Team nr. 16

Kim Ström, 080898-3399, kcs1@hi.is (presenting)

Irma Leinerte, 270188-5599, irl3@hi.is

Vision and Scope document

1. Business requirements

1. Background

The value of the PC Gaming Market Industry is steadily rising. More people have access to computers and the Internet than ever before, and therefore we wanted to make a product that everyone in this market could potentially use.

2. Business opportunity

As stated earlier, the PC Gaming Market is growing and with that growth there is room to capitalize. No one game can hold the attention of the user forever. By creating an eye-catching and engaging game we can make sure to be the next product the user chooses to play.

The typical customer prefers quick and easy access to the game from anywhere. By using Javascript for the web browser we could make this a reality. An overwhelming majority of computer games allow player-to-player interaction. All these needs are covered by our game.

Modern technology allows us to include microtransactions in our game. This allows us to potentially monetize any user's game interaction.

3. Business objectives

The objective is for our game to attract long-lasting users that also spend a lot of time playing. The user is considered engaged if they spend at least 30 minutes playing as we assume that in 30 minutes you have not only understood the mechanics of the game but participated in the game-loop multiple times over. This is relevant to our business, seeing how we know that engaged users are more likely to spend money on future microtransactions.

4. Success metrics

We can keep track of the number of concurrent users and the time they spend playing. Later, when we start monetizing, we can measure the money spent by independent users.

5. Vision statement

Our game is for people who have free time and wish to play computer games. The game is a multiplayer pirate browser game based around an open 2D world with focus on exploration, and the game is also easily accessible anywhere. Unlike hardware-demanding and complicated computer games, our game is easy to play and runs on all standard hardware.

6. Business risks

The game market is vast and there are new games coming out every day. It is possible that less users than we hoped would choose to play our game. In order to prevent this, we must focus on marketing and attract as many users as possible from the beginning.

Browser games are not the most popular type of computer games. This doesn't necessarily mean that people have a dislike for browser games but rather that there are not enough good browser games available. So, by simply having relevant marketing for the right platforms and focusing on having a product of high quality we should be able to attract a large user base.

7. Business assumptions and dependencies

The assumptions are that people want to play browser games requiring a persistent Internet connection.

2. Scope and limitations

1. Major features

Explored in Use Cases.

2. Scope of initial release

The initial release should be able to run on all major browsers and support the game's main game-loop of finding treasure via diving, selling it, and buying things inside the in-game shop. It should also support side-activities such as player-to-player interaction via fighting, as well as drinking, and dying. This allows our users to make use of our product while being able to partake in the core features that we want to offer.

The focus of the initial release is to attract and engage a player base.

3. Scope of subsequent releases

In subsequent releases we can add in more ways to monetize the product. Adding an in-game shop that allows real currency to be used is a non-intrusive feature that won't disturb the current player base. Beyond that we can allow mobile users access to our game by developing an app.

The focus of the subsequent releases is to provide more content for the player to engage with in order to keep our current user base, as well as finding new avenues to monetize the interaction of the users.

4. Limitations and exclusions

In order to ensure game security, we are using an authoritative server, this means we will exclude the chances of playing without a persistent Internet connection.

3. Business context

1. Stakeholder profiles

The stakeholders can be any person or group interested in playing games. Other stakeholders would be potential hosts of our game's servers and/or services.

2. Project priorities

It is a simple product and the cost of production is low, so if we run into any issues down the road we could easily spend a bit more money on hiring extra help.

3. Deployment considerations

The users will access the game for free from their browsers, so there is no need for downloading or installing anything. In order to get the game into the hands of the users we would make use of game hosting services such as itch.io and advertise our game on social media platforms.

Use Case document

Use Case Name

Find treasure

Primary Actor

User

Preconditions

The player must have a treasure map and a boat.

Success Guarantee

The user should have treasure in their inventory

Main Success Scenario

The user sails with their boat to the place marked by the treasure map. The user engages in a dive for the treasure. The server hides the user model from the world space, and displays the diving mini game to the user. The user manages to avoid the obstacles within the game while also finishing the game on time. The server displays the world space to the player, returns the player model to the world, as well as adds the found treasure to the player's inventory.

Alternate Scenarios

The user could potentially run out of time in the mini game, causing death. The server would then clear the user's inventory, remove all existing player items in the world, display the world space to the player, and finally transport the player's model back to the world's starting point.

The user could engage in a dive at the wrong place. The server would hide the player model from the world space for a given time. The user is unable to interact with the model and game world currently. The server returns the player model to the world.

Open Issues

-

Use Case Name

Buying items the player character

Primary Actor

User

Preconditions

The player must have entered the store. The player must have enough gold to pay for their desired product.

Success Guarantee

The player's new items are stored in their inventory.

The player's gold amount is updated

Main Success Scenario

The shop system displays different items and their prices. The player explores the items and chooses their desired product. The system deducts the price of the item from the player's current gold amount as well as adds the desired product to the player's inventory.

Alternate Scenarios

The player does not find an item they want and instead exits the shop system. The shop system shuts down.

The player buys an item which is not stored in their inventory, but one that exists in the real world. The shop system still deducts the gold, but does not store anything in the player's inventory. Instead the shop system updates and/or spawns in the world object.

The player decides to sell an item from their inventory. The system removes the item from the player's inventory and adds the item's gold value to the player's gold amount.

Open Issues

-

BRIEF USE CASES

1. The user approaches another user and starts a fight. Whenever user attacks, the system chooses a random number that is then subtracted from the other player's health bar. The fight ends with one of the two players dying. The system transfers the dead character's items to the one who lived.

2. A user presses the drink button. The system checks if the user has a drink in their inventory, upon a successful check it removes the item from the user's inventory, and then adds to the player's drink value.
3. The user arrives at the login screen and enters their desired player name. The system makes sure the name is unique, and if it is it displays the game world and instantiates the player model.
4. The user walks to their boat and presses the 'enter boat' key. The system transports the player model onto the boat and sets the player's input to control only the boat.