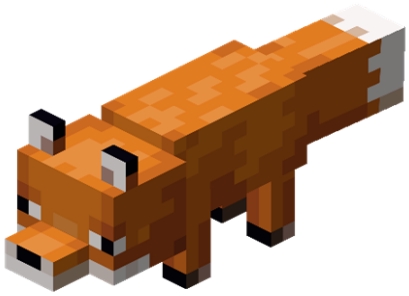
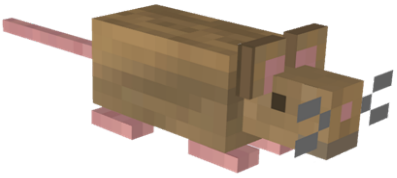
Score 85/100

Pac-Rats



Pac-Rats



A Browser Based Implementation of Multiplayer Pac-Man

Prepared by
Jonathon Repta, Jennifer Alonso, Edward Liang, Amal Syed
at the
University of Illinois Chicago

October 2020

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

Pac-Rats is a multiplayer implementation of the arcade classic, Pac-Man. The purpose of Pac-Rats is to offer a unique twist on a classic game and, in doing so, reinvigorate the interest of players who are familiar with the original, but are looking for something fresh. In order to achieve our goal of revitalizing Pac-Man, we will introduce a number of new features. Features such as new game modes (co-op, free-for-all), an extensive leaderboard system, and community based features like map creation/sharing. Pac-Rats will offer an accessible experience that will be enjoyable to players of all skill levels.

Users can easily link up with friends and do not require a high end pc to play. Onced signed-up they will be registered into the database and can login anytime as long they have a computer and internet. This game is intended for all ages and since it is inspired by the classic pac-man game, the mechanics/controls are easy to learn. Even a child will be able to pick it quickly and play with each other.

2 The Purpose of the Project

The project aims to create a game that combines traditional Pac-Man style gameplay with in order to create a game that will be accessible to a wide range of players whilst still maintaining the attention of the serious "hardcore" gaming audience. The hope is that the project will be a profitable venture for the publisher whilst being an enjoyable experience for a variety of players.

2a The User Business or Background of the Project Effort

The business that the client is a part of is the gaming industry. Specifically the section of the games industry which is in charge of remaking/remastering classic games. This is relevant to the project since the developers need to understand the wants and needs of the average gamer and balance that with the requirements given by the stakeholders.

2b Goals of the Project

The goal of this project is to take a classic game and breathe new life into it by applying modern gaming practices. Furthermore, the project aims to make a classic game profitable again by retrofitting modern monetization schemes into a game that didn't have them originally. Essentially the project aims to take the addicting retro gameplay of Pac-Man and combine it with innovations in multiplayer gaming that have come since.

2c Measurement

Pac-Rats will keep detailed usage statistics about users in order to monitor audience retention. Things like the number of new users every day/month/week that log on, the amount of time the average user spends on the app, etc. The success of our product will be measured against the user retention of the average Steam game in order to see if our product is able to compete with comparable dedicated gaming products. If our

game is able to match the retention rate of the average Steam game (or better yet, exceed it) we will have achieved our goal.

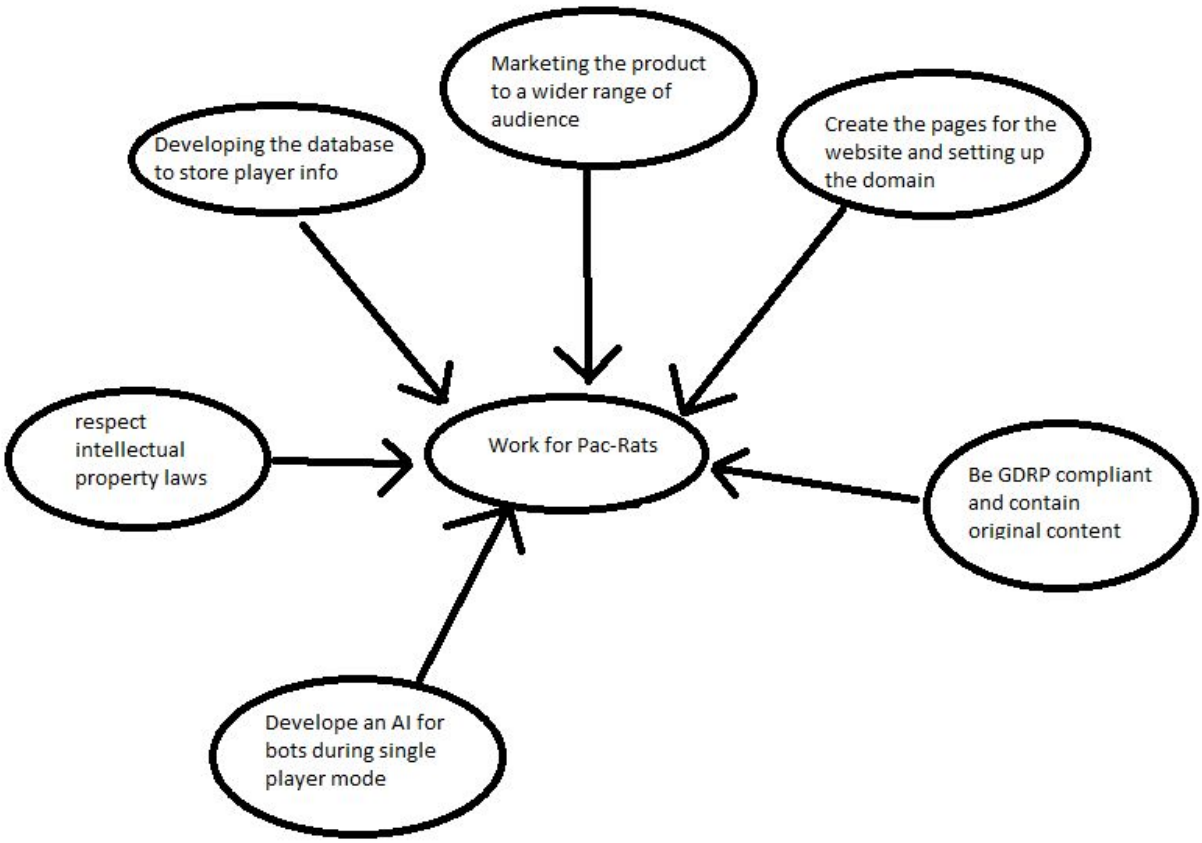
3 The Scope of the Work

The business is that of the games industry. Specifically, recreating old classical arcade games that everybody knows growing up and featuring new and original content. The work can be addressed with this project by recreating a traditional pac-man game from the arcade with pac-rats.

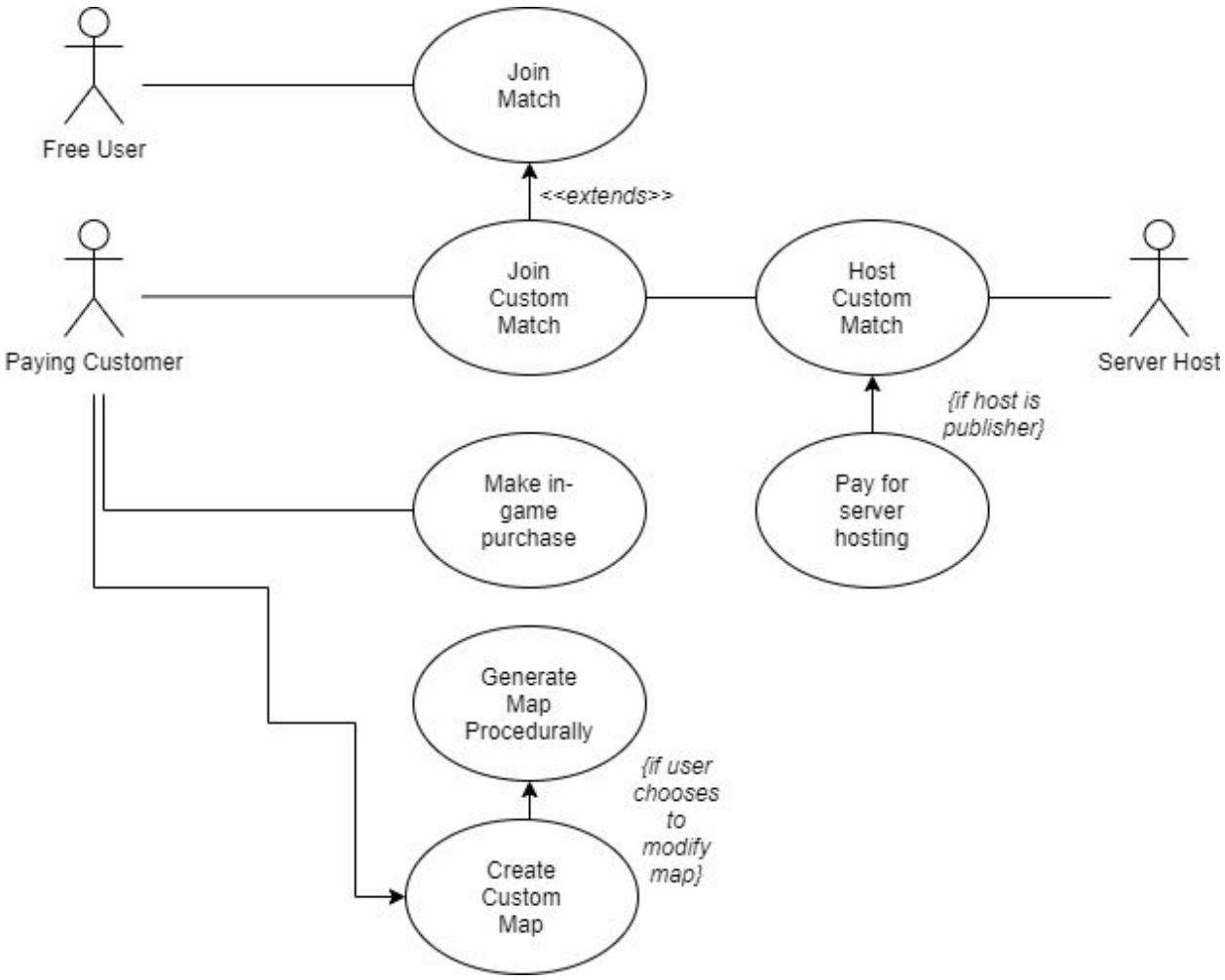
3a The Current Situation

Presently, the client has ported a standard version of Pac-Man to a wide variety of dedicated gaming consoles and mobile devices. All of these releases contain monetization schemes which either involve upfront payment for usage of the software, or are monetized via in game advertisements.

3b The Context of the Work



3c Work Partitioning



3d Competing Products

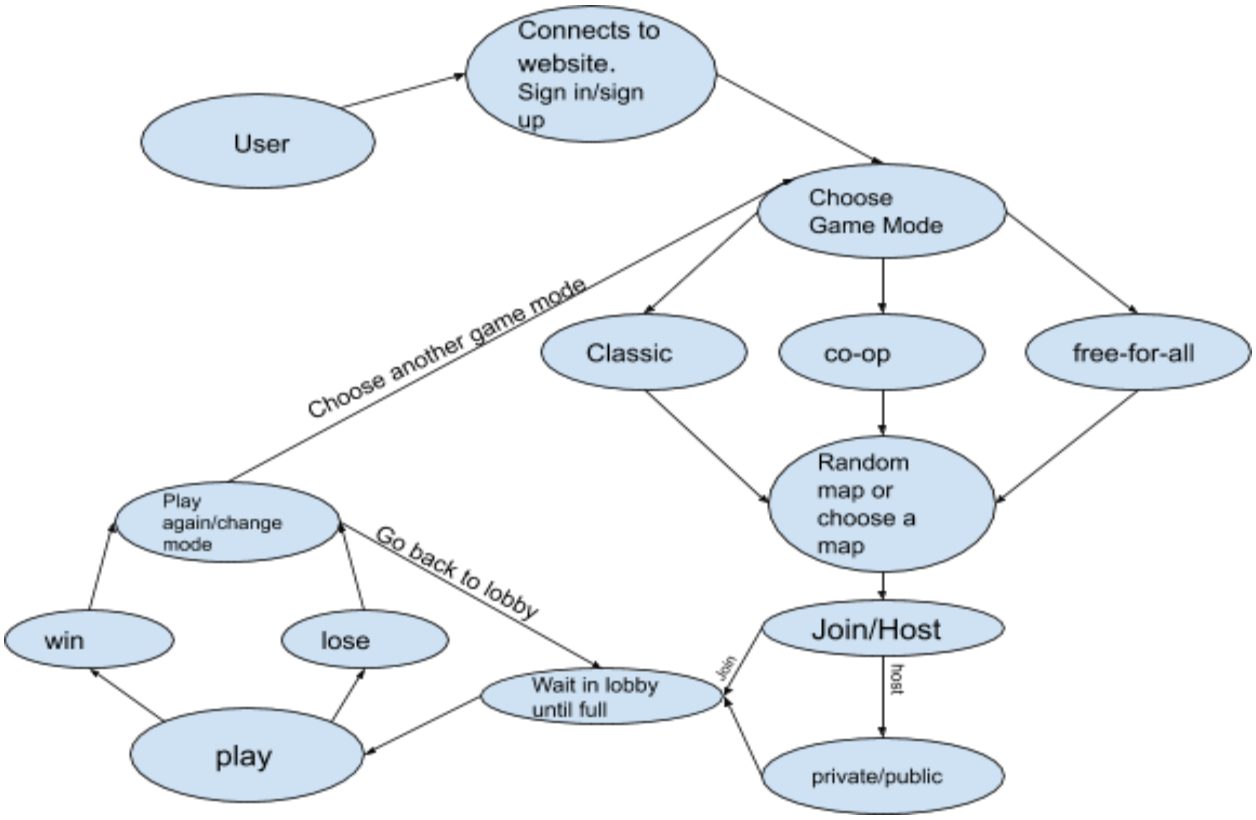
The game will be competing in the casual, free-to-play, market. Direct competition includes other Pac-Man style games, whereas indirect competition includes free to play mobile games in general. Our product is still useful as there is still room to innovate within the free-to-play market. By revilitazing a classic game we would be able to capture a market of players who would not otherwise be interested in a more serious/novel game.

4 The Scope of the Product

The product must handle all aspects pertaining to the product's gameplay. This means that the product must allow for the user to join a server, host their own custom server, browse community made maps, create their own maps, browse the leaderboards and share their creations with the community. The product will not cover the creation of

things outside of the scope of the game. This includes the creation of a payment platform, the infrastructure needed for server hosting, etc.

4a Scenario Diagram(s)



4b Product Scenario List

Scenario name	External Actors	Relevant information
Account creation	users, customers	user creation will be the same regardless of whether the user is a customer
Gameplay mode selection	users, customers	
Hosting/Joining game	host, publisher	The user can opt to host the game themselves, or pay for hosting from the publisher
Waiting in Lobby	users, customers	
Game Starts	users, customers	
Returning to lobby or start new game	users, customers	

4c Individual Product Scenarios

1. New users create signup through the website and returning players sign in.
2. Choose one of the many game modes to play (co-op, free for all)
3. Enter a seed to randomly generate a map for the match or choose from one of the many community made maps or one of your self made maps from map creator.
4. Choose to host or join a game. If hosting, choose to make it private for friends to join or public for random players. If joining a game, wait in the lobby.
5. Wait for the lobby to fill up to the required number of players and host to start.
6. Play game
7. Choose to remain in the current lobby or return to the main menu to try another game mode.

5 Stakeholders

5a The Client

The client would be a game company/publisher such as Bandai Namco, Koei Tecmo, Capcom, etc. Any company interested in revitalizing one of their existing IPs and applying modern gameplay mechanics and monetization schemes in order to capitalize on a growing market would be interested in our product.

5b The Customer

The customer of the product would be a user who is invested in the game to a point where they would be willing to spend money upgrading their experience. Upgrades

such as private servers hosted by the publisher, cosmetic upgrades like player skins, and the removal of in game advertisements would allow customers to differentiate themselves and enhance their experience.

5c Hands-On Users of the Product

Hands on users of the product would be both customers and people who play the game free of charge. In contrast to our paying customers, “free” users are less involved in the game and are more likely to be a casual gamer who plays in order to kill some time -- rather than to be involved competitively. This is relevant as customers interests may conflict with the interests of the average user.

5d Maintenance Users and Service Technicians

Since the game will run entirely in the browser, the game will require the user to do absolutely no installations/updates/etc whatsoever. If the player wishes, they will have the option to host their own private server or pay a subscription for hosting from the publisher.

5e Other Stakeholders

Beta-testers: This group will test out the game before it is fully released for the general public. They test out the game for still remaining glitches and bugs that still need to be addressed.

Marketing Manager: They are responsible for advertising the product. This game can only reach so many players by itself and from friends/family. Marketing Managers help spread it to other players from various backgrounds and create a diverse community.

5f User Participation

User participation is not required for development of the project, however, the developers may find it useful to provide polls for the community to determine which direction the game should head in in order to improve player retention and increase the quality of the product.

5g Priorities Assigned to Users

Paying customers are given much higher priority than the average user as they are the ones who are most invested in the game. Furthermore, they are the ones who are most likely to continue to spend money on the game . Whilst a free user may install the game and play it for a week, a premium user is likely to continue playing the game for a far longer period of time.

6 Mandated Constraints

6a Solution Constraints

The game must be color blind friendly in order to be accessible for color blind users. The game must not rely on auditory cues in order to accommodate users who are hard of hearing or are in an environment where they cannot hear audio cues (public transit)

6b Implementation Environment of the Current System

This game operates exclusively within the browser, meaning that it is inherently compatible with any device that is capable of running a web browser. Because of the web based nature of our product, developers are afforded the ability to easily create *native* versions of the product through the use of software like React Native, Electron, etc. In regards to hardware requirements, since the game will retain a retro aesthetic it won't require a powerful machine to run. All the user needs is a stable internet connection and a way to interface with the game.

6c Partner or Collaborative Applications

The product must be compatible with all major browsers. Specifically, it should be compatible with browsers that account for at least 80% of the world's internet users. This means no use of outdated software like Flash, and no cutting edge software like WebAssembly. The game should be written using a game engine that utilizes JavaScript and HTML5 -- to maximize compatibility.

6d Off-the-Shelf Software

No commercial off the shelf software is required for our product. The product will instead rely entirely on free, open source, software for auxiliary code outside of what is needed for the game itself. Relying on free software will allow development costs to stay down whilst still utilizing high quality libraries.

6e Anticipated Workplace Environment

The product is intended for a wide variety of environments, in accordance with the games philosophy of accessibility. Because of this, information must be conveyed to the user in a clear and concise way that doesn't necessarily require any sort of auditory feedback.

6f Schedule Constraints

The developers must focus on the most fundamental elements of the application before moving onto smaller issues. This means getting the foundations of the program functional before focusing on niceties. Graphical fidelity comes second to a functioning game and in game purchases come second to graphical fidelity. If the game is too broken, players will not be interested in the product. If the product is too frustrating to look at/use, players will not be interested in spending money on the

product. Because of this, our priorities must be gameplay, then aesthetics, then monetization.

6g Budget Constraints

The budget given will be relatively small due to several factors. Firstly, since the developers will not be required to write any auxiliary structural code, the development team will not need as many members. Secondly, the aim of this project is not to push graphical fidelity or story but instead create a fun game that appeals to a wide audience.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Pac-Rats: Equivalent to Pac-Man in a typical game of Pac-Man, only in *Pac-Rats* there will be multiple player controlled Pac-Men on a single board.

Foxes: Natural predators of Pac-Rats. They are the equivalent to ghosts in Pacman and work collectively to hunt down the Pac-Rats.

Free-for-all: A game mode in which every rat is pitted against one another. There are no foxes on the board, instead, Pac-Rats compete with one another.

Co-op: A game mode in which multiple Pac-Rats play alongside one another to achieve the highest score possible without any opponents getting in their way.

7b UML and Other Notation Used in This Document

The database(s) must be blueprinted thoroughly through the use of an ER-diagram. Use case diagrams must be employed to model both casual player interaction as well as dedicated user interaction.

7c Data Dictionary for Any Included Models

The project will use a relational database to store all of the data required for gameplay. The specific flavor of the database is down to whichever implementation the developers are most comfortable with, however, they may not use a non-relational database. Map recommendations will not be on a star system, but instead be either "liked" by the user or "disliked". This is because both Youtube and Netflix have discovered that almost every user rates their experience either 5 stars or 1 star.

8 Relevant Facts and Assumptions

8a Facts

- The game must comply with the privacy laws of every market the game is intended for. Most notably, the game must be GDPR compliant.
- The game must not infringe on any other publishers intellectual property. This means no stolen assets, copyrighted terms, gameplay systems, software, etc.

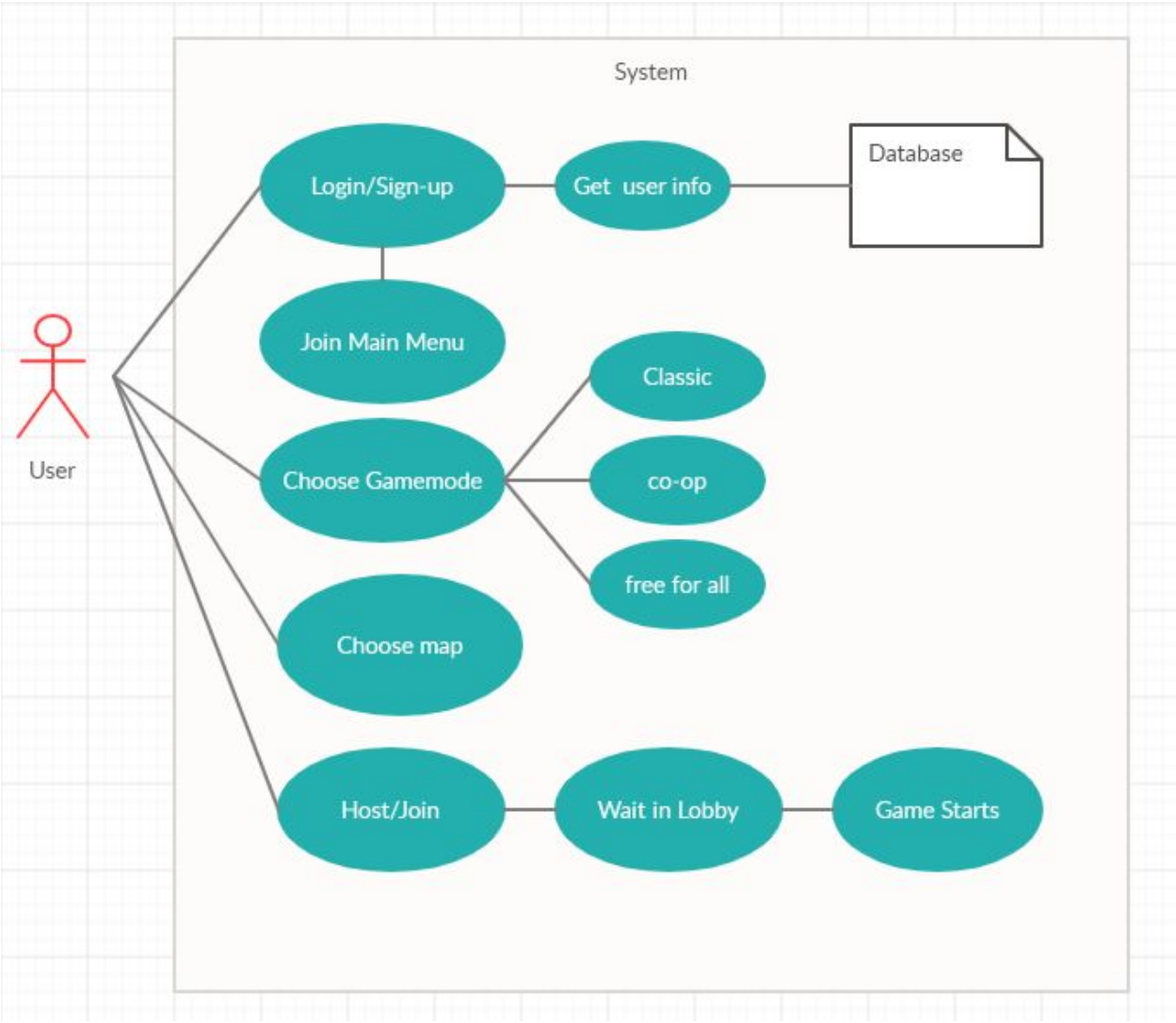
8b Assumptions

- 90+% of the players will not spend any money on in game purchases of any variety
- The player is familiar with how a standard game of Pac-Man
- The player has a stable internet connection

II Requirements

1 Product Use Cases

1a Use Case Diagrams



1b Product Use Case List

1. Signing in
2. Gameplay loop

1c Individual Product Use Cases

Use case ID: 1 Name: Signing in pre-conditions: User has internet access post-conditions: User is able to play Pac-Rats Initiated by: User (either F2P or P2W) Triggering Event: Visiting the Pac-Rats homepage Additional Actors: None
Sequence of Events: 1. User visits the domain that hosts Pac-Rats 2. User reaches Pac-Rats landing page 3. User clicks on the “login” option so that they can log in to their previously existing account 4. Server responds with a login 5. User signs into their account using their login credentials 6. User is redirected to the gameplay scene of Pac-Rats and is able to join a game
Alternatives: The user does not have an account and instead would like to create one Exceptions: There is an internal server error when accessing the user’s credentials/creating the user’s session information

Use case ID: 2 Name: Joining a game

pre-conditions: User has internet access

post-conditions: User is playing a game of Pac-Rats

Initiated by: User (either F2P or P2W)

Triggering Event: Signing to an account

Additional Actors: There has to be at least one other user who is also in the lobby

Sequence of Events:

1. After the user has successfully logged into a Pac-Rats account, they will be greeted with a list of gameplay modes
2. The user will select the gameplay mode that they want to play
3. Once the user has selected a game mode, they will be shown several additional options regarding how they would like to play said game mode.
4. If the user has selected the default/standard options, they will immediately be put in a lobby
5. If the user has specific options for how they would like to play a game (they would like to play with specific people/have a privately hosted game) they can proceed to enter those options
6. The player will wait a certain period of time in the gameplay lobby and will be able to chat with other players.
7. Once the countdown has finished, the players will begin their match

Alternatives: The user decides they would rather play a different game mode

Exceptions: The user is unable to join any game because there are no other players currently available to play with.

2 Functional Requirements

F0# - Account Creation

Description: The player should be able to sign up for an account.

Rationale: In order to keep track of important information regarding many aspects of the game, it is important to have a functional account creation/maintenance system.

Fit Criterion: Users should be able to create an account and their data should persist across sessions (be stored permanently within a database).

Acceptance Tests: Account handling

F1# - Account Sign in

Description: Similar to there being an account creation requirement, the system will also need to have .

Rationale: This is important for both obvious and non obvious reasons. In contrast to a game like slither.io that has you sign up for an account every session, our game will require that there is a sign in system so that users can save their progress.

Fit Criterion: Users should be able to sign in to a previously created account and access all of their saved information.

Acceptance Tests: Account handling

F2# - Joining a game

Description: The player should be able to join a game with options matching their gameplay preferences.

Rationale: In order to have a functioning game, we need to have a system in place for users to select the type of game they are interested in playing and to have them be placed into a lobby that matches their gameplay preferences.

Fit Criterion: Users should be able to join a game with all options reflective of their choices.

Acceptance Tests: 2

F3# - Gameplay loop

Description: The user should be able to to play a game of Pac-Rats with other users

Rationale: In order to have a properly functioning game, it is paramount that the gameplay loop works correct and all player actions are perfectly accounted for and coordinated cross connections. This means having a system of real time bidirectional communication between clients (users)

Fit Criterion: The user experience during gameplay should be equivalent to as if they were playing a multiplayer game locally rather than over the internet.

Acceptance Tests: gameplay loop

F4# - Completing a game

Description: Information regarding a completed game should be stored and reflected in other aspects of the game's design.

Rationale: In order to have an enjoyable user experience the results of a game should persist between sessions and be stored for review. This includes keeping track of player score, the amount of games played, and the ability for users to share their gameplay snippets with other players so that they can be viewed.

Fit Criterion: The user experience during gameplay should be equivalent to as if they were playing a multiplayer game locally rather than over the internet.

Acceptance Tests: gameplay loop

3 Data Requirements

U1# - Account Creation

Description: The system must provide a mean for all users of this product to create an account with a unique name/id and email

Rationale: Having unique id/email helps the system to track the users easier and create reports.

Fit Criterion: The name/email of every user must be unique and contain no duplicate in the database.

Acceptance Tests: Account handling, data persistence

4 Performance Requirements

4a Speed and Latency Requirements

U2# - Gameplay Latency

Description: The system must limit lag between stimulus and provide fast response time.

Rationale: We do not want latency in the system to produce a horrible UI experience for the user.

Fit Criterion: The login process to the server should not take more than 300 milliseconds.

Acceptance Tests: Gameplay loop

4b Precision or Accuracy Requirements

U3# - Server client communication

Description: The system must provide the means for the user to be able to precisely retrieve and render information.

Rationale: Handing out misinformation may result in bugs or crashes in our system

Fit Criterion: The system must provide means for the user to play the game in real time with other players..

Acceptance Tests: Gameplay loop, data persistence

4c Capacity Requirements

U4# - Concurrent Players

Description: The system must be able to support many user connections at the same time.

Rationale: This is an online multiplayer game and it would not function as one without multiple connections to the server.

Fit Criterion: The game, per match, holds upward of 10 connections concurrently.

Acceptance Tests: Gameplay loop

5 Dependability Requirements

5a Reliability Requirements

U5# - Data integrity

Description: The system must provide the means to safely store user information from system failures

Rationale: Unreliable systems might result in unnecessary data lost in the server.

Fit Criterion: If a system failure occurs, the user can safely login back to the with all their previous info once the system is back operating.

Acceptance Tests: Data persistence

5b Availability Requirements

U6# - Game availability

Description: The System must be on active for every day of the year unless due to mandatory system maintenance.

Rationale: If the system is kept active only up to a certain time zone, then other players at a different time zone will not be able to play the game.

Fit Criterion: Users in the United States will be able to play with users from Britain any time of the day as long there is stable inist ID# and/or names here . . .ternet.

Acceptance Tests: Gameplay loop

5c Robustness or Fault-Tolerance Requirements

U7# - Maintenance notifications

Description: The system must deliver messages to all users in game when maintenance is required.

Rationale: This is to keep some functionality intact and notify players with a warning before the systems go on maintenance.

Fit Criterion: Upon detecting a fault the system will go on a 1 hour maintenance break.

Acceptance Tests: Data persistence

5d Safety-Critical Requirements

U8# - Safety compliance

Description: The system must provide messages for the user to view the number of hours they have been in game.

Rationale: This is to prevent any user from abusing the game and playing more than over the recommended amount.

Fit Criterion: For users playing this game for more than 3 hours a message will appear to remind them the total hours played today.

Acceptance Tests: Gameplay loop, user experience

6 Maintainability and Supportability Requirements

6a Maintenance Requirements

U9# - Game maintenance

Description: The game will be maintained by its administrators, in charge of the build of the administration's technical aspects will be the Sysadmin.

Rationale: Having a dedicated team of administrators will alleviate other stakeholders the need to maintain the site. Having a dedicated Sysadmin will allow for the administration team to handle technical issues without the need to contact the development team.

Fit Criterion: Have a team of at least 5 administrators and one system administrator.

Acceptance Tests: Data integrity, user experience

6b Supportability Requirements

U10# - Customer Support

Description: The system must provide the means for the user to be able to write an inquiry. There must be a team of people with experience in customer support which will read and answer customer inquiries.

Rationale: We want a product to have reliable customer service and be able to gain the trust of our customers. In order to do so we must ensure that there is a reliable team of people who will be able to answer as soon as possible.

Fit Criterion: There should be a team of people available 24 hours a day on weekdays if possible all seven days of the week.

Acceptance Tests: User experience

6c Adaptability Requirements

U11# - Devices

Description: The system must provide the means for the game to be able to run on devices that support browsing.

Rationale: Since the game will be run on browsers such as Chrome, Firefox, Edge, etc. Devices like tablets and computers should be able to run the browser game.

Fit Criterion: A library computer is sufficient for entering the game with the proper url.

Acceptance Tests: User experience, gameplay loop

6d Scalability or Extensibility Requirements

U12# - Concurrent users

Description: The game must be able to host thousands of users simultaneously without the server getting overloaded

Rationale: In order to allow for the potential of the game growing in user base drastically, we must create a scalable system that allows for thousands of users to play the game concurrently.

Fit Criterion: Write unit tests to test strain on the system with 10,000 concurrent users.

Acceptance Tests: Data persistence

6e Longevity Requirements

U13# - Lifetime of Product

Description: The system must run as long as there are customers playing and supporting the game

Rationale: The browser game will be developed and further supported as long as there are people playing and supporting the game. Since we need to have enough revenue to keep the game up and running.

Fit Criterion: Every year we will have to review if the game is still being played or people are purchasing in game products to continue supporting the game.

Acceptance Tests: User experience

7 Security Requirements

7a Access Requirements

U13# - Accounts

Description: P2W (pay to win) users will have access to more advanced features than F2P (free to play) users.

Rationale: This will incentivize users to pay for premium P2W accounts.

Fit Criterion: Separate features into those available to P2W and F2P users.

Acceptance Tests: Account handling

7b Integrity Requirements

U14# - Database access

Description: Database routes should only be given write access to the database if absolutely necessary, otherwise, they should be read only.

Rationale: This greatly lowers the potential of data loss due to an SQL injection attack.

Fit Criterion: Routes should be split into two categories, read only, and read/write.

Acceptance Tests: Data integrity

U15# - Database deletions

Description: Database deletions should be exclusively soft deletes unless regional laws conflict with.

Rationale: Soft deletes will increase database integrity by keeping all information stored within the database, even if a mistaken deletion occurs.

Fit Criterion: Deletions from the database should only occur after a certain amount of time has passed from the soft delete.

Acceptance Tests: Data integrity

7c Privacy Requirements

U16# - Privacy of the Costumer

Description: The system must provide the means to safely store confidential information for the user.

Rationale: This is to protect the safety of all the users and integrity of the business.

Fit Criterion: The system must safely store all surveys conducted and provide an option to be anonymous for any of our surveys.

Acceptance Tests: User experience, data integrity

7d Immunity Requirements

U17# - Password protection

Description: The system must provide the means for all user sensitive information.

Rationale: This is to reduce hackers from possibly going into the database from stealing sensitive information.

Fit Criterion: The password entered must fit the length requirement and certain symbols and characters need to be added before creating an account. The system must prompt the user to enter their credit card information every time before making an in game purchase.

Acceptance Tests: The system must provide a safety procedure whenever a user has to deal with something confidential (account creation, in game purchases).

8 Usability and Humanity Requirements

8a Ease of Use Requirements

U18# - Easy usage of UI

Description: The system must provide a user interface for the users such that it is easy to use properly and hard to use improperly.

Rationale: Players can easily pick up the game and start experiencing the game without much trouble.

Fit Criterion: All symbols and icons in the game must be easily distinguishable.

Acceptance Tests: The game will conduct a survey to check for ease of use in the product.

8b Personalization and Internationalization Requirements

U19# - Personalization

Description: The systems must provide the means for user to modify some of the controllers for gameplay.

Rationale: This is to improve the ease of use in the product when the defaults in the system are not compatible with the user.

Fit Criterion: Users can edit controls made to the keyboard to move the avatar.

Acceptance Tests: All the letters, numbers, and symbols in the keyboard must be configurable.

8c Learning Requirements

U20# - Easy Learning

Description: The system must provide the means for the user to easily pick the game without previous experiences or spending much time at a tutorial.

Rationale: If a game is too difficult to learn then it will reduce any incentives for users to try the it out

Fit Criterion: The tutorial supplied will not take more than 3 minutes to go through

Acceptance Tests: The game will track the number of times each user will go through the tutorial.

8d Understandability and Politeness Requirements

U21# - Family friendliness

Description: The system must provide the means to censor and block out profanity displayed in messages.

Rationale: This is to protect younger audiences. The purpose of this product is to not promote profanity/racism/discrimination

Fit Criterion: All inappropriate words are censored out and displayed with asterisk

Acceptance Tests: A list of illegal words will be taken from a dictionary and applied to the algorithm to check for possible errors.

8e Accessibility Requirements

U22# - Screen reader compatibility

Description: The game must be fully compatible with screen readers, with labeled buttons and all images having alt text.

Rationale: This will make the game more accessible to visually impaired players, giving it a wider audience and potentially more customers.

Fit Criterion: Game should be accessible to users who use a screen reader

Acceptance Tests: User experience

U23# - Color blind friendly color schemes

Description: The game must have color schemes that use high contrast to allow even colorblind users to be able to distinguish elements.

Rationale: This will give the game a wider audience and thus more customers.

Fit Criterion: Game should be able to be played by users with all types of color blindness

Acceptance Tests: User experience

8f User Documentation Requirements

U24# - Usage Documentation

Description: Although the game will focus on being accessible without any documentation, we will nonetheless provide documentation to any users who find it necessary.

Rationale: In order to keep our game accessible, we must provide thorough documentation regarding game operation. This includes not only how to play the game, but also how to join a specific server, submit gameplay snippets to the highlight browser etc.

Fit Criterion: Documentation should be accessible to any user

Acceptance Tests: User experience

8g Training Requirements

U25# - Moderator training

Description: What specific users need to know in order to moderate the game

Rationale: Moderation is key when operating any sort of multiplayer game. In order to have a good community, the moderators must understand how to use moderation tools to their full potential.

Fit Criterion: Moderators must be quizzed on their ability to use the moderation tools before they can become moderators.

Acceptance Tests: Gameplay loop

9 Look and Feel Requirements

9a Appearance Requirements

U26# - Friendly Branding

Description: Website branding should focus on friendliness and accessibility over professionalism

Rationale: In order to convey the accessible, friendly, and open nature of the game, the branding must reflect that. These design goals can be reflected in the branding by employing the use of colorful colors, rounded sans-serif fonts, and a cute mascot.

Fit Criterion: Use a colorful, yet consistent colors for UI, have a cute mascot

Acceptance Tests: User experience

9b Style Requirements

U27# - Friendly Styling

Description: Website style should focus on friendliness and accessibility over professionalism

Rationale: In accordance with the game's design philosophy of accessibility, Pac-Rats must present a forgiving, friendly, and intuitive UX. Making the UX friendly will increase accessibility to users less familiar with online games.

Fit Criterion: Avoid using serious technical language, allow for undoing/redoing in regards to state management.

Acceptance Tests: User experience

10 Operational and Environmental Requirements

10a Expected Physical Environment

U28# - Devices and WiFi

Description: User must have a device which can support browser games, such as a computer, phone, or tablet and have strong wifi connection.

Rationale: We want audiences from many different backgrounds and including the ones that do not have access to a computer.

Fit Criterion: The game will be cross platformed from pc to android and apple phones.

Acceptance Tests: Accessibility

10b Requirements for Interfacing with Adjacent Systems

The product has no particular partner applications it is required to interface with.

10c Productization Requirements

U29# - Internet Connection

Description: The system must provide the means for the user to connect from the browser through a stable internet connection

Rationale: This to minimize the requirement for users to be able to play the game and maximize the number of users that will be able to join.

Fit Criterion: Even people with low end devices such as chrome will be able to join and play.

Acceptance Tests: Gameplay loop

10d Release Requirements

U30# - Continuous release cycle

Description: The game will take the approach of many modern online games and have a continuous release cycle with "seasons" separating each major release.

Rationale: Giving the game perpetual updates and distinguished by unique "seasons" will keep the user base interested in the game and allow for the product to be continuously improved.

Fit Criterion: Gameplay loop

Acceptance Tests: There should be a new gameplay season for each season of the year. Although these seasons don't need to align with actual seasons, there should be at least four of them.

11 Cultural and Political Requirements

11a Cultural Requirements

U31# - Careful symbol usage

Description: Symbols need to be carefully chosen to avoid confusion

Rationale: When publishing the game for an international audience, we need to choose symbols wisely so that the message conveyed by them makes sense regardless of culture.

Fit Criterion: Symbols must have a generally agreed upon understanding. We can find which symbols work by looking at which symbols are used by other major websites.

Acceptance Tests: User experience

11b Political Requirements

ID# - Subscription reminders

Description: In order to increase revenue, we will include payment reminders in order to remind F2P (free to play) users of the advantages of a P2W account.

Rationale: Interaction reminders like these have been shown to increase revenue, and will make management happy.

Fit Criterion: The landing page must contain at least one P2W reminder

Acceptance Tests: List ID# and/or names here . . .

12 Legal Requirements

12a Compliance Requirements

#1 - GDPR Compliance

Description: Our game must be compliant with the GDPR (General Data Protection Regulation) for users within the European union.

Rationale: This will open up the EU as a market, increasing the pool of potential players greatly.

Fit Criterion: To fit this requirement, we must follow all guidelines specified within the GDPR guidelines document.

Acceptance Tests: Accessibility

12b Standards Requirements

There are no particular standards the product needs to adhere to.

13 Requirements Acceptance Tests

13a Requirements – Test Correspondence Summary

Tests	Requirements																		
	F0	F1	F2	F3	F4	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15
Account handling	X	X				X														
Gameplay loop			X	X	X		X	X	X		X		X			X				
Data persistence	X	X						X		X		X		X			X		X	X
User expierence													X	X	X	X				
Accessibility																		X	X	X

	U15	U17	U18	U19	U20	U21	U22	U23	U24	U25	U26	U27	U28	U29	U30
Account handling	X														
Gameplay loop		X		X		X		X			X		X		
Data persistence			X		X		X		X			X		X	X
User expierence								X	X	X	X				
Accessibility					X	X									X

13b Acceptance Test Descriptions

T1 # - Account Handling

Description: This test handles all of the functionality surrounding account creation and maintenance. This includes creating an account, updating account information, and moderation duties like the banning of accounts.

T2 # - Gameplay loop

Description: This test handles all of the functionality surrounding the core gameplay loop of Pac-Rats. This includes setting up a game, playing a game, and completing a game.

T3 # - Data persistence

Description: This test handles all of the functionality surrounding data persistence. This includes data persistence between sessions, database integrity, and database maintenance.

T4 # - User experience

Description: This test regards the user experience of the game. This includes what users think of the game's branding/styling, whether they are able to find what they are looking for within the GUI, how intuitive the game is, etc.

T5 # - Accessibility

Description: This test handles all of the accessibility features/requirements of our game. This includes testing the color blindness options, the hardware requirements, environmental requirements, and so on.

III Design

1 Design Goals

SV: Identify the important design goals that are to be optimized in the proposed design.

Our design goals of this product is to have a fast rob

2 Current System Design

*SV: **IF** the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

N/A - There is no pre-existing system at the time of writing this document

3 Proposed System Design

This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However don't overlook finite state, activity, communication, or other diagram types as needed for effective communication.

3a Initial System Analysis and Class Identification

SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.

Your text goes here . . .

3b Dynamic Modelling of Use-Cases

SV: Insert sequence diagrams of (at least the most important) use-cases, as a means of identifying other needed classes.

Your text goes here . . .

3c Proposed System Architecture

SV: Identify the Software Architecture to be applied to this project, such as Client-Server, Repository, MVC, etc., along with justification for the choice.

Your text goes here . . .

3d Initial Subsystem Decomposition

SV: A slightly more detailed class diagram, showing the classes identified in sections 24a, 24b, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.

Your text goes here . . .

4 Additional Design Considerations

SV: The sections listed here do not need to be presented in the order given, and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.

4a Hardware / Software Mapping

SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.

4b Persistent Data Management

SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.

Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section ***Error! Reference source not found.*** *to avoid undue repetition, while reviewing only the most relevant items here.*

To store persistent data we will be using a relational database, specifically sqlite. It will contain user information such as username, password, email, etc. The database will be accessible and modifiable by game admins. Other persistent information stored might include reports generated, and other possible user confidentials.

4c Access Control and Security

SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.

Your text goes here . . .

4d Global Software Control

SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.

Your text goes here . . .

4e Boundary Conditions

SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown (normal or abnormal), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.

Your text goes here . . .

4f User Interface

SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.

Your text goes here . . .

4g Application of Design Patterns

SV: Any design patterns applied as a result of previous sections should have been addressed there, and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. (If any.)

Your text goes here . . .

5 Final System Design

SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing all the different subsystems and the (important) classes contained within each one. Still not a lot of internal details.

Your text goes here . . .

6 Object Design

This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces (public method signatures and responsibilities) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or by (Java) packages if those are designed. It may also be appropriate to

address additional design pattern considerations here, but not to the point of being redundant of previous documentation.

Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.

6a Packages

SV: If the design involves assigning classes to packages (.e.g Java packages), then the packages to be created should be documented here.

Your text goes here . . .

6b Subsystem I

Your text goes here . . .

6c Subsystem II

Your text goes here . . .

6d etc.

Your text goes here . . .

IV Project Issues

1 Open Issues

SV: Issues that have been raised and do not yet have a conclusion.

Since this game is currently browser based there won't be an offline mode. Users with no no stable internet connection will not be able to play the game properly. Another issue we have is raising this game to be cross platformed. This game currently works for all the popular browsers (i.e microsoft edge, chrome, safari, firefox, etc) but it is still not released for mobile devices (ios, android).

2 Off-the-Shelf Solutions

SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing (parts of) the new solution. The distinction between sections 35 a, b, and c is subtle, and not very important.

Your text goes here . . .

2a Ready-Made Products

SV: Products available for purchase that could be used either as part of a solution or instead of (a part of) a solution.

For the design of our avatars (rats and foxes) we will be using the animation software blender.

2b Reusable Components

SV: Similar to 35a, but for components such as libraries or toolkits instead of fully blown products.

Since this is a web based game, we will be using node.js and libraries such as ...

1. express to help routing with the pages
2. sqlite3 to manage data stored in our relational database
3. passport and express-session to save the user session

2c Products That Can Be Copied

SV: Products that could legally be copied would typically be past projects developed by the same development group, provided there were no restrictions that would prevent their reuse.

This game is a variation of pac-man but with a twist. The pixelated art style and the digital sounds can be legally used. The other aspect of the game such as the new game modes, character design, ingame powerups is original.

3 New Problems

SV: The proposed new system certainly has its benefits, but it could also raise new problems. It is a good idea to identify any such potential problems early on, rather than being surprised by them later.

3a Effects on the Current Environment

SV: Could the new system have any adverse effects on the working environment, e.g. the way people do their jobs?

This system does not have any adverse effects on the environment. This product is designed so that every user can easily log in or log out wherever or whenever as along there is stable internet connection.

3b Effects on the Installed Systems

SV: Could the new system have any adverse effects on other hardware or software systems?

This system does not have any adverse effects on other hardware or software because it is lightweight. This is made so that every device including those such as chrome books can play. It does not require installation or much memory. The requirements are very minimalist.

3c Potential User Problems

SV: Could the new system have any adverse effects on the users of the software? Could users possibly have a negative response to the new system?

We work to create a system where it thrives in an environment that does not create any adverse effects.

3d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

SV: Are there any (physical) limitations in the expected environment that could inhibit the proposed product? (e.g. weather, electrical interference, radiation, lack of reliable power, etc.)

As one of the core functional requirements in this game is having a stable internet, there might be environmental factors that could inhibit the proposed product. Such can include blackouts due to thunderstorms, electrical interferences, etc. All which can hinder from having a stable internet connection to the web server.

3e Follow-Up Problems

SV: Basically any other possible problems that could occur.

Your text goes here . . .

4 Migration to the New Product

SV: This section only applies when there is an existing system that is being replaced by a new system, particularly when data must be preserved and possibly translated / reformatted. Otherwise just write "Not Applicable" under section 38 and remove sections 38a and 38b.

“Not Applicable”

4a Requirements for Migration to the New Product

SV: These are a list of requirements relevant to the migration procedures. For example a requirement that the two systems be run in parallel for a time until the client is satisfied with the new system and the users know how to use it.

Your text goes here . . .

4b Data That Has to Be Modified or Translated for the New System

*SV: This section specifically addresses **data** that must be preserved and/or translated / reformatted during the migration process.*

Your text goes here . . .

5 Risks

SV: Consideration of the potential risks that could cause the project to fail / underperform.

One potential risk is not attracting the right audience and generating enough revenue to keep the game running on the server. The game requires a customer support and developer team for which all are very expensive. The server also has to be on everyday and that can prove very costly.

Making the game free can also prove fatal. We can charge users for a fixed price and be able to predict revenue using statistics but we might lose out on more than half of the audience. Due to the spike of many free indie games many users do not want to spend any money on games. Making it free and relying solely on ingame purchases/ads however might not generate enough revenue to power the system.

6 Costs

SV: An estimate of what it will cost to complete this project. Think not only in terms of dollars, but also time, resources, lost opportunities, etc.

- Following the scrum methodology the game will be ready for the general public in about a year and beta testing in half a year.
    - This includes the time for getting sound effects for the avatars
    - The animation for the avatars
    - the website UI
- Typically the cost from a low end indie game to a high end game ranges from ($25,000-$250,000)

- For and like all businesses, this is a gamble. There might be many lost opportunities
    - quitting a job to work on this wb game.
    - investing in other products due to the competitive nature of indie game development.

7 Waiting Room

SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.

As stated previously this game is currently all browser based. This meant that this game is not cross-platform. We might make it downloadable for ios and android users in the future so everyone can play uniformly. In addition there might be an option to install the game and have players go through online mode when having a stable internet is not an option.

8 Ideas for Solutions

SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution. However they can pass along any ideas they have here as suggestions to the developers. For CS 440 this report includes system and object design, so this section would make suggestions for implementation and testing that would come after design, such as the use of a particular language, IDE, library, or other tools.

Your text goes here . . .

9 Project Retrospective

SV: At the conclusion of the (CS 440) project, reflect back on what worked well and what didn't, and how the process could be improved in the future.

There is no retrospective because this project is still nonexistent. I will push forth, however, that the project can be done properly using agile methodologies, scrum specifically. By having proper sprints and sprint review, a developing team can improve productivity and be able to upload a functional but incomplete product for the client to use.

V Glossary

SV: The glossary is a more complete and inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.

Your text goes here . . .

VI References / Bibliography

This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the "Insert Citation" and "Bibliography" buttons in the "Citations & Bibliography" section under the "References" tab of MS Word. Creating new citations will not update this list unless you click on it and select "Update Field". You may need to reset the style for this paragraph to "normal" after updating.

[1] Robertson and Robertson, Mastering the Requirements Process.

[2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.

[3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

[4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

VII Index

This section provides an index to the report. The sample below was generated using the "Mark Entry" and "Insert Index" items from the "Index" section on the "References" tab, and can be automatically updated by right clicking on the table below and selecting "Update Field". To remove marked entries from the document, toggle the display of hidden paragraph marks (the paragraph button on the "Home" tab), and remove the tags shown with XE in { curly braces. }