Game Design Document

Johnny’s Revenge

COMP7903 -Game Design Fundamentals

COMP7051 – Intro to Computer Games Development

December 6, 2020

Team RJSJ Members

**Richard** **Tesch**

**John Dong**

**Shaun Tseng**

**Jaedin Dhatt**

Table of Contents

[Design Team 4](#_Toc57793825)

[Methodologies & Tools 5](#_Toc57793826)

[Methodologies 5](#_Toc57793827)

[Tools 5](#_Toc57793828)

[Concept Document 6](#_Toc57793829)

[Game Rules 6](#_Toc57793830)

[Operational 6](#_Toc57793831)

[Constituative 6](#_Toc57793832)

[Implicit Rules 6](#_Toc57793833)

[Core Loop 6](#_Toc57793834)

[Discover 6](#_Toc57793835)

[Fight 6](#_Toc57793836)

[Improve Deck 6](#_Toc57793837)

[Core Mechanics 6](#_Toc57793838)

[Cards 6](#_Toc57793839)

[Opponents 7](#_Toc57793840)

[Deck 7](#_Toc57793841)

[Hand 7](#_Toc57793842)

[Mana 7](#_Toc57793843)

[Health 7](#_Toc57793844)

[Card draw 7](#_Toc57793845)

[Rewards 7](#_Toc57793846)

[Drag/Drop 7](#_Toc57793847)

[Notable Mechanics 7](#_Toc57793848)

[Timer 7](#_Toc57793849)

[Score 7](#_Toc57793850)

[Stage 7](#_Toc57793851)

[Level 7](#_Toc57793852)

[Signifiers 7](#_Toc57793853)

[Shield 7](#_Toc57793854)

[Sound 7](#_Toc57793855)

[Animation 7](#_Toc57793856)

[High Score 8](#_Toc57793857)

[Bonus 8](#_Toc57793858)

[Analysis of Dynamics 8](#_Toc57793859)

[Battling 8](#_Toc57793860)

[Mana strategies 8](#_Toc57793861)

[Rewarding 8](#_Toc57793862)

[Scoring 8](#_Toc57793863)

[Game Aesthetics 8](#_Toc57793864)

[Sound Design 8](#_Toc57793865)

[Visual Design 9](#_Toc57793866)

[UI components 9](#_Toc57793867)

[Intended Target 10](#_Toc57793868)

[Player Profile 10](#_Toc57793869)

[Persona 10](#_Toc57793870)

[Interest Curves 10](#_Toc57793871)

[Decision Trees 11](#_Toc57793872)

[Game Design 11](#_Toc57793873)

[Software Design 13](#_Toc57793874)

[ALPHA 13](#_Toc57793875)

[Prototypes 16](#_Toc57793876)

[Playtest 16](#_Toc57793877)

[Design Flaws 17](#_Toc57793878)

[ALPHA 17](#_Toc57793879)

[BETA 17](#_Toc57793880)

[RELEASE 17](#_Toc57793881)

[Next Steps 17](#_Toc57793882)

# Design Team

A list of the team members and their contact information (student emails work) ***John***

|  |  |
| --- | --- |
| **Name** | **Contact Information** |
| **Richard Tesch** |  |
| **John Dong** | [**mahoushisaber@gmail.com**](mailto:mahoushisaber@gmail.com) |
| **Shaun Tseng** |  |
| **Jaedin Dhatt** |  |

# Methodologies & Tools

To design and develop Johnny’s Revenge the team encompassed certain methodologies and tool sets to enforce parody, reproducibility and assure success.

## Methodologies

Given the nature of the environment the team will work in to complete the game it was recognized that we needed a development solution that would provide quick requirements and accompanying solutions through cross-functional members. The environment would have to support quick informal but productive collaboration of members to iteratively produce a package that would meet the requirements of the project.

The team concluded the best methodology is to use the iterative waterfall method to tie us to the milestone set out in the requirements. For building our requirements and developing solutions the team adopted Agile practices. Using short sprints of once a week and progressing to 3 times a week near milestones or end of project we can complete the project with success.

In the document folder the team has recorded all sprint activity in the agile folder and include a link to Trello used for tagging assigned sprints.

TODO: SUBMISSION PACKAGE: Agile/Sprint folder containing one or many documents for the sprints. Including dates, agenda, sperate Trello link.

* Trello Link (https://trello.com/b/8JgkpTTS/project-management)

## Tools

* Discord – Meeting platform for all online meetings and discussions
* Unity – Free cross-platform game design platform.
* Blender – Used for 3D modelling of objects used in animations and reward screens. All used within Unity.
* C# – The language used to script objects within Unity
* Target platforms – Initial release only on Windows platform. Future targets after release are the mobile market.
* Version Control – GitHub repository using WEB and Git Desktop for access. Use of developer branches to develop and test assigned features before main code base merges.
* UML – Diagram of Implementation Classes, Hierarchy, Package Template, Package Diagram, Type Hierarchy, Game Design
* Testing – all developers playtest, play testers, and bugs recorded in sprints
* Bug fixing - pulled from sprints and corrected before next sprint

## Concept Document

***Shaun*** *– formalize use WORD heading for table of content to pick up*

* + Sketches
  + Our concept doc from presentation. Use the new core statement from this document.

## Game Rules

**(WIP) John**

### Operational

* Objective:
* To win the game a player must be able to defeat the opponent by eliminating their health and move on to the next stage until winning the final boss.

### Initial state:

* The player begins with a given deck of 20. Randomly drawing each turn.
* Starting with drawing 5 cards to the player, placing them in the player’s hand.
* Every time card dropping

### Moving a card:

* Cards can be repositioned at will.
* Only one card may be played at a time.
* A card cannot be replaced by another card.
* A card must be dropped for the game to continue, in either allowable drop zones battle/mana. For example, a roll of 2 requires the disc to move exactly two squares in any allowable direction.
* If the mana is lower than 25, the card will not be able to be upgraded. The card will be bounced to its original position.

### Player’s turn:

* Decide whether to use mana (Drag the card to the mana icon) for doubling the card effect or not.
* Drag and drop only one card to one of the drop zones (battle/mana), with any card in the player’s hand.
* You must make a play to move on.
* Your turn ends after moving a card into the drop zones.

### Constitutive

* The player selects the card wanted to play this turn, drag it into either drop zones (Mana zone and battle zone) to play.
* The Player’s ability to play cards is 1 card per turn as each action to the drop zones count as an end turn.
* The opponent will automatically draw a card at the start of their turn, prepare to play, and alternate to the player’s turn.
* The card the opponent will play during the current turn will be displayed.
* When the player has no cards left from their deck and is not capable of making a selection of the cards, the deck will be reshuffled and draw 5 cards to the player’s hand.
* The game is composed of 3 levels and each level contains 3 stages.
* When the player damages the opponent’s health to 0, the player won and is capable of making a selection between recover health or recover mana.
* After making a selection of recovering health/mana, the player will automatically go to the next stage. If the player is already at the last stage of the level and finished collecting reward cards, the player goes to the level selection screen with the next level unlocked and the previous level locked.
* The amount of the health/mana recovered will be calculated based on the score received in the current stage
* After finishing the last stage of each level, the player has an opportunity to drag one of two randomly picked cards from the reward deck that is originally exclusive from the player’s deck to their deck. (At the end of the second level, the rewards are expanded to “selecting 2 out of 3 cards” for the player to collect)
* A player cannot play more than 1 card each turn. Cards are not droppable when a card is already in place in the drop zone. until the turn is over.
* The timer will be reset at the beginning of each stage, counting the time used for finishing a stage. The time will be taken as a parameter into the algorithm of score counting at the end of each stage and level.

### Implicit Rules

* + Card tables with values and stats ***(HAVE write up each card?)***
    - Player
    - Reward
    - Enemy

## Core Loop

***Jaedin (we HAVE need explanations)***

### Discover

The player will begin by entering a new Level and/or Stage on their journey to the final boss.

### Fight

The main section of the Core Loop that consists of the player making decisions and playing cards to defeat the opponent.

### Improve Deck

After completing a level, the player will be able to add additional cards to their deck to aid them along the rest of their journey.

## Core Mechanics

***Shaun***

### Cards

### Opponents

### Deck

### Hand

### Mana

### Health

### Card draw

### Rewards

### Drag/Drop

## Notable Mechanics

***Richard***

### Timer

### Score

### Stage

### Level

### Signifiers

### Shield

### Sound

### Animation

### High Score

### Bonus

## Analysis of Dynamics

***Shaun (REVIEW)***

### Battling

### Mana strategies

### Rewarding

#### Stage rewards

#### Level rewards

#### How are they calculated

#### How does one select

#### How it ends the dialog view

### Scoring

#### How are they calculated

#### Stage and running game total

#### High score

## Game Aesthetics

### Sound Design

***John*** List

* Pirate themed

All sounds are stored in file directory Assets/Resources/Sounds (Including a folder “For Release”).

Note that BGM stands for Background Music and SE stands for Sound Effects. The listed credit with name is the ones specifically asked for crediting. The rest are credit free sounds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sound Name in Audio Manager | Sound File Name | Location Used | Description | Credit |
| Epic String | alexprost\_\_epic-strings | GameController.cs | BGM for Level 1 stages | alexprost |
| Theme | Theme | /Sounds (Used before Release) | Backup BGM | freesound.org |
| Theme2 | Credit(Mike TheTunk Woloszyn)Fight2 | GameController.cs | BGM for Level 2 stages | Mike TheTunk Woloszyn |
| Theme3 | Credit(Tyops) , the game (Defense of Lanosia) | /Sounds/For Release (Used before Release) | Backup BGM | Tyops,  Defense of Lanosia |
| Theme4 | Credit(Tyops\_Defense of Lanosia)Dark-battle | GameController.cs | BGM for Level 3 stages | Tyops,  Defense of Lanosia |
| Theme5 | epic-finale-loop | LoadScene.cs | BGM for Main Menu | freesound.org |
| Theme6 | Leaderboard-synth-pad (mp3cut.net) | HighScores.cs | BGM for High Score scene | freesound.org |
| Theme7 | tyops\_\_adventure-fantasy | AdventureMapCtrl.cs | BGM for Level selection | tyops |
| DrawCard | DrawCard | Draggable.cs | SE for Drawing card | freesound.org |
| ShufflingCards | ShufflingCards | Draggable.cs | SE for Shuffling card | freesound.org |
| DealingCards | DealingCards | Draggable.cs | SE for Dealing card | freesound.org |
| Sword | Sword | PlayerController.cs | SE for Attack animations | freesound.org |
| CardFlipRepeat | CardFlipRepeat | PlayerController.cs | SE for Non-attack animations | freesound.org |
| Upgrade | Upgrade | PlayerController.cs | SE for Applying mana | freesound.org |
| Button | Button | AdventureMapCtrl.cs | SE for pressing button | freesound.org |
| Pistol | Pistol | PlayerController.cs | SE for Pistol animation | freesound.org |
| PaperFlip | PaperFlip | Draggable.cs | SE for Non-attack animations | freesound.org |
| CardDrop | CardDrop | Draggable.cs | SE for Dropping card | freesound.org |
| DoubleShot | DoubleShot | PlayerController.cs | SE for DoubleShot animation | freesound.org |

### Visual Design

***Jaedin*** ***(What they are, how they are used)***

#### Theme

Johnny’s Revenge was developed around a pirate theme which is expressed through card design, visuals, and game audio. Things such as enemy artwork and animations are used to increase immersion and make the theme more apparent during gameplay. All of the artwork, models, and sounds were chosen or created with the intention of enhancing the theme of the game.

#### Scenes

(literally unity scenes and any dialogs)

##### Menu scene

The first scene that players will be presented with upon launching the game. Includes two buttons for either starting a new adventure or viewing current high scores.

##### High Score

Accessed directly from the Main Menu. Lists the top five local high scores and allows the player to also reset the scores if desired.

##### Adventure Map

Accessed from the Main Menu when starting a new adventure and when advancing between Levels during an adventure. Used to select a Level.

##### Battle Screen

Accessed through selecting a Level in the Adventure Map. Scene where all of the card battling and reward selection takes place.

### UI components

#### Health

Two health bars on the screen; one at the bottom for the player and one at the top for the opponent.

#### Mana

One mana bar in the center to the left of the primary card drop zone. Represents the player’s current mana. Serves as an additional card drop zone that is used to enhance cards for a cost of 25 mana.

#### Shield

Two shield icons on the screen to the right of the player and opponent’s health bars. Represents the player or enemy’s current shield amount.

#### Status visuals

##### Level / Stage location

Text component in the top left of the screen to help keep track of the player’s current progress. Displayed as <Level>.<Stage>.

##### Timer

Text component in the top right of the screen to keep track of the total time spent on the current Stage.

##### Signifiers

Image components which appear below card drop zones after a certain amount of time to indicate where players can drag and drop their cards.

##### Player Hand

Image component and drop zone used to display the player’s currently playable cards.

#### Score / Reward Dialog

##### Bar % of Health and Mana Reward

Two image components to represent the amount of Health and Mana players can choose as a reward after completing a Stage.

##### Text to Represent Game Results

Text components used to indicate results from the completed Stage. Includes time spent, damage taken, mana used, cards used, stage score, and total score.

##### Animated Chest to Associate Reward

Animated object used to represent gaining rewards and score

##### Buttons for Reward Selection After Stage

Button components used to select either Health or Mana as a reward.

##### Drag and Drop Cards

Card objects able to be dragged and dropped onto the reward zone to choose as a reward at the end of a Level.

## Intended Target

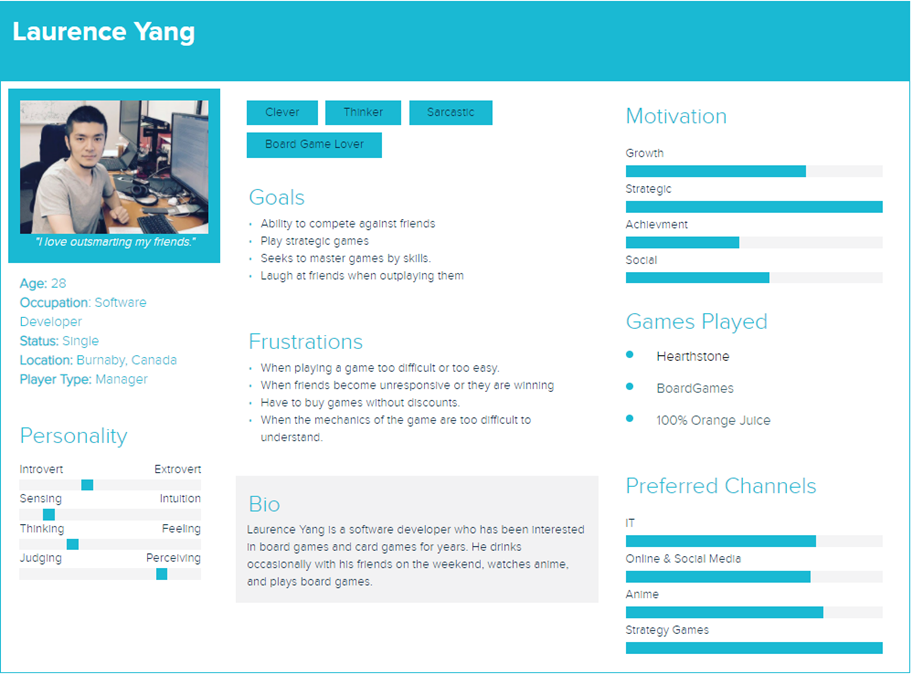
***John – brief introductory blurb (REVIEW have but needs formalizing and write up. I.e. research done persona and profile, any other analysis, how type and game are compatible)***

Everything from previous presentations

* Achiever
  + Achieve the highest score
  + Master the game
* Manager
  + Complex mechanics
  + Different cards to create different strategies
* Conqueror
  + Difficult boss fight
  + Scoring system

### Player Profile

Persona and Analysis



## Interest Curves

***Richard (enemy progression plus card reward)***

* Interest graph for each level
* Flow zone
  + Show each stage and how the progression increase
    - MaxHealth = levelNum \* 10 + (levelNum - 1) \* 5 + stageNum\*10;
  + How do we represent the deck building in the flow zone?

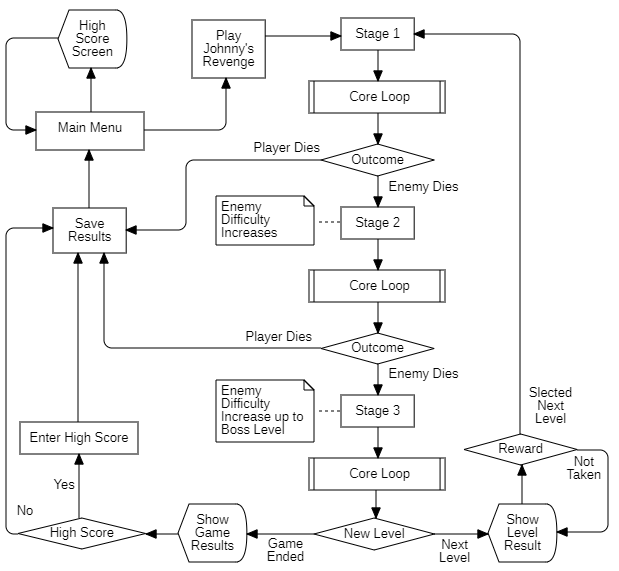
## Decision Trees

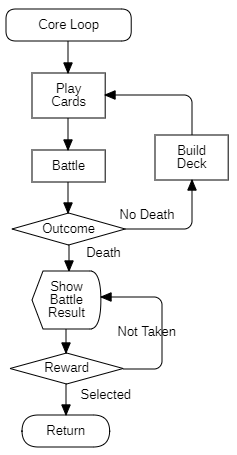
Our game design did not use decision tress as it is linear in design.

We have discussed this earlier in development and agreed that future development after release will build upon a string multi level adventure map that includes story telling. It is at this time that decision trees would be incorporated to allow for variations in choices provide stronger agency and increases immersion and habitation in the magic circle.

## Game Design

***Richard intro and explain the diagram***





## Software Design

***Richard***

Our software design was captured using UML to produce the necessary packages and subsystems to deliver the mechanics, dynamics, and aesthetic control.

The designs for each phase Alpha, Beta, and Release are recorded here are the

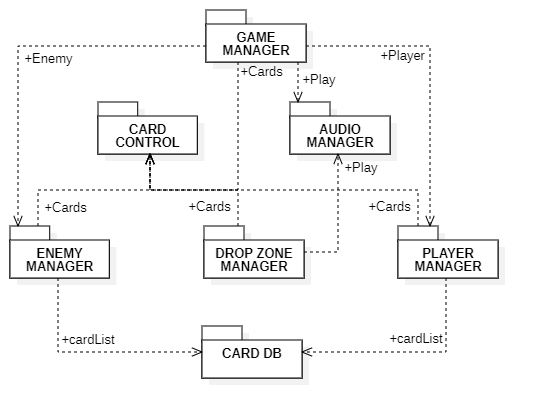
Package diagram: used to capture the high-level overall design of the system

Package template: used to start drilling down by showing the class block (no details) and how they perform simple interaction but only within a package.

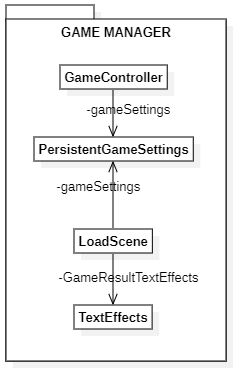
Diagram of Implementation Classes: Lastly, what UML traditionally called the class diagram. This is the meat and potatoes of each class and how the layered interaction between each other occurs. As it is of the highest granularity it also evolved the most over time and some spilled over into new classes of sub systems as they became to bloated or obtained cross purposes.

### ALPHA

Johnny’s Revenge : Model : Package Diagram



Johnny’s Revenge : Model : Package Template Diagrams :



#### Game Manager

Introduces the concept of an evolving game manager that will take care of all the higher-level details but all in one location. This include various support classes to help meet the package needs. It was decided that this effort of design early on would keep growth compartmentalized but time will tell.

##### Game Controller

Main class used as the conductor to orchestrate and coordinate all other packages to follow the constituative rules and some main core mechanics. For example, turn control, deciding when turns end, begin, or battles occur, scene navigation which includes the saving of persistent data to travel amongst scenes to share important values.

##### Persistent Game Settings

Classes only job is to have a collection of properties that can be abstracted from underlying platform storage. Once implemented it allows clients to manipulate hidden date at their needed layer of abstraction all the while making sure the data is present even when objects requiring it move out of scope or existence.

##### Load Scene

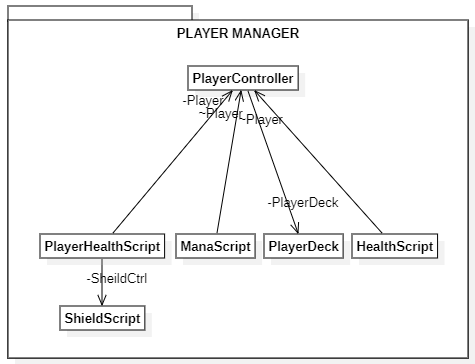
Created to handle movement between the only scenes restricted and controlled by the games design. When demands for switches occur it will take actions to assure that data and house tasks are dealt with before the switch.

##### Text Effects

The class is a cheap method to encapsulate some text effects such as fade, throb, flash or hide and show. Not a complete class as it may not be there in the final design but is a quick way to evolve what needs to be done with place holder affects.

#### Player Manager

Encapsulates the concept of delivering a manager that will deal with all the details of the player and its mechanics and dynamics.



## Prototypes

(Document all the prototypes that you will be making. There should be at least 1 per iteration)

***Jaedin (REVIEW have to pull out each iteration, screen shots and describe each change and what learned from it ) fill in the information below in yellow***

* Screenshots of each release
  + Shots must allow for single or combined demonstrate changes of
    - Mechanics
    - Dynamics
    - Aesthetics

### Playtest

***Shaun*** iterations and results first intro blurb then each milestone

* Format
  + - * How during proto, alpha, beta, & release
        + Surveys
        + Interview
        + What did each question try to answer
      * Results
        + pie charts
        + Where questions duplicated most recent result and then deltas
        + What it taught us

Past efforts

List beta and previous

Future goals

Each iteration leaving out release

#### ALPHA

#### BETA

#### RELEASE

## Design Flaws

***Jaedin (REVIEW Include ALPHA, BETA, & RELASE presentation playtest findings and what they drove. This is not actual playtest results as that is in the previous section. This is the “what did we learn summary” for flaws and if we took them on into the next release – documented in the “next steps” section.) Remember write an intro blurb of what is documented (probably after its done) and the follows the meat and potatoes you fill in.***

* + Insert and expand on the presentation” Future Scope”
    - Where they came from
    - What hope to achieve

### ALPHA

### BETA

### RELEASE

## Next Steps

#### ALPHA

#### BETA

#### RELEASE