**Funding Issues (Nikolaj)**

The indie game industry has experienced some rapid growth over the past few years. So have the various funding opportunities for independent game designers. So-called crowdfunding has enabled independent game designers to upload their game prototypes along with and detailed description of the finished product online. Depending on what crowdsurfing portal is used, users of the portal can then donate money directly to the game developers, or up-vote the game, which in turn increases the chances that the project will receive funding from a larger-scale game development company (Valve is a good example here).

Currently, there are two major crowdfunding portals - kickstarter and steam greenlight. While kickstarter relies on user donations only, steam greenlight is based on up-voting. Game developers upload game prototypes and concepts for users to promote. If a project gathers enough popularity valve provides funding. Steam greenlight requires a $100 fee per project submission. Kickstarter is free but requires a working prototype of the project. Kickstarter also lets the developers determine the exact amount of money they want to raise. Through Steam Greenlight the company of Valve only provides “sufficient” funding. Both Steam Greenlight and Kickstarter have already proved to be successful funding sources, as various games have already been successfully funded via these portals.

However, Steam Greenlight and Kickstarter are very competitive. Especially on Kickstarter, many of the projects are being developed by major companies (Bioware, Lionhead Studios). It is hard to compete for the money of users with companies that have thousands of professional, experienced developers at their expense. The advantage that ROOMS has over these large-scale games is that the development costs are much lower. Steam Greenlight is more focused on supporting small, independent game-development teams. Realistically, it is more likely to receive funding through Steam Greenlight than Kickstarter. Additionally, once a game is funded through Steam Greenlight, it will also automatically appear on Steam, which eases the process of commercialization and distribution of ROOMS.

Two more alternatives can be taken into consideration: indie game contests, as well as funding through purchasable ingame content. There are various game contests that award monetary prizes as well as increase publicity, which, in turn, can attract the attention of potential publishers that might agree to fund further game development. Once again, Valve is very involved and has funded and supported indie developers (the game Portal is a well known example).

Lastly, releasing a prototype of the game on a free-to-play basis but including additional, purchasable in-game content is a very common practice (League of Legend, as well as various browser games are prominent examples). The game concept of ROOMS is heavily based around collecting in-game items - adding purchasable in-game items, therefore, is a promising funding option we will most certainly consider.

**Commercialization Issues (Thom)**

As a computer game, ROOMS has some advantages in terms of commercialization compared to other software products. Unlike web-based applications and services, most computer games have historically been commercial products, so asking for monetary compensation in exchange for the game is standard. Unfortunately, ROOMS cannot adopt the standard model for two reasons, ROOMS platform, the web, has not been traditionally been one where the users are expected to pay for the content, and secondly, ROOMS is largely user-content driven, and so a barrier to entry will reduce the amount of content present in the game, and thus the value that ROOMS provides to its customers. In this essay, I discuss the potential options available to ROOM, before concluding with what is, in the author’s opinion, the best of these options, and the business issues inherent in implementing it.

The traditional model of video game commercialization, where users pay for a game license, after which point they have infinite access to the game is an obvious choice. However, as mentioned above, this is problematic for ROOMS, due to it’s primary platform and due to the need for user created content. Additionally, ROOMS is a multiplayer online game, and as such, costs money to run. We must provide game servers, website hosting, and several other recurring fees that make this an unsustainable choice, even in the hypothetical best case scenario, where we achieve market saturation. However, this would be the simplest and cheapest of the described schemes to implement, as we would only need to verify that a user has paid for a game before allowing them to connect to our servers. If this option were chosen, it would likely be beneficial to provide potential users with a ‘free trial’, however that doesn’t provide any significant benefits, nor would it likely be particularly costly to implement.

The next most obvious model is the subscription model, traditional with massively multiplayer online games (MMOs). In this model, users pay a fixed amount periodically in exchange for access to ROOMS servers. This solves the sustainability issue, as we would now have money to pay for our own recurring hosting fees and servers, however it leaves the content issue. This leaves us with the issue of the platform, as selling a subscription to a web service may seem foreign to users, and of content creation (preliminary research shows that while subscription-based games like Blizzard’s World of Warcraft allow for user ‘mods’ and customizations, users do not typically engage in the practice, and the bulk of content is created by the game developers). ROOMS may be able to avoid the first issue, as our multi-platform architecture could allow us, with moderate cost and effort, to transition to a primarily desktop-based environment for the ROOMS game itself, instead of the current primarily web-based environment. This would be more costly to implement, especially if we transitioned to a desktop game, and would incur additional costs as we would likely need to provide more content to justify the subscription rate, especially if users do not provide a great deal of high quality content on their own.

Finally, I will describe what I believe to be the best possible option for commercialization of ROOMS, which comes in three parts:

First, advertising on the web has proven itself to be an effective strategy to generating revenue for a great number of web-based companies, however some users are offended by advertisements, however by allowing users to opt-out of advertisements for a small fee, we may be able to keep many of these users while still generating revenue from them. Services like Google AdWords make this option, for all intents and purposes, trivial to implement.

Second, we could allow users to play ROOMS on the web for free, but charge for ROOMS on other platforms, which would allow us to leverage our cross platform architecture. Users who own multiple devices would likely need to pay for a copy on each device, but this could be dealt with if there is user backlash.

Third and finally, ROOMS could allow users to buy, for real money some form of “in-game credit” (e.g. item-building materials) which then could be used to instantiate particularly rare artifacts. Additionally, this would provide a solution to the problem of users using the custom content creator to create items which are overly powerful. Recall that as users an item, a meter is to show its rarity, which would be a function of it’s power/usefulness. We could then use this power to calculate a price (in “credits”) for the item. If the user does not want to wait for the item to appear on its own (which may never happen depending on the rarity of the item), they can purchase it immediately for their character using credits. Allowing users to “sell back” items they collect which they no longer want for credits may prove to stimulate the in-game economy in a positive way, but this would require experimentation and testing. This is by far the most difficult to implement of the given options, as it would require deep integration into the game to be successful, however has the potential to be very popular, and very lucrative.

The combined difficulty and cost of implementing all three of these would likely be greater than that of the subscription model or the one time payment model, however it has none of the drawbacks associated with those models, as it actively encourages content creation, has no up-front barrier to entry, and, moreover, is a good fit for ROOMS, and, moreover, it doesn’t force us to compromise on our original design goals.

**Intellectual Property Issues (Nhat)**

Our senior design project is a game which in its current state has a very thin storyline, and therefore a small amount of concepts that could possibly be patented. However, moving into the future, the issue of who “owns” the game becomes very realistic if the game's audience were to grow in popularity. Not only should we patent the artistic aspects, but also the technical concepts which we adapted to our game. Patenting in the gaming industry is almost essential, especially for companies who publish Triple-A titles, such as EA Games or Blizzard Entertainment. They are also important to company who provide gaming technologies, such as Epic Games or Crytek, both of which publish gaming engines.

The key to understanding what we can and can't patent boils down to one fact: “there is no prohibition on patenting business methods”. (1) This means that while we can't patent game content, we can patent concepts such as the way we deliver such content. Our game is written from the ground up, not using any proprietary software, so we have very little to worry about in terms of violating software licenses. In our project, the language we're using is HaXe, which uses variants of the GNU license as well as the BSD license for it's various components. (2) This means we can distribute our game as long as we follow the requires of distribution, like including the original copyright notices.

If we could develop the game to a point where the lore is significant, then we should think about copyrighting or trademarking significant characters and story lines. Companies such as Blizzard have whole universes of lore which drive their main titles like Warcraft, Starcraft, or Diablo. Recently, the THQ studios went bankrupt, most of its value was in the assets that they owned. For example, the “Homefront” IP was sold for $544,218. Their “Company of Heroes” and “Warhammer 40,000” IPs went for $26.6 million along with the studio. (3) Although the company itself who owned the property is going bankrupt, other gaming companies acknowledge that their IP is still valuable and profitable in the right hands. This shows how important it is for companies to be able to have ownership over valuable ideas. As for the actual technical components that we could copyright, the actual engine we used to support our custom monsters, items, and rooms. The idea of creating and engine that can allow for custom content with custom AI is not new, so the portions that are patentable or copyrightable would be the way we implemented it, or novel game play aspects.

To address the issue of proper attribution to the university, since this is a school project, we must assess what university resources we used to aid our final product. In our case, nothing of significant importance was borrowed from the university to assist us in developing our prototype. I assume that we will not be using any university resources in our final product as well. Since all work is done on our personal computers, and none there are no licensing fees which the school covers, then no credit to the university should be necessary. Also, and more importantly, we should consider whether or not the instructor should be recognized as a contributor to the project. Since the results of the class is the product, I believe that the instructor has a fair share in the decisions made to the product if it were to become important enough to consider filing patents and copyrights. In all, the Intellectual Property concern is a very realistic one which affects any project that grows to a significant size.

**Ethical/Legal Issues (Mevludin)**

When wanting to play the ROOMS game, it is required for a user be registered through our website before being able to play the game. The registration process requires basic user information such as name, username, password, email and etc. Although this information may not compare to medical record information in terms of privacy, it is still private information that should not be shared with other users, unless the user does so themselves intentionally.

Assuming that there is a user that plays the game, when this user logs into the website they will be able to access all of their information. This includes their in-game statistics, personal information, in-game friends, the custom created content (rooms, items, and monsters), and other related content. So this allows the user to access all of the user’s information with just a single sign on. Most of the information will not be editable, and that information is the information from the game, such as friends, in-game statistics and etc. The editable information will be the information the user creates through the website, this is user’s basic information (excluding username) and the custom generated content. Although the website does not directly allow the user to change the user’s in-game information, it does allow the user to launch the game through the website. This would then allow the user to edit their characters, friends, and all other in-game data. Since the user login information grants the user the ability to do many tasks, this information needs to be well secured and hidden from other users.

Securing the information has become easier now-a-days and there exist many methods to do so. The first thing to do when building the game is to secure the users password and not make it so easy to read. This is done through encryption. In our game, this can be achieved with known functions such as MD5 and SHA1. Although these functions provide a level of encryption, there exist rainbow tables which are massive tables of encrypted password and allow “hackers” to look up encrypted password and get the corresponding password.(4) So the users will have to create uncommon passwords so it is not in the rainbow table. To insure this, the website will have to force the user to use symbols, numbers, capitalizations, and a minimum character count when creating their password.

There becomes a bigger issue with keeping the information secure when it comes to multiplayer games, and ROOMS is a multiplayer game. “Cheating, virtual frauds, and other security attacks are becoming increasingly widespread in the virtual world. “(5) To prevent malicious users from getting other users information, the data sent from the game to the server, or vice versa, should not contain the users’ username or other private information. The next to do is make the users know about the threats of online frauds, and scammers. One way of doing this would be to let users know not to share their personal information online and that no administrator would never require such information from the users. This would let the user know not to share their information when they encounter a person trying to lure them into giving their login information.

Another issue that the developers need to keep in mind when dealing with an online game is the server host. Since this is a start-up game, there will be no need for an expensive server host. So when choosing the host, the game developers need to make sure the server host is safe and secure, and will not violate users information for their own needs, such as advertisement.

**Payment Issues (Ashley)**

Running a website involves several unavoidable costs. Register a domain, pay for hosting, time to maintain the system. Naturally, a business needs money to support these costs. Today, the app market has developed many interesting ways to induce the flow of income. The one most relevant and interesting to our application is the in-app-purchase. The key philosophy is to allow users of your application to purchase upgraded content or perks that can be used within your application. ROOMS has several opportunities to capitalize using the in-app-purchase strategy.

A large feature of ROOMS is the ability for users to create custom content and upload it to the game. While playing, users will be able to collect blueprints & building materials for content creation. They can then use these materials to create new weapons, artifacts, and items that they will then be able to use (and which can be found by other users). While all materials will be collectible within the game, users will be able to spend credits to purchase rare or additional building materials to use for their creations. This plan offers the chance of a recurring source of income from the power users of our system.

Another opportunity is to create a perk system. These perks will offer users different benefits within the game. For example, perks could offer users immunity to poison or to keep all their items when they die. These perks will be available for one time purchase.

To support these features, we will need to be able to handle payment. This introduces a variety of very difficult security problems. To write our own system to handle payments we would have to be able to process credit card numbers and personal information. Even if we were able to build a bulletproof payment service, our customers would still feel uncomfortable with putting their credit card information into our system because our users trust us to provide them with entertainment, not keep their information secure. For these reasons, we would definitely use paypal and the cost associated with it would be a worthy one in our case. Using paypal simplifies our system and offers security & trust to both us and our potential customers.

While paypal is a good option for purchases made through a browser, it will likely be cumbersome if we incorporated it into our mobile applications. We have already decided that the content creation will only be available in the website version, however, we may be able to include the perk purchase in our mobile app. If we do decide to make in-app-purchases available in the mobile version, we will likely take advantage of tools such as the apple app store to handle these purchases for us.

Making our users comfortable is our main goal in choosing these external systems to handle our payment interactions is. We want them to feel comfortable that their transactions are secure and that their information is safe. For that reason, paypal is certainly our best choice for web transactions. Users are used to it and have grown to trust it’s security. Likewise, users trust their native app store’s transactions. They have grown to expect that a transaction made in-app will be made through their app store. Thus, it is impractical for us to waste our efforts in designing and developing an alternate payment system. Doing so would only offer us added difficulty and our users would feel reluctant to use it.

In conclusion, for the reasons explained above, we plan to solve the payment issue by using an external service. This service will be determined by user desirability and will be different based on platform. We will use Paypal in the website and the native app store on mobile devices.

**REFERENCES**

NHAT

(1) Bilski v. Kappos, 561 U.S. (2010)

(2) http://haxe.org/doc/license

(3)<http://www.forbes.com/sites/danielnyegriffiths/2013/01/24/the-break-up-bankrupt-thq-assets-sold-at-auction/>

MEVLUDIN

(4) <http://www.techrepublic.com/blog/australia/securing-passwords-with-blowfish/1274>

(5) <http://uwcisa.uwaterloo.ca/Biblio2/Topic/ACC626%20Security%20Issues%20in%20Massively%20Multiplayer%20Online%20Games%20X%20Jiang.pdf>