

Game Mechanics & Game Play Programming



Wizards

In Concrete Boots



Assignment 2 Report

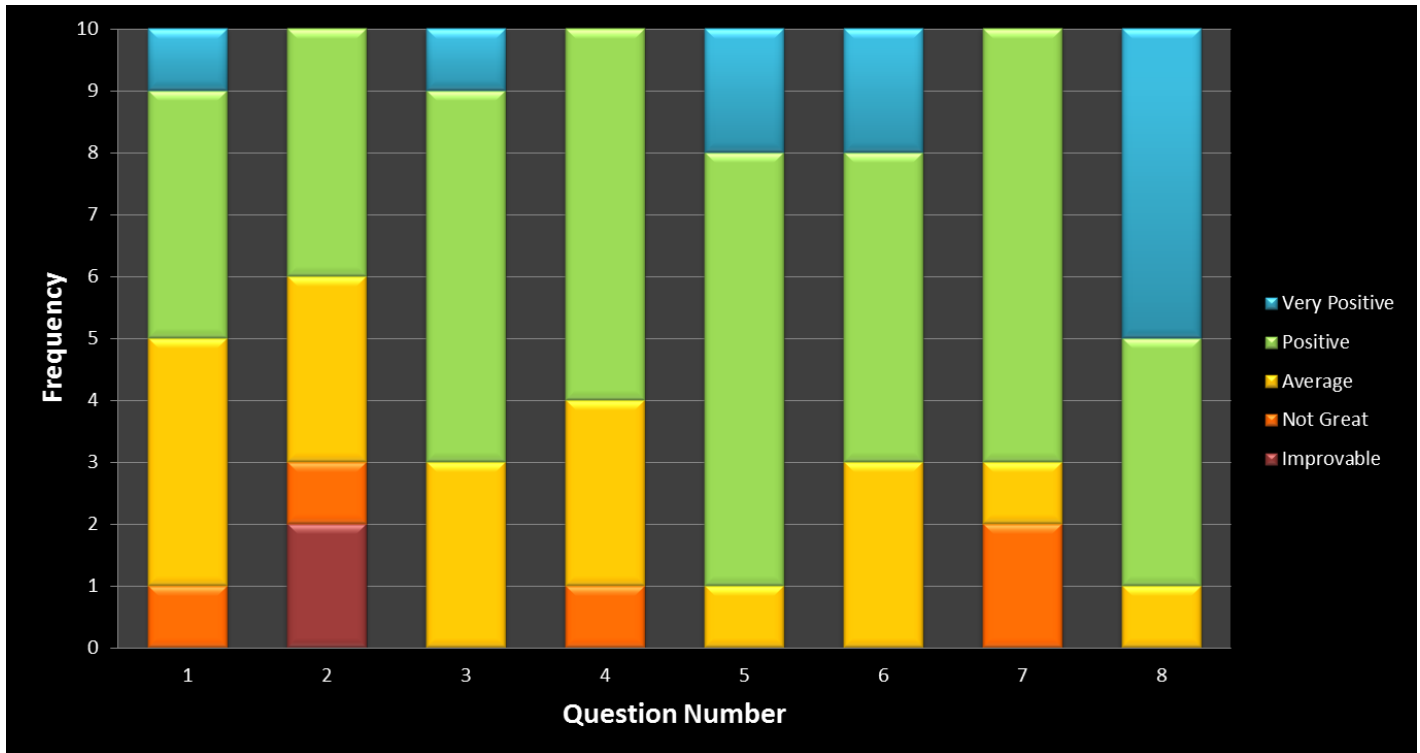
By

The Aero Tanks

Tom Harris	3236050
James McCormick	3229518
Alon Gal	3216299
Clark Lavery	3238021
Kyhil Duggan	3199447

1. User Study

Raw survey data is available in a complimentary PDF file.



1.1 Introduction

The graph indicates generally positive results to the questions asked. It is very pleasing to see that “Positive” was the most common.

There is however a few topics where we received feedback of “Not Great”, these areas will need to be looked at in more detail.

1.2 Question By Question Analysis and Resolution

Q1 - Do the difficulties of the gestures scale well between the 3 levels?

Large majority of responses were positive, the users were happy with the difficulty curve of different levels of spells.

Q2 - Are the gestures between the different classes equivalent in difficulty?

3/10 Users gave a negative response to this question and 0 users gave the most positive response. Below are two examples of the textual responses we captured for this this question.

“Fire gestures, seem easiest, simply because lateral and vertical movements seem easiest. Smooth curves are not overly difficult, where angled strokes are surprisingly difficult.”

“I think it's pretty alright, the water ones seem a bit harder to copy.”

These responses are very useful. Although 3 users felt that the gestures were not well balanced between the classes, they each felt that a **different class was advantaged**. This is a great result; each user felt a different class to be “the best”, implying that the gestures are in fact well balanced.

This supports our class mechanic, as we expected, the users had unique opinions on which gesture style is easier, meaning that they in future will chose their favourite class, based on their own experience and ideas.

Q3 - Are the time limits reasonable?

Large majority of responses were positive, the users were happy with time limits imposed upon them. Although we did receive a negative textual response, "time limits should be more restrictive".

Our Resolution: We plan to further test and possibly reduce the amount of time given to players to complete their gestures.

Q4 – Is the menu and card selection intuitive?

We didn't receive any very positive feedback for this question, and in fact received a negative response, and a negative textual response. ☹

"Whos player turn it is?"

Players were slightly confused by class selection screen and also had trouble keeping track of whose turn it is.

Our Resolution: Our game requires much better user interfaces. Our UI needs to be more intuitive and provide more feedback to the user as to the state of the game.

Q5 –Was the game enjoyable?

Pleasingly we received no negative feedback to this question; this is a great indication that our game has the potential to be a very enjoyable experience for players.

Q6 – Was the 2-player experience more fun, given the defending and choosing cards that the other player would want?

Large majority of responses were positive, the users were happy with the competitive side of the game. As an observation, we noticed that players really enjoyed the "Attack then Defending" mechanic, and the rush of competing head to head with their friends.

Q7 – Did the class system and class based spells add value to the game?

8/10 responses to this question were positive. The two negative responses however indicate that maybe the class system isn't having a large enough impact on the game.

Our Resolution: Consider ways to allow the Classes to have a larger impact on the gameplay, or make them more distinguished from each other.

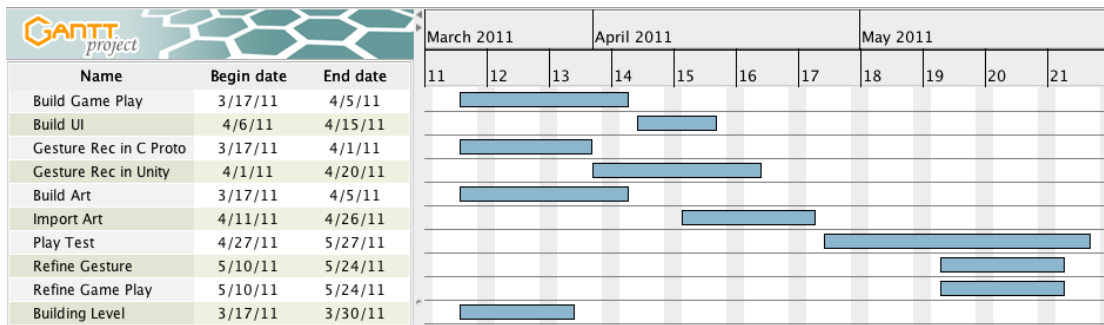
Q8 – Are the cards understandable, to read what they are and how they function?

This question received all positive responses, players seemed to have little trouble interpreting the information we were presenting to them on the spell cards.

1.3 Summary of Study

The results show that we need to work more on our UI, further test and balance the time limits given to players to draw gestures and consider ways to enhance the involvement of the Class system in the game. Overall though the large amount of positive responses are pleasing, and imply that our game is well balanced and very enjoyable.

2. Our Progress



Looking at our Gantt chart from A1, we are well on schedule, as planned we have completed implementing the game play, built a UI, implemented Gesture recognition and built and imported game assets.

As scheduled we are currently play testing, refining gesture recognition and refining game play. We expect to have all promised features completed to a high standard for our final submission.

3. Final Game Design – Amendments

Here are amendments to our design; all other features are as outlined in our last report which we have added as *Section 4* to this document.

Spell list

We have decided to add more spells to the game than we initially intended to have. Through play testing we realised that having a “Heal” gesture available adds suspense to the game, as players are sometimes able to “save” themselves if they possess a heal card in their deck. This mechanic has enhanced the competitiveness of the game and keeps the result of a “duel” less predictable, thus increasing enjoyment.

There are multiple levels of heal spells available and similar to the attacking spells, vary in complexity and effectiveness proportionately.



Gesture Recognition

At the time of submitting the first assignment we were still deciding upon how we would implement our gesture recognition. We have now implemented and are refining our gesture recognition.

We decided on the following approach because it allows us to compare complex gestures, and gives us an accuracy value, rather than a discrete indication of yes or no.

We have tried 2 different methods of capturing the data, 4 different filters and 2 different methods for archiving a uniform number of points between the reference gesture and player drawn gesture.

We have found that one method of capturing the data was slightly better than the other and have chosen to include 2 of the 4 trailed filters to act on the data, this decision was based on the output each achieved individually and in combination.

1. Capture the gesture as a list of points
Mouse locations are saved in mouse space (not world space) as the player moves across the drawing board.
2. Filter the points based on their displacement
We remove clumps of points and replace them with a single point that represents their average to improve comparability.
3. Filter the points based on their significance
We calculate the delta angle between adjacent vectors, and remove points below a threshold that can be considered insignificant.
4. Explode the gesture
We then add points back to the gesture; they are placed between the significant points with even displacement from each other. This ensures that the gesture has the correct number of final points to be compared with our “reference gesture” and means the points within the gesture are meaningful and represent the gesture the user drew in an organised and comparable way.
5. Compare Gesture to Reference
Compare vectors between points to those in the reference gesture. From these comparisons an accuracy is determined.

The code that handles these tasks can be found in the *gestureFilter* script.

Ideas for gesture recognition

<http://www.bytearray.org/?p=91>

http://en.wikipedia.org/wiki/Levenshtein_distance

http://www.youtube.com/watch?v=j7U_0kHvfA8

<http://www.webcam-whiteboard.com/?c=44&b=18>

4. Original Game Design

See Section 3 for amendments

Play our Flash prototype for our prototype!

<http://raws.adc.rmit.edu.au/~s3238021/gmgpp>

Game Type:

Two human players will duel in a last man standing style battle. This will create a competitive environment for players to put their “Casting” skills to the test. The game will be viewed from the 1st person perspective while casting, although the camera will pan between players in a cinematic style between turns.

Time Granularity:

The game will be turn based, this allows for 2 players to share the one input device.

Level Design:

The field of play will feature two opposing towers. Each player will stay on their respective tower and engage the other player at range.

Spell Casting System/Combat Mechanic:

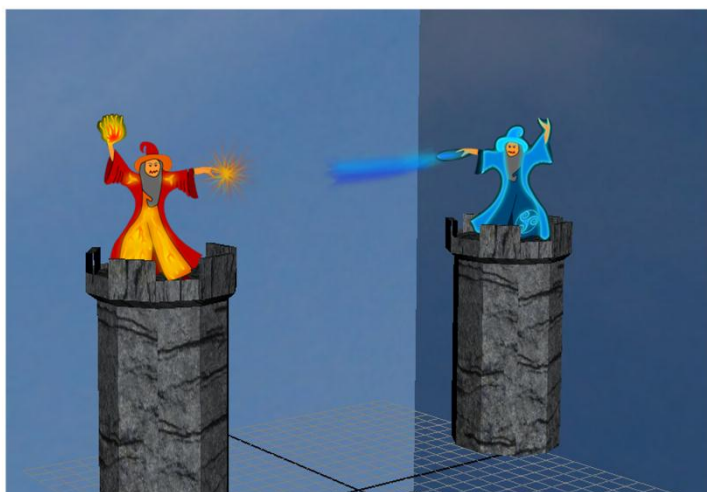
As the match begins 4 “Spell Cards” are assigned to each players hand.

The cards will be randomly chosen from a universal deck, introducing an element of chance to the game play.

Each card represents a different style of spell, outlining its basic stats and displaying the gesture required for performing it. Players are able to see the cards in their hand, although not those of their opponent.

The combat consists of an attack phase and defence phase.

Each phase has a time limit to ensure players are acting under pressure.



In a players attack phase they select a card from their hand.

Once a card has been selected players use the mouse to attempt to produce the spell outlined on the card.

Spells are cast from the 1st person perspective, displaying the gesture on the screen in real time as the player draws it.



Once the spell has been conjured its effectiveness is calculated. The effectiveness of a spell is dependant on three factors, the complexity the spell (gesture), the accuracy with which it is performed, and class of the player. (see *Class Mechanic*)

$$\text{Power} = \text{Complexity} \times \text{Accuracy Factor} \times \text{Class Bonus}$$

This system rewards players skilled “Casters” but also recognises that gestures have different complexities. The class modifier introduces a tactical element into the game play.

The card that was selected has now been used, and is removed from the attacking player’s hand. Two cards are randomly selected from the universal deck and shown to the attacking player. The player may select on of these cards to replace the one they have just used.

The attack phase for this turn is now complete and the camera switches to the perspective and the control of the defending player.

The defending player has a chance to defend against the incoming spell in the “defence phase”. The opposing player is presented with the gesture that they must perform in order to defend against the incoming attack. The effectiveness of the incoming attack may be reduced or even reflected, based on the accuracy with which this defence cast is performed.

If the defence spell is cast with poor accuracy the defender will receive the full effect of the incoming spell, if the defence spell is cast with great accuracy the effect of the incoming spell is eliminated, and if the defence spell is cast with amazing accuracy the spell is reflected back towards the other player.

Each time a spell is reflected, the amount of time given to the opposing player to perform the defence is reduced, this prevents players of an even skill level becoming stuck in a seemingly infinite loop of spell reflection.



It is then the “attacking” player that must defend in the same manor as the defending player. This is repeated, until one of the players performs a defence gesture inadequately and receives the effects of the spell.

The rolls of attacker and defender are then reversed. Damage is applied after both players have completed their attack phase; this is based on the idea that everything is happening at once.

Class Mechanic:

A class mechanic is used to personalise the player’s experience. Before a match begins each player selects one of three classes. Each class is based on an element: fire, lighting and ice.




Each class receives an effectiveness bonus for cards of their own element, although all classes are able to use any card.



Each element’s gestures follow common style, so that players may select a class based on the style of gestures they are most comfortable with.

Each class has 3 levels of complexity; gestures with a higher complexity have a higher effectiveness multiplier (see *Spell Casting System/Combat Mechanic*)

Element	Strength/Complexity Level		
	1	2	3
Fire (Square)			
Lighting (Triangle)			

Ice (Circle)			
--------------	---	---	---

Gesture Recognition:

A mouse will be used to collect the player’s input, this enables players to play and enjoy the game without special hardware.

Ideas for gesture recognition algorithms

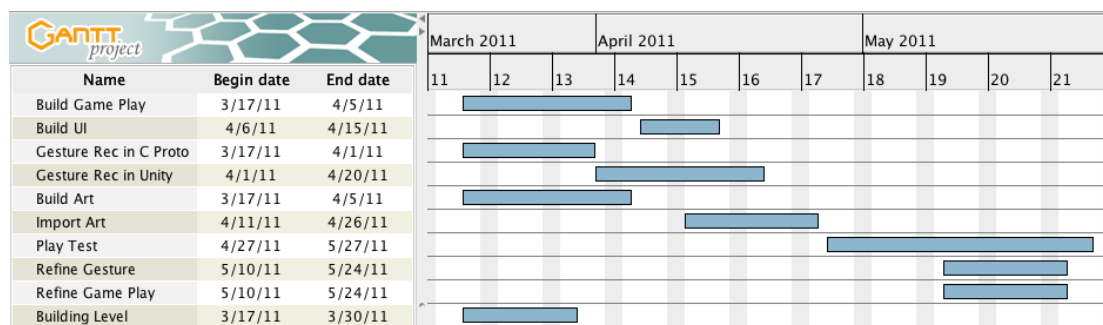
- <http://www.bytearray.org/?p=91>
- http://en.wikipedia.org/wiki/Levenshtein_distance
- http://www.youtube.com/watch?v=j7U_0kHvfA8
- <http://www.webcam-whiteboard.com/?c=44&b=18>

Feasibility Outlook:

Hardware:

The project should not suffer major hardware constraints as we have elected to use the mouse as the input device. This is a device that is readily available and has excellent software support.

Milestones:



References:

Magic the gathering – “Spell card system”
<http://www.wizards.com/Magic/Multiverse/>

Pokémon - “Combat System”
<http://www.pokemon.com/intro/>