Post-Mortem

Steve Cossell

Contents

Additions to Design Diary

The design diary leading up to playtesting can be found here: SteveCossell/Assignment2/Design_Diary

After the playtesting we tabulated the problems that were found with the game and discussed solutions to these problems (see SteveCossell/Assignment2/Reflections for the complete table). From those James made some changes to the game on the Tuesday of week 12. They are outlined below.

Problems With the Swimmers

Problem From Playtesting Reflection	Fix
There was confusion and dislike of the swimmers returning to the water after being saved.	The allowed movement area was changed so that the swimmers could only walk on the beach area if they had been saved, otherwise they would stay in the water and swim.
It was sometimes unclear if a swimmer was drowning, what state of drowning they were in. If a person knew of these states then it was found to be further hard to tell at a glance	The swimmers did have speech bubbles during playtesting to help overcome this problem, but the speed bubbles would not remain visible for the course of the stage of drowning the swimmer was in. The swimmers now have a permanent speech

of the screen what state the swimmers were in. That is, the player had to look closely at the swimmer to tell how much trouble they were in.	bubble when they are in the first stage of drowning and a flashing speech bubble when in the final stage before death. We also had a speech bubble in the game that let the player know when a swimmer died, but this did not always show during the playtesting due to a bug. That bug has since been fixed.

Problems with the Lifeguard

Problem From Playtesting Reflection	Fix
One of the stand out problems with the lifeguard during testing was that the lifeguards spawned in the same location.	The spawning location of each of the lifeguards was moved towards the side of the center to match the controls. That is, the player using the left hand configuration on the keyboard had their lifesaver spawn on the left of the screen.
Not so much from the playtesting, but an aesthetic feature noticed afterwards...	The lifeguard image was made to look less blocky when walking on the beach.

Problems with the wildlife

Problem From Playtesting Reflection	Fix
Some players complained during playtesting that they sometimes could never get away from a jellyfish sting because the jellyfish would keep coming back and stinging the lifeguard before they got a chance to move away.	The jellyfish movement had another mechanic added where the jellyfish now moves away from the player for 5 seconds if they stung the lifeguard.
In reference to the above problem and other people complaining about the helplessness of being stung by a jelly fish we concluded that the sting might be too long.	We reduced the stun time to 2 seconds. This change will require further testing to see if it makes the game too easy.
An idea that was thought to be working before the playtesting session was that sharks would stop for a little while after they had eaten someone to symbolise the shark not being as aggressive when their appetite had been fulfilled.	The piece of code that was meant to carry out this action was never called due to our misinterpretation of the group collide sprite function in pygame. This problem has since been fixed.

Other Game Features

Problem From Playtesting Reflection	Fix
People either didn't notice or didn't thing gaining energy after saving a swimmer was worth while.	We removed regaining energy after successfully saving a swimmer.
Many people suggested during playtesting that the energy bar should gradually replenish over time. This was actually considered by our group during the design stage but was never implemented.	An energy replenishing over time feature was added to the game.
People hinted that the amount of sprint was too much and that they'd rather have more in a shorter period of time so that it truly became a strategic choice to use it.	We reduced the amount of sprint that is available to the player to run out in around 2 seconds. As a result of this we figured that the value of the energy bar was irrelevant so that was also removed from the interface.
One the main points raised was the meaning of the level timer when people were destined to drown after a certain amount of time anyway.	We decided to remove the timer from the game.

The Game

What worked and what didn't work with the game can be easily seen in the Design Report Reflections page found here: SteveCossell/Assignment2/Reflections In addition to this page I'll discuss some other points about the game which were not already mentioned in the Reflections page.

The Aussie and Beach Themes

I think we made a good attempt at putting across the Aussie beach theme by using sharks and jellyfish as wildlife, dressing the lifeguards like Australian lifeguards and having meat pies symbolise health. Given this, there were still responses from the playtesting of "How do we know we're on an Australian beach and not just playing a Baywatch game?" To bring the game closer to having a Aussie beach feel, ambient beach sounds can enhance the beach fantasy and a verbal ambient of people shouting things usually heard at the beach in Aussie accents would confirm the Aussie-ness of the game. Other visual additions could be random seagulls flying across the screen, surfers riding the waves, people playing beach cricket and British tourists getting sun burnt. Including the planned physical influence of waves on objects in the water could have further enhanced the beach feel and added a new mechanic to the gameplay.

The Game Mechanics

I think that the speed/energy bar was a good idea in theory. We initially included it to add an extra mechanic to the game and gave a replenish of energy to the player when they save a swimmer to give some positive feedback on completing a good task. We found through playtesting that the amount of boost the speed bar gave was too stretched out, so we've fixed it to run out quicker and provide more boost.

The repetitive nature of pulling drowning swimmers out of the water may seem boring because of the repetition, but this is how it is for a lifesaver. We could not remove this aspect from the game because it s a core mechanic to the lifesaving theme, but we did find ways of improving it. One major example that will be looked into is the use of power ups. With the improvements made after playtesting i think the interaction with sharks and jellyfish now also works nicely in the game.

The Point Scoring System

During development we thought that the scoring system added a nice way to let the player know when they're doing the right thing. However, after playtesting we realised that the score was rarely noticed. I believe that the scoring did not work in the context of the class playtest because people were playing the game because they had to. If the game were put in the context of a working facebook game then I believe the points system would have a greater influence. Therefore, although it was not evident during playtesting, I think that the points system works.

Interface

After some fine tuning after the playtesting session I think the controls of the game work. The problem was with the person on the left having a weirdly placed sprint button. This button has been changed to a similar location as that of the other player, or whom we received no complaints about the positioning of the sprint button.

The graphics were also very fitting for the game in terms of the theme and game play. The only improvement would be to animate the graphics to further add to the theme and gameplay.

The Design Process

Brainstorming

The Process

Our brainstorming activities actually started before the lecture where we found out that the game had to be Australian. One main idea we were running with was a laser game similar to Khet. (We didn't know about Khet at this stage and only found out such a game existed later in session).

When we found out that the game had to be Australian then we completely forgot about the initial idea. The main Australian theme related brainstorming session would have been during the tutorial where we asked to think of things that are

Australian and then later how things could be combined into activities. A few of the ideas we came up with during that class were:

- Footy players in ugg boots going on a pub crawl.
- Swimming/diving with sea creatures under the sea on The Great Barrier Reef.
- Having a BBQ with beer and something either about cooking the food for everyone or keeping the ants and blowflies away from the food.
- Something to do with Gold Rush times and managing land and gold to get the best profit.

We continued the brainstorming of a final idea a few days later and came up with three ideas that we ended up comparing and voting on. These three ideas were:

- A 2D platform game involving footy players (in ugg boots) going on a pub crawl and having to avoid rowdy fans and people who want to give them trouble. Beer and pies would be the energy and there would be some goal of reaching the next pub or getting home safely.
- An turn based strategy game in the gold rush theme were you had to manage your land and resources and acquire these throughout the game.
- A top down RPG-ish type game where you explored islands in The Great Barrier Reef. The theme was that giant creatures had kidnapped decent hard working Australians and you had to explore the world, fight the minions, beat the bosses and save them.

We looked at our liking of the theme of each idea (pub crawl vs. gold rush vs. lifesaving), our liking of the type of game it was (2D platform vs. TBS vs. RPG) and how easy we thought it was to implement. We then voted on each category, added the votes for each category and decided from this that we would pitch the life saving game with some of the elements from the other games that we thought would help put across the Australian theme.

The pitch then went how it went.

We then went back and scaled down the actions of the game without losing the general lifesaving theme and decided on our final idea which we developed during session.

Reflections

The main outcome I got from this was that our game idea, although I didn't think was that bad, was too ambitious for the time frame of the course. Therefore, I don't actually believe there was much we did wrong in this stage of design other than not consider the scope of the game in the context of the required time period, which is more a general software design issue than specifically something to do with game design.

I also think that our ambitious ideas sprouted from some of our initial intentions for choosing the course and suddenly not being about to make the FPS we always wanted to make. There may also have been influence from an over confidence in

our programming abilities and the fact that we wanted to be Australian and give everyone a fair go. Letting everyone have a say lead to too many ideas being included and not many being culled. I suggested the voting system to try and cull ideas by hiding people's dislike for others ideas in numbers.

Lessons Learnt

In terms of game design, this section opened my eyes to the process of thinking of an idea. I am definitely one of those people who is not very creative given an open ended choice, so a restriction of the choice and the systematic method of thinking of nouns, then verbs to fit the theme really helped me. Now I can take away another skill that will help a very systematic and logical me make creative and open ended decisions more easily.

As a software engineer I have the software development life cycle hard coded into my head and as software engineers we are usually set some guidelines on what the piece of software specifically has to do. I have now added something to my understanding of this model that applies to gaming software specifically. The brainstorming stage , which is rarely present in normal well specified software projects, should occur after the traditional "Requirements Elicitation" and before the "Design" stage of the life cycle.

In terms of general software design as opposed to game design specifically, I learnt the importance of understanding your scope and incorporating it into your design decisions. This is an important lesson to learn before entering the industry (of any type of computing) so that you don't promise a piece of software you can't deliver. In this stage, this would be the main thing I would to differently next time.

Design and Prototyping

The Process

This stage of design was done iteratively but was lacking proper user feedback as outlined in the reflections part of this section. The exact details of what we did at this stage can be seen in the design diary under: SteveCossell/Assignment2/Design Diary so I won't re-write them here.

Reflections

The way we designed the game mechanics and programmed them, I think, followed a well structured process. We followed an iterative process involving: deciding what the game needed, programming it, then evaluating how the change improved or worsened the game. I believe the first two of the three steps in this cycle were executed well. We would always discuss what the game needed next then code the modifications as a group so that we could all provide input at once.

The third step, the evaluation of the modification, in hindsight, was limited. We would discuss the modification and whether we thought it solved the problem and if it worked, but we were limited by our own views as the game developers. The evaluation step should involve other people as we learnt later in the session when

doing game play testing. That is, when you bake a cake, or at least when I've tried in the past, I think it tastes really good but everyone else complains about it being too sweet or burnt. The one main exception to this would be when we got bored of testing the program because the repetitive retrieval of drowning swimmers didn't change in any way. We realised this and made some modifications to the game halfway through the process that helped make the game more interesting.

Purely from a software development point of view I think the use of pygame was a good implementation decision. As a hardcore C++ programmer (because I was forced to be for my degree... James knows what I mean) I enjoyed the "n00bness" of using python, to put it bluntly. It fit well with the Rapid Application Development style of software engineering. We also found that working literally in the same cubicle in the lab increased productivity by allowing us to work on different things but discuss any ideas together immediately. An great example of this close proximity style of teamwork would be: "Viv can we get a copy of the lifeguard with different colour shorts for the second player" (when we'd just created the second player). Ten minutes later it was loaded into the game. I think this was one of the things that really worked during our prototype construction.

There were also little things that we did well during the coding stage which made integration of parts easier. One great example of this was creating stub images of blobs and having pygame load these in early so that when Viv had some images ready for export into the game we would just copy the images into the images folder on top of the stubs. Another small thing at the beginning of implementation when we were getting used to the pygame libs...one or two of us would be programming something, while another person would look up how to use a library function that would be required by the programmer in a few minutes time.

We also found ourselves volunteering for tasks that were seen as our speciality. The obvious example of this was Viv doing all of the graphical design so that our game didn't look like a screen of bouncing dots (I was sad to see the dots that I'd worked so creatively on be replaced by better graphics). As a result of my physics degree I volunteered to work out the maths involved in reflection of objects from the side walls of the window, directional vectors between two objects and the motions of the waves. David, with his psychology background, and Viv, with her HCI (lol) background lead the design of the playtesting script and questionnaire. Me, James and David would also contribute significantly to the coding of the game.

One small thing that the others might not even be concerned with was the fact that as designated lead programmer I chose to work on the code when I knew I'd be productive (midnight onwards). Maybe that doesn't work for others, so that may be a fault on my behalf more that a group fault.

In summary, I think our teamwork, design and implementation was excellent. The only let down in this stage of the process would be the lack of external feedback on the game's effectiveness in being a fun Australian game.

Lessons Learnt

It's obvious now, but at the time we didn't understand the importance of playtesting

until the playtest. The main realisation from this section is the importance of the play testing stage being included in the iterative process.

To be honest, in the lecture on playtesting the week before the playtesting actually happened, I saw the playtesting included in the design process as just another "testing" stage in any prototyping development life cycle. The difference, and I guess the lesson learnt, was that this course actually provided a practical situation where you got feedback on your playtesting. Other subjects mention that it's important to include the user in the design loop to make sure they are important, but I've never experienced what it's like first hand to have testing come back on a piece of software and require improvements. This was maybe not a specific game design lesson learnt, but an important one to realise the true importance of user testing early and often in the development.

Also, not really a lesson learnt, but something I should think more about would be programming when everyone is usually awake and not just some of us. That is more of a project management lesson, rather than a specific game design issue, but is still important.

Playtesting and Evaluation

The Process

This stage involved having others test our game and then fill out a questionnaire. We also took the guidance from the playtesting lecture and created a Logger class, which was hooked into everything and logged significant events during play. In addition to these two we also took observational notes during play. To help with this we asked the players to try and vocalise what they are thinking.

Reflections

I think this stage was executed successfully in that we were able to collect evidence of the success of our game and analyse where the game needed improvement.

The questionnaire was a good tool in working out what the players were feeling and what they thought about aspects of the game. Our questionnaire had a wide range of questions about the different aspects of the game to give us ample feedback. This was useful in finding out what people liked and didn't like, and what parts of the game people felt fun or annoying.

The Logger class gave us information on what the players were actually doing as opposed to what they were feeling. This was also a good tool used to let us know the strategies people used to play the game and how often they used features of the game. One stand out example, was that the game statistics revealed that barely anyone made use of the flip ability.

As outlined in the updated version of the design diary we have since looked at the problems testers had with parts of our game and we have implemented changes to match the results.

Lessons Learnt

As described in the Design and Prototype stage of game design, the main lesson learnt here would be the importance of running a playtesting session regularly throughout the development.

Not as much a lesson, but a newly acquired skill would be now knowing the tools used to run a successful playtesting session and how to use those tools.

Oh, and don't tease your playtesters or they'll flip you off.

Comments

I forgot where, but I remember Steve you mentioned somewhere stating my comment about "the game has no difference from original baywatch". Since I couldn't remember where, so I'll just leave my feedback here...

- I don't know if you have been to Surfer's Paradise in Gold Coast? Remember the sign of "surfer's paradise" at the entrance of the beach? I think that uniqueness will give people feeling of Aussie (given that they have been to there).
- Or even more, add random kangaroos jumping left to right or vice versa on the beach. This doesn't really affect the game, but adds in a feeling to it.
- Add a tree on the beach, with a koala on it. And probably some movements of it moving up and down.
- My own experience yesterday, I saw a humpback whale off the sea shore not far away at Rose Bay. I don't think you could spot a whale in any other beach in the world. So might as well add something like that in as well.

Hope this helps. Cheers. Oscar

From David

I think if we added whales, people would straight away think of japan :). Anyway a koala added subtly in a tree is a nice idea.

Thanks for the feedback and good luck with assignment 3

SteveCossell/Assignment2/Post mortem (last edited 2008-06-08 17:29:28 by SteveCossell)