles by combining actions or objects with other objects or entities. The successful progression of combinations serve to advance the games plot and bring on new sets of puzzles until the completion of all the content that the game has to offer. At this point, I am considering using pixel art for characters and objects, but leave the background in the original art style, although this is subject to change.

At this point, the way in which I will be able to implement this is by using one image as a background and another as a definition of movement boundaries. Clicking on any part of the background image will give the program coordinates of the clicked location, which will then move the player character sprite to that location. Apart from this, there will also be clickable locations on the screen, which will be triggered by onclick events.

The game is ultimately event driven, and will not do anything that isn't a reaction to user input. I decided to work on this project as many point and click games have complicated and unintuitive inventory and action systems. This game will simplify gameplay by inferring all actions. An example will be clicking on a character will assume the "talk to..." action, or clicking on certain objects will assume "pick up..." or "examine..." on other objects.

Quality Assurance

To make sure the product is of an acceptable standard, I must set a series of protocols that it has to adhere to:

- Usability: It is imperative that the user can understand how to use the system correctly. User interface plays an important role in this, as its design can make or break a software solution. To prevent any complications due to usability, I will need to constantly keep in contact with the user-base to gauge how well the user interface works as well as creating a consistent UI design. I can also add a tutorial and how-to to the software to teach the user basic use of the software and this would show the user the functionality of the gameplay. To make sure the user is sufficiently knowledgeable about the menu, I am considering adding tooltips that explain the use of each component of the menu, combined with a question mark that give a brief explanation of the menu. Finally, in order to test whether the solution properly addresses to criteria of usability, I will need to observe new users adapting to the software and report my findings.

- Maintainability: Another control I can use to test quality is maintainability, where the ease at which changes can be made are tested. The way in which I will ensure that my solution is maintainable is through abstraction. By isolating single jobs to single functions, I can easily interchange them with new code without disrupting the rest of the system. Another way to ensure that the code is maintainable will be to add internal documentation, where I will both fully comment my code, and use meaningful variable names (intrinsic commenting). Internal documentation will hopefully make the task of editing code far easier and less painful. To test maintainability, I will have to judge how maintainable the code is through my own experiences in changing the code.

- Efficiency: Efficiency basically is how well the software solution will run on any given hardware environment. At this stage, I aim to have it running on at least my own personal laptop, and therefore set my target system specifications as such:
  - Memory: 1GB
  - Processor: 2.6 GHz
  - Graphics: Intel Iris 1536MB
  - Storage: 200 MB
  - Operating System: OSX 10.10.1

  Although I have 100GB of storage to play with, I would hope to keep it below 200MB total. To ensure that I make the project as efficient as it possibly can, I will compress all my resources, such as images and sounds to the smallest size possible (within reason) this will hopefully prevent its size from exceeding 200MB. I can also make code efficient by reusing parts of it as modular functions. The smaller size of files as well as efficient code, will result in less pressure on the memory and processor. The test for this protocol will be whether it runs on my personal laptop, but I can also track its memory and processor usage through the system tools. Assuming that it runs, I will consider the solution to be sufficiently efficient.

- Accuracy: Accuracy is the measure of how correctly a solution can process and output information. This can be tested any number of ways throughout my project, by desk checking my functions output, or by assessing the validity of the data being output by the program. by using abstraction, I can test each function by passing in data, and receiving other data and then I can check the data to ensure its accuracy. I can also test the UI as the location that I click on the screen can be assessed by how close the sprite moves to that location, or alternatively, I can check that the dialogue being displayed by characters is the correct response.

User Feedback

The following outlines the requests and expected features suggested by surveyed individuals. Asterixed suggestions are not considered viable and therefore may be withheld from the project.

His Royal Excellency Jalfor
- Voice over *1
- Loading screen
- Changing cursor over clickable objects

Dandolo
- A character named 'Daniel'
- Online multiplayer *2
- Import music

Kapoon
- No branching option
- Gritty realism (pixel art)
- Character named 'Andrew' *3

PALZMADOLE
- Save files
- Hint sytem

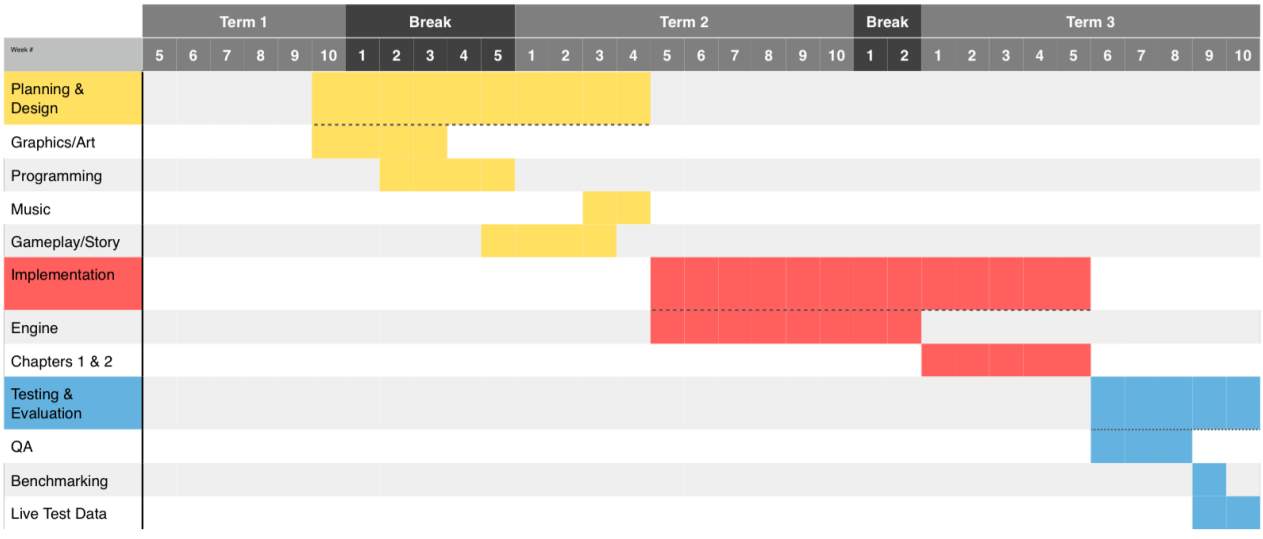
BB3B123
- Gmod *4
- Large menu icons
- Animated characters

MyStore
- Twitch streaming *5
- Bosses *6

*1) I do not have the funds available to pay for voice-acting for each character.
*2) At this point, there will be no online connectivity, and multiplayer would not make sense with the mechanics of the game.
*3) As the game is based on the 'Gumption Fables' series, I would not be inclined to meddle with the lore for the original authors sake.
*4) I don't believe I will be able to obtain Garry Neumann's IP, 'Garry's Mod' from Facepunch Studios.
*5) Similar to the online multiplayer, the project is not looking to have network connectivity.
*6) The nature of a point & click game does not involve 'boss' mechanics, and while the puzzles will get harder as the game progresses, there will be no ultimate character that you must fight - although, this is subject to change.

Gantt Chart

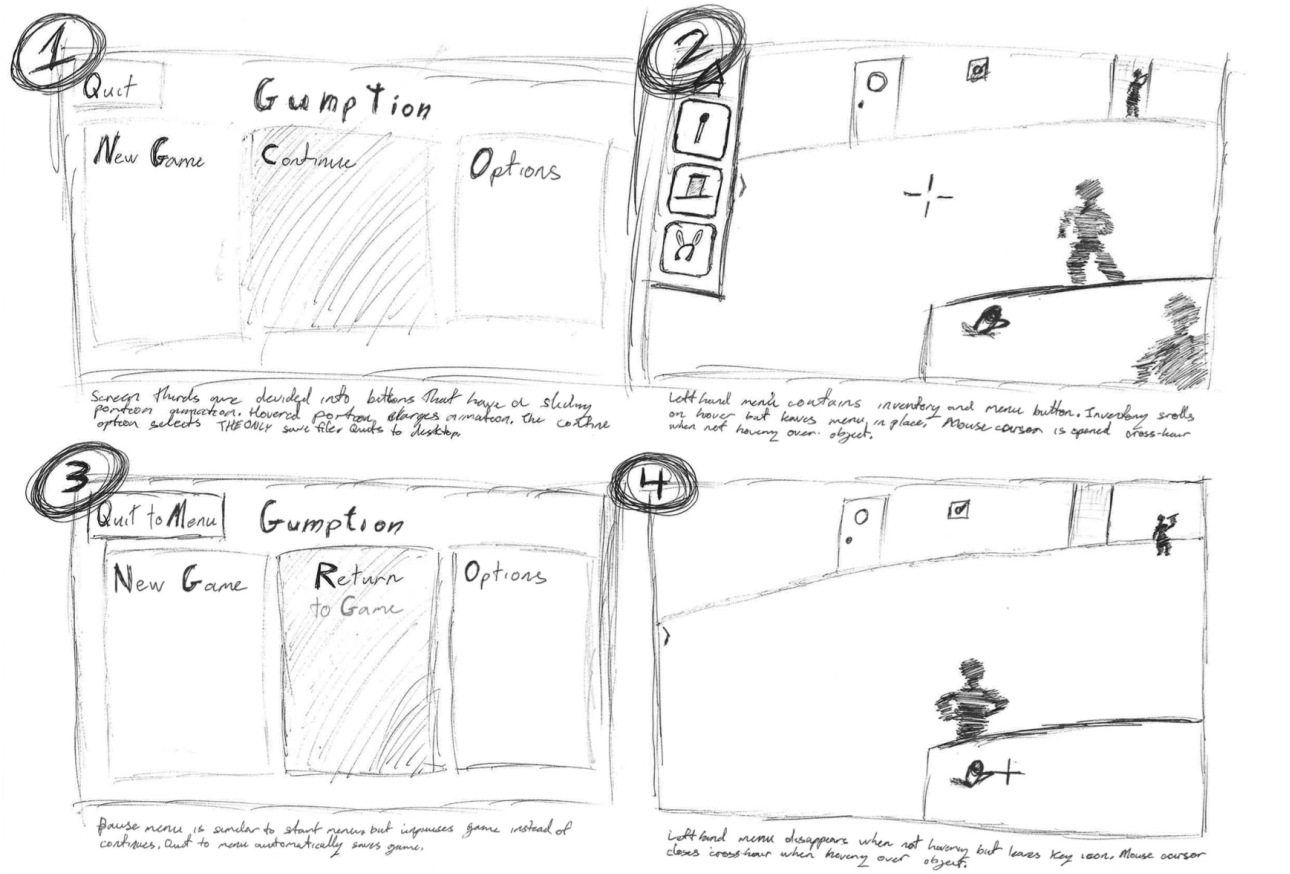

Term 1
Break
Term 2
Break
Term 3
Week #
5 6 7 8 9 10 1 2 3 4 5 1 2 3 4 5 6 7 8 9 10 1 2 1 2 3 4 5 6 7 8 9 10
Defining & Understanding
Identify Problem
Dev Approach
QA
Social & Ethical
Feedback
Gantt
Reflection
Website
IPO
Context
DFD
Storyboards
Structure Charts
Flowcharts
Data Dictionary
Test Data
Logbook

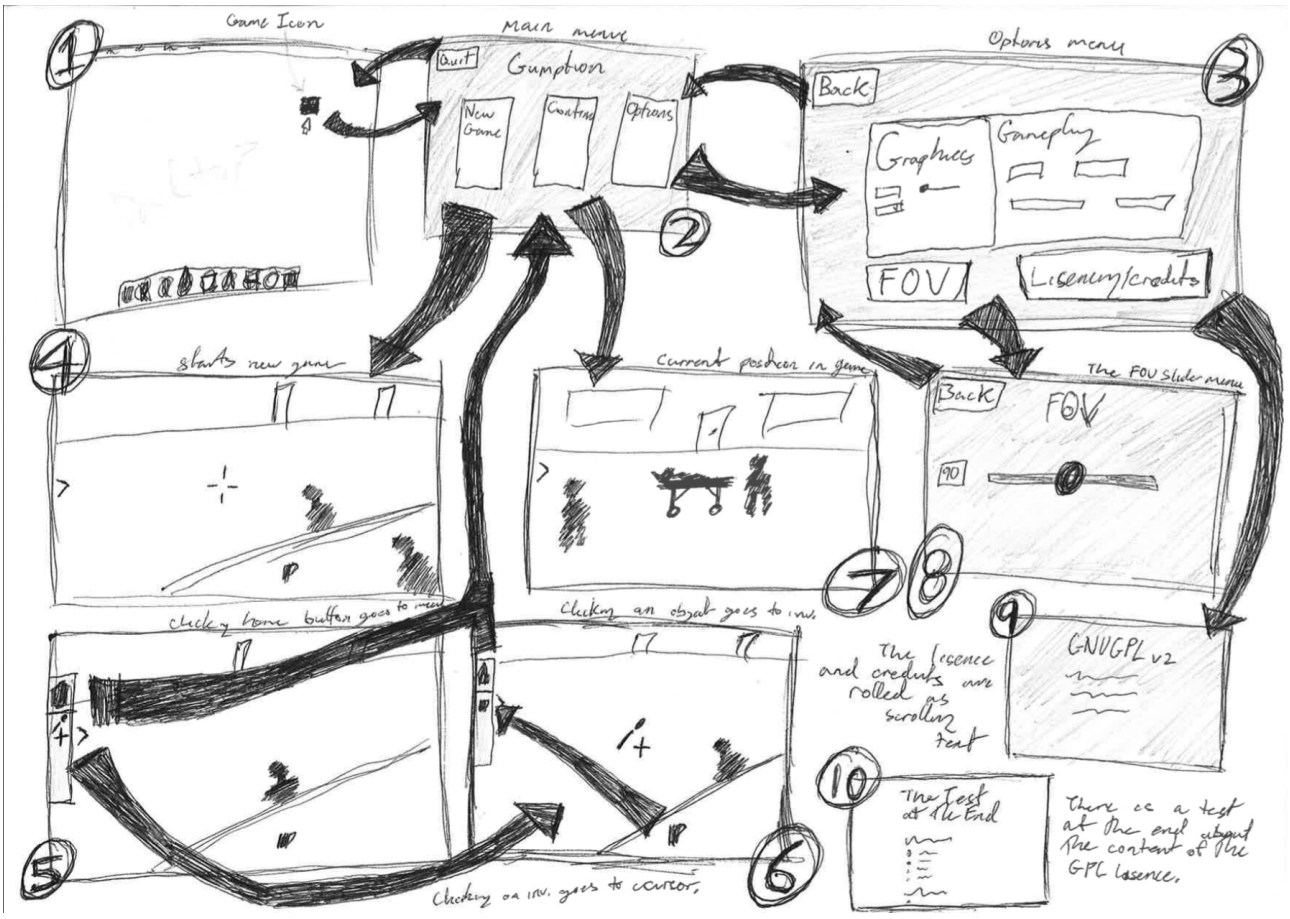

Term 1
Break
Term 2
Break
Term 3
Week #
5 6 7 8 9 10 1 2 3 4 5 1 2 3 4 5 6 7 8 9 10 1 2 1 2 3 4 5 6 7 8 9 10
Planning & Design
Graphics/Art
Programming
Music
Gameplay/Story
Implementation
Engine
Chapters 1 & 2
Testing & Evaluation
QA
Benchmarking
Live Test Data

Data Dictionary

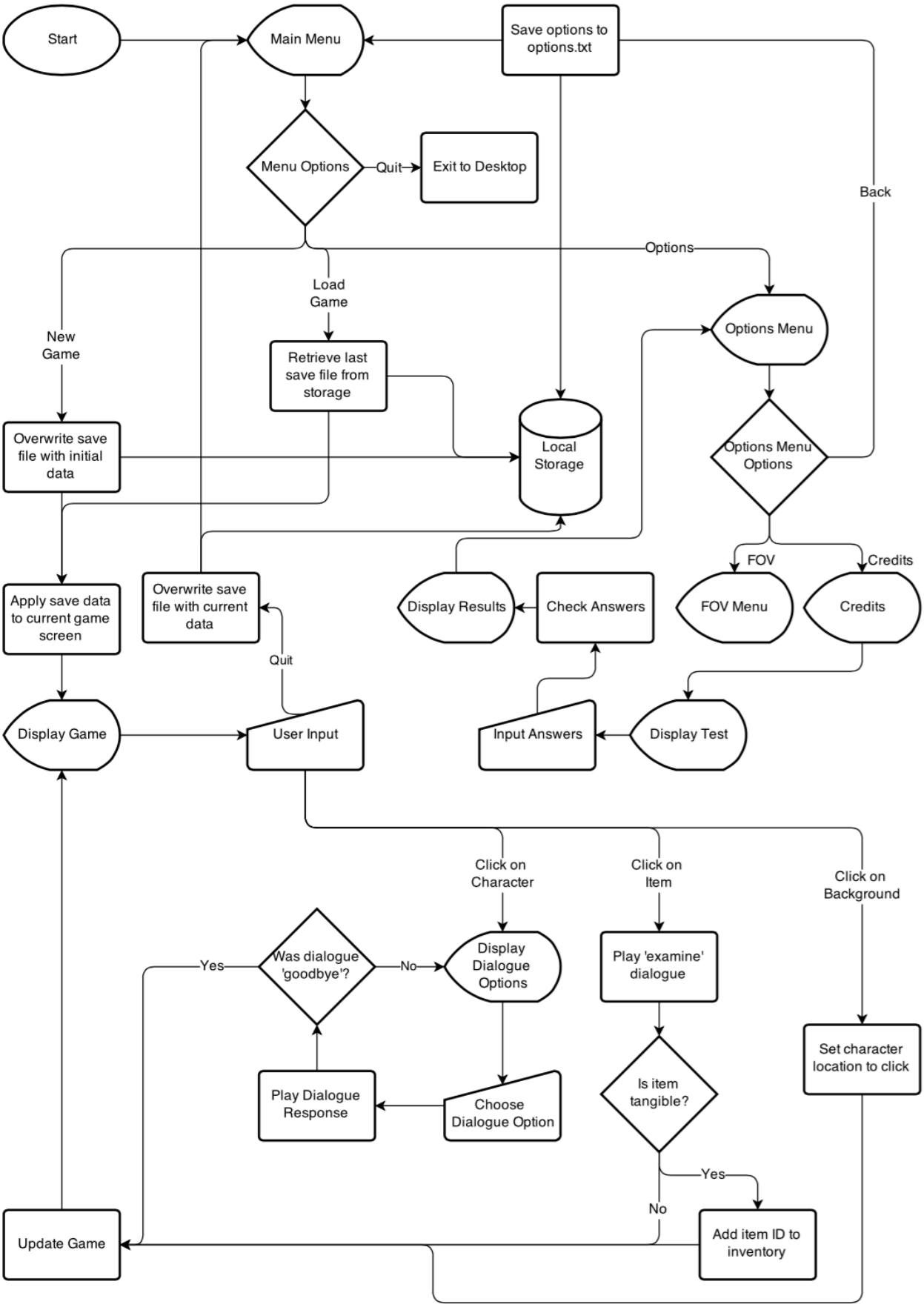
Data Item	Type	Size	Format	Example	Description
x-val	Floating Point	~ 4 bytes	x.yz	4.20	A value that indicates the x coordinate of where the user clicks.
y-val	Floating Point	~ 4 bytes	x.yz	4.20	A value that indicates the y coordinate of where the user clicks.
link	String	~ 19 bytes	res/<name>	res/dousticonb.png	A piece of data that indicates the location of sprites and other files.
id	String	~ 5 bytes	id_<item>	id_mug	An identifier that allows the program to reference objects and characters.
inv	Array	~ 25 bytes	[x, y]	[id_chair, id_spider]	An array that contains the elements that are in the inventory of the player.
event	Boolean	~ 2 bits	event_x	event_17	A piece of data that allows the game to tell wether certain events have been triggered.
dialogue	String	1 - ∞ bytes	dialogue_<character>_<number>	dialogue_matt_17	A value that references the witty banter and descriptions that will occur.
tan	Boolean	~ 2 bits	tan_x	tan_mug	A boolean value that indicates whether an object is tangible.
itemslink	Dictionary	100+ bytes	{x:y, a:b}	{id_mug: res/mug.png}	A dictionary that stores gives items identifier an image to reference.
itemsdesc	Dictionary	100+ bytes	{x:y, a:b}	{id_mug: dialogue_mug_01}	A dictionary that stores each items identifier and dialogue description.
itemstan	Dictionary	100+ bytes	{x:y, a:b}	{id_mug: tan_mug}	A dictionary that stores each items identifier and whether they are tangible.

Storyboards

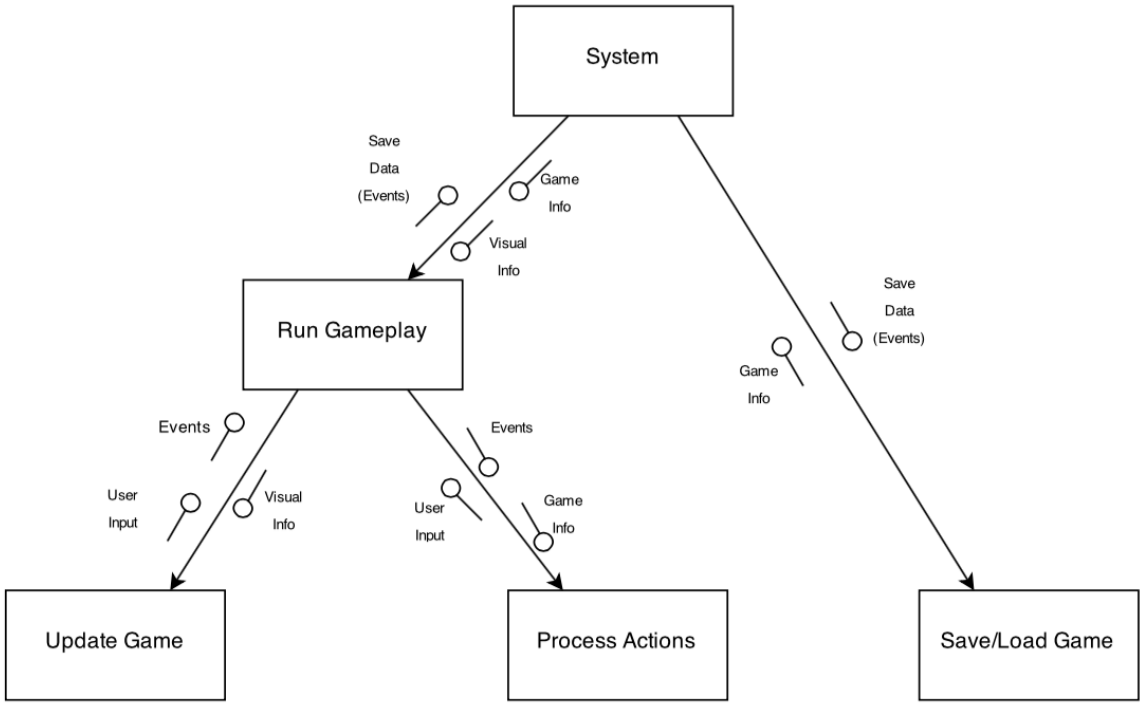




System Flowchart



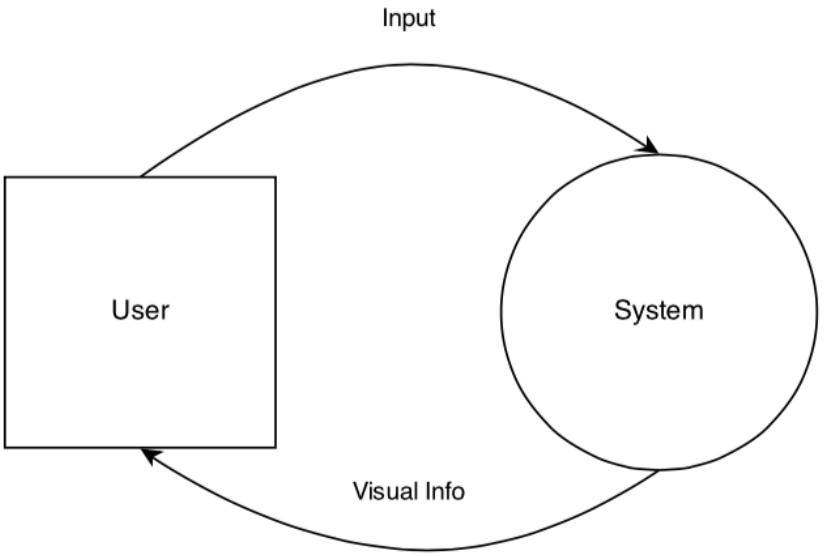
Structure Chart



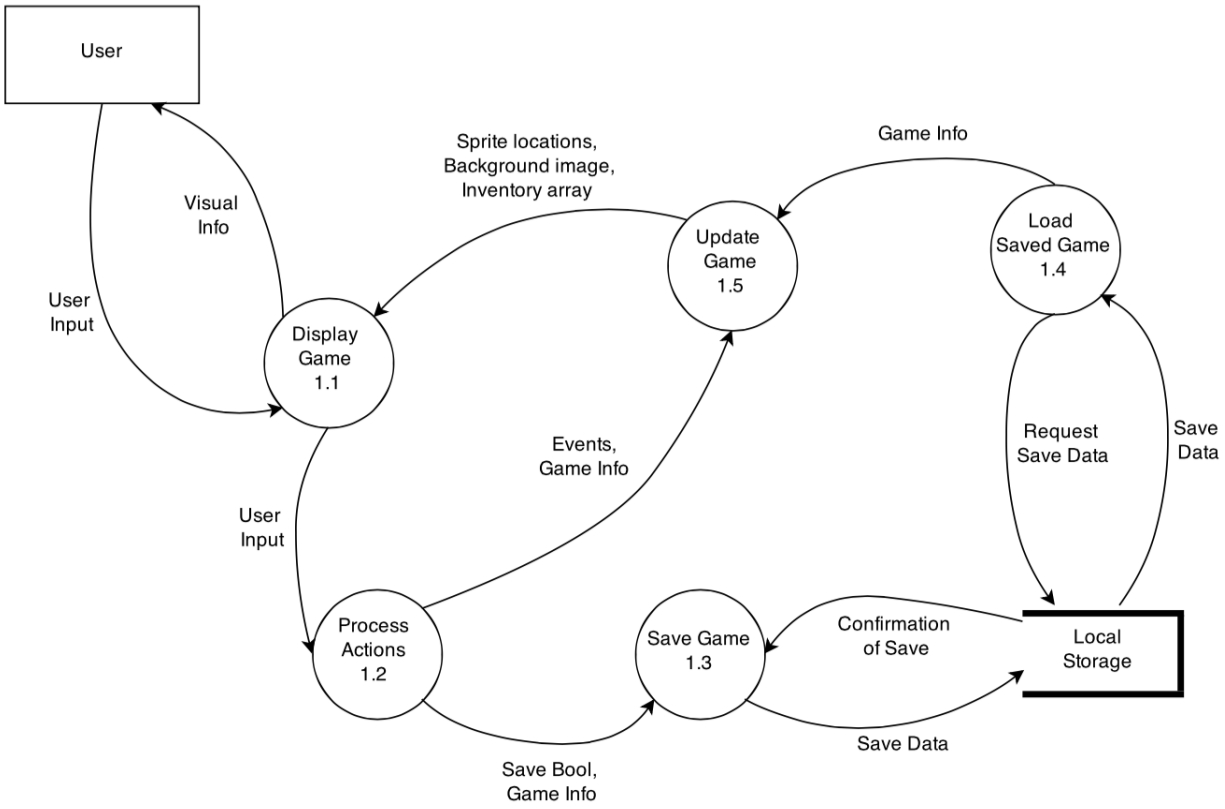
IPO Chart

INPUT	PROCESS	OUTPUT
Start application	Initialise game graphics and menu data	Game window —> loading screen —> main menu
User input	Process user input	Updated game
Press start game	Creates save file, appends data to save file, starts game	Shows in-game screen from first chapter
Exit to menu	Initiates save game and then replaces game screen with menu screen	Displays game is saved, shows main menu
Click on game background	Gets coordinates of click, moves sprite to location	Character moves to coordinates
Click on inventory item	Sets item ID to currently selected item	Item appears at cursor
Pick up item	Adds item to inventory array, removes item from canvas	Item disappears from canvas, item appears in inventory
Update game	Processes location of sprites and backgrounds on game canvas	Updated game visuals
Save game	Writes save data to local storage, checks save data	Save data is in save file
Load saved game	Opens and reads save file	Makes game info available to update game
Exit options menu	Writes options data to local storage	Displays options are saved, shows main menu

Context Diagram



DFD - Level 1


User
Visual
Info
User
Input
Sprite locations,
Background image,
Inventory array
Game Info
Update
Game
1.5
Load
Saved Game
1.4
Display
Game
1.1
Events,
Game Info
Request
Save Data
Save
Data
User
Input
Process
Actions
1.2
Save Game
1.3
Confirmation
of Save
Local
Storage
Save Data
Save Bool,
Game Info

Link: http://hippo.emanuelschool.nsw.edu.au:8080/Emanuel/Moallem.Jonathan/info.html

INTRODUCTORY POST

This blog will contain development notes and updates on the upcoming point & click game, The Ballad of Gumption. Updates should be posted bi-weekly. Project documentation can be found on Hippo
For the next 15 or so updates, the posts will be retrospective, so to catch up to the current state of development.

COMMITMENT

Retrospective Update (27/10/2014)
I have officially started the project by adding a commit to the Github, there is no content yet. At this point, I am looking at doing a point and click adventure game. In light of me starting this major project, I have prepared a haiku:

the first commit,
even Drake seems to want,
systems documentation,

I will begin work on planning tomorrow, although, I know my general idea in building the mechanics, which will be getting the coordinates of the mouse and moving the sprite towards that location until they are the same. I also changed the readme file to have a more descriptive.

CHAPTORIAL

Retrospective Update (28/10/2014)
I have included my puzzle notes and added a MainProgram.py to the Github. The puzzle notes are a set of events that trigger other events in the game. So in effect it is a storyboard for the gameplay. I have chapter 1 and chapter 2 done, and implemented into a flowchart on GoogleDrive. When I finish writing the first three chapters, I will turn it into a pdf and share it on the Github. The initial version of the game will only consist of the first three chapters of gameplay, and I will continue adding more chapters as I go.
I also spent some time this week considering whether the first chapter of the game would be a tutorial. After careful deliberation, I decided that the first chapter of the story is too good to tutorialise, and therefore I will create a tutorial that occurs before chapter 1.

GITIGNORE

Retrospective Update (12/11/2014)
In this commit I have added a scan of my initial storyboard – the storyboard shows gameplay and menus. This shows every possible situation in the software I have currently planned. In this commit, I also deleted gitignore.txt. I have now included ProjectLog.md in the Github files. I have also concluded from speaking with Drake that I will also need to make another storyboard containing arrows that point to the flow of movement around the program.
The following is an image of the first panel, where I mocked up a UI design, amiable to presently surveyed users:

/R/NOCONTEXT

Retrospective Update (13/11/2014)
I have created a context diagram in which it outlines what data is passed in and out of the system. After an hour a procrastination, I figured out that it was exactly what I didn't think I thought it was. Essentially it is a data flow diagram in its most meta form – where in my case, only the user input and visual info is shown to be exchanged.
I have also begun considering the artiste for the game. As much as doing it in the original Gumption Fables art style would be nice, that style is my own, and therefore cannot be accurately reproduced by another artist. I do not believe I have the time or patience to animate it that way consistently. I dabbled with a graphics tablet for about an hour in order to make this decision. I most likely will animate it in my own artistic forte, pixel art.

APPROACH WITH EXTREME CAUTION

Retrospective Update (19/11/2014)
Began writing DevelopmentApproach.md and revised context diagram layout. I continued to receive user feedback based upon the current storyboard – having no complaints about its current state. The development approach I suggested in the documentation was an amalgamation of structured and agile approaches.
I have also begun to look into the use of music in the game. I can't seem to find much on how to implement sound files into Kivy.

LIKE SKYRIM?

Retrospective Update (21/11/2014)
I added user feedback, omitting a number of things, such as "a normal AND 'sexy' mode" and "concave joysticks on my controllers"?!?!
My favourite was 'translations to Esperanto'. Although I think that making my game into 'Gmod' is stupid, I guess the customer is always right, that being in mind, the game also now has to have 'boss levels' and add 'weapons loadouts'.
The hardest part I have encountered thus far is that most people think a point and click adventure is "like Skyrim?". Having people understand what my game is, is the hardest part of my project, and probably will continue to be.

THAT AIN'T NUTTIN'

Retrospective Update (26/11/2014)
This week, I got very little work done in the way of the project as I have three assignments due. I was able to get a bit more user feedback, which I will be incorporating into the rest of the documentation.
Soon, I will begin work on the project website, where I will host all the necessary documentation for Defining & Understanding The Problem.

HACKERS HAVE IT GOOD

Retrospective Update (27/11/2014)
This week, I added another section to my storyboard, outlining the flow of user interaction with the system. I have also added another feature to my project, where the user may navigate to the credits and the license from the options menu, where it will play as a scroll. After this is completed, there is a test at the end. I feel like the 'mini-game' is a necessary addition in recreating the Gumption Tales style.
I also wish to allow users to add custom assets to the game. Instead of doing this in-game, I will outline which files to replace from the README file. This was a user request and I feel I want to support hacking and hackers.

CSS IS BEAUTIFUL

Retrospective Update (03/12/2014)
Today, I made short work of the project website. It is currently being hosted on Hippo. The page, unlike some other's pages, is made from scratch. I would like to start by saying CSS is beautiful, and to clarify, beautiful is not the same thing as logical or efficient. I have been able to create EXACTLY what I was thinking of.
Finally, I would like to point out that the development website will have all the links to documentation, a project description, screenshots of the software and a shortcut to the Github repository.

AFFIRMATIVE ACTION

Retrospective Update (04/12/2014)
I worked on social and ethical issues today, completing everything that I feel is relevant to a point and click game. These issues included:

- Licensing of programs and assets
- Socially sensitive content
- Avoiding controversial topics or features
- UI consistency
- Ergonomics issues
- Economic exclusion

I will have to run most of these by Drake, as I am not too sure of its accuracy and phrasing. I have also made a number of changes to the project website to accommodate for new content.

2 META 4 U

Current Update (05/12/2014)
This is the first current update. I am currently writing the project log update about the project log that you are presently reading.
After some decision making, I came to the conclusion that I would be more inclined to write my log entries on time through the use of a blogging site. As WordPress is free and convenient, I decided to put the post backlog up today. I have updated the website to incorporate a link to this log, as can be seen on the documentation page under 'Project Log'. The following image will hopefully wig you out:

CAUGHT IN THE CROSSFIRE™

Current Update (08/12/2014)
Working on the identification of the problem, I looked at past major works to get a good idea of what kind of content is needed in that piece of documentation. I decided to add target system requirements, as I don't believe that users will want to have to run two Nvidia Titans on SLI to play a 2D point and click. At this moment, I don't really know what the size of the game assets will be, and so I have decided to allow for a target of 100GB of storage. This is well over what I will eventually need, but I didn't have even a rough estimate of what it might use and therefore I allowed it to take up the rest of my laptops free storage.
I have also begun work on the Gantt chart, which is based on the term calendars and assessment due-dates. I have no content for the next two stages, so I am just showing the timescales for submit dates.

CRUNCH TIME

Current Update (10/12/2014)
Due to the encroaching crunch-time, I am rushing to finish all the documentation. I have noticed that I rather enjoy and understand system flowcharts, and I'm fine with DFDs. I am still mostly clueless on how structure charts work.
I have finished 90% of the documentation, and need to wait to do the reflection until I have finished all of the documentation. The only document I have not at all started at this point is Test Data & Expected Output – I will probably need to get some information on what it is exactly. The rest of the work is just adding more content and fixing previous documents so that they are the best they can possibly be. In fact, I think I just realised how to complete the development approach document, BRB. Ok, good, another one down. I believe that part of the reason why this didn't get this work done faster is that I have run down almost two seasons of Arrow in the last 5 days.
Anyway, I'll probably make one more update before I hand in the first task.

Development Approach

I suggest on using a hybrid of the structured and agile approaches fro this specific project, possibly because it is the most fitting for a game.
The mechanics of the game will be done using the structured approach as I will know all of my requirements when I have finished my planning and design. On the other hand, there is a huge amount of content involved in the game, and as such, I will be implementing stages as the games development progresses. This combination will allow me to pass the requirements of the course, while at the same time, permit me to extend or contract the length of the game based on development cycles possible within the project timeframe.

The structured approach is best for this project, as it involves providing initial documentation with an outline of the final product, as well as has to plan for a deadline to the project, ensuring that it stays on schedule. The reason why I added agile to the approach is only to allow extra content to be added to the final project after completion of the main mechanics.

The structured approach consists of five defined stages:

- Defining & Understanding
- Planning & Design
- Implementation
- Testing & Evaluation
- Maintenance

I will be progressing from stages one to four, but according to the assessment, I will not be working on on maintenance. So, in the next stages of the project, I will be planning the game, and I will work methodically through it to testing making sure that I have a functional point and click engine. This is where the agile method comes into play, where it is repeated a number of times, adding new content - in this case, more chapters and features like voice acting. The structured approach is good for knowing exactly how long a project will take, while agile gives me the freedom to add more content, time permitting.
The mechanics of the game will be done using the structured approach as I will know all of my requirements when I have finished my planning and design. On the other hand, there is a huge amount of content involved in the game, and as such, I will be implementing stages as the games development progresses.

Social & Ethical Issues

Being a game, there are many social and ethical issues to consider when creating the project. The following summarises them in point-form:

Intellectual Property

Intellectual property is the creators original idea and all of the creators original ideas belong to him and cannot be infringed on by any other people, and therefore I will have to take this into account when using tools and resources in my project. Firstly I must must pay respect to the original creator of any used software or resources, while also obeying the license agreement that it has/usage laws.
I should also have a software license agreement, where I state my rights and responsibilities as the creator of the software - making sure that the user is aware of this prior to their use of my product.

Licensing of the various programs and modules that I will be incorporating into to projects development, such as Brackets, Kivy and the assets I will be using. I should also state the license terms explicitly and avoid threatening terms, such as 'we reserve the right to eat your guts'. This includes making my game Garry's Mod or Call of Duty, as I do not own the appropriate IPs.

Ergonomics

To prevent repetitive stress injuries, I may have to limit play session times to one click per play. Alternatively, I could add a cool-down to clicking, possibly neutralising the possible RSI. Should also implore that the user practise proper posture habits, similar to the warnings displayed on the start of Nintendo consoles - this can be in the form of . Eyestrain is also a prevalent issue, so making menu options large and easily seen is necessary. I must make UI consistent so to make sure that the user can easily understand and less effort will be wasted on training. This includes legible and readable text. Just to make sure the software is user friendly, I will also make sure to add loading symbols to indicate a process is taking place.

Inclusivity

To factor in inclusivity, I must make all content user-base sensitive, considering disabilities and backgrounds. Examples include using a contrasting colour palette, or not using racial slurs as menu options. The user request of 'sexy mode' may have to be omitted due to its controversial premise. I need to cater audiences by being culturally sensitive, economically sensitive, socially sensitive and gender sensitive. By making the game pay what you want, I will avoid excluding users by economic means. For a while I considered freemium, although I was too busy being exploited by EA to think for myself. I can also translate the game, time permitting so to cater to different social and cultural backgrounds. I need to take into account both genders when designing software to accommodate various thinking styles, such as preventing gender inequality, not only sexualised women, but also sexualising men in my game. This is an example being conscious of the social issues related to the portal of gender that may exclude a group of users. I will also need to cater to the vision impaired, by using high contract colour schemes to combat colour blindness as an issue in using my software.

Test Data & Expected Output

Bracketed test-data indicates location in storyboard (page 7)

Test Data	Expected Output
Open program (b.1)	Program starts and initialises
Move mouse	Cursor moves
Hover over menu option with mouse (a.1)	Menu option is highlighted
Click menu option (b.2)	Screen replaces previous menu with destination
Press quit (b.1)	Program closes to desktop
Press new game (b.2)	Screen replaces previous menu with in game screen and adds game-state to save file
Press continue (b.2)	Screen replaces previous menu with in game screen and retrieves game-state from save file
Click on game background (b.4)	Character moves to clicked location
Click on left-hand inventory arrow (b.4)	Inventory menu opens
Click home button (b.5)	Screen replaces previous game screen with main menu, saves game
Click on item in inventory (b.5)	Cursor is replaced with item
Click on left-hand inventory arrow while inventory is open (b.5)	Inventory menu disappears
Click on non-tangible objects (b.6)	Description for object appears
Click on tangible objects (b.6)	Item shows up in inventory
Click on FOV button (b.3)	Options menu is replaced with FOV menu
Click on licensing & credits button (b.3)	Options menu is replaced with licensing & credits
Slide FOV slider (b.8)	FOV slides
Click screen while dialogue is present (b.5)	Dialogue disappears

Reflection

The defining and understanding portion of the project is much like a Magic The Gathering tournament in the Sistine Chapel… actually I have no idea where that was going.
Oh wait, I have one! The defining and understanding portion of the project is much like an Excel Spreadsheet themed birthday party; lots of meaningless waffle exchanged to the tune of a number of incomprehensible charts.
Or is it like an OHS report in Guantanamo Bay? A tortuously meticulous evaluation that may result in years of psychiatric assessment. I get bonus marks for that, right?

I joke. I actually found some parts of this assessment 'fun', namely the website portion, being my strong suit. I built the website from scratch and learned a few neat CSS tricks from doing it. This is undoubtably further my ability to do web-dev in the future, such as building custom WordPress themes for clients. Which can actually be monetised by online Joomla and WordPress theme shops - so, I might do that in my free time.
The cover sheet took a long time for me to decide on a final design, as it is really a satire and clever statement about society. I would like to take the time to thank Matan for his input and helpful suggestions thought this portion of the assessment.
I found that completing the documentation for this assessment was rather easy this time around since I didn't have to think about implementation yet.
The identification of the problem was quite logical and made a lot of sense to me, being something aligned with what I think about primarily about, the pitch. I also realised that I should be differentiating my product from other point and click titles, and therefore I came up with the idea to focus on an intuitive inventory and selection system in-game.
For quality assurance, I decided to focus on four principles and addressed them and how I would o about working on them. This document was flagged by Drake for being brief, and therefore I gave further descriptions of each principle in order to meet requirements.
User feedback was also a concern of Drake's as I had only interviewed seven individuals. I found it would be hard to do anything but a personal survey, as almost every participant (all who were gamers) didn't know what point and click adventure games were as compared to an RPG adventure game like Skyrim. I found that dealing with users was very frustrating under these circumstances. This may be because the style of game I intend to make has slowly gone out of fashion throughout the years, although seeing a resurgence in the last few years with Leisure Suit Larry, Broken Age and The Journey Down. For the rest of the project, I will need to communicate more with my user base on things like interface.
The Gantt chart was actually a nice change of pace as I get many happiness from creating aesthetically pleasing things, one of those things being the Gantt. I included everything from defining and understanding, as well as my speculation for planning and design, implementation and testing and evaluation.
The data dictionary's creation was a bit 'wonked', as I haven't gotten into serious designing, and therefore I'm still not 100% about how the game is going to work, only a rough idea. This too was flagged by Drake, to which I admit, I don't have a clear idea of how it is going to work.
The storyboards were also nice, in that I was able to put my concept design skills to use, much like the way I do by concept sketches for previous work. These sketches will need to be updated to be more accurate in the next part of the assessment.
I don't know why, but I just **get** how flowcharts work. Flowcharts get me. Flowcharts are by best friend. Flowcharts don't judge. Flowcharts make sense. Flowcharts don't stab you in the back. Flowcharts is I and I is him. I derived much pleasure from hanging with flowcharts again.
I had many issues with my structure chart, as I want too sure how to format my program, as some of the parts are cyclical by nature. After getting some advice from Drake, I decided to create a 'run game' sub-section to show the cyclical parts.
The logbook was originally a document, but later on, I decided that the best course of action would be to put it on a WordPress, which would coerce me into updating it more frequently, which has been an issue in the past.
I believe that is all for now. Stay in school, kids!