## Project Details

**Project Title: Texas Hold’em Poker**

**Student name: Joseph Kellaway**

**Student number: 10503639**

## Summary

I intend to write a Texas Hold’em poker game using Visual Studio 2012, primarily in HTML/VBScript as that is the only language that I currently know. It will obviously only be able to run from within Internet Explorer because of the use of VBScript. Texas Hold’em involves several instances where cards are dealt either individually to each player or to a communal pool of cards with players taking turns to bet on what hand they currently have (or don’t have, as the case may be). I will need to include parameters that alter how the computer players bet and include some degree of variation between the way that the computer reacts to the same situation to make it having a good hand more subtle so that they player can’t learn how it reacts to having a specific hand. Depending on how quickly and easily I get through it I will add the option for a second human player (probably prompting the first player to turn away to hide your personal cards from each other). I would also like to add a time limit to each turn and a slight delay during the computer turns to allow the player to see what is occurring and simulate the way an actual poker game would play out against real players at a table. I can also increase blind values as the number of hands played goes up to simulate the way that poker tournaments are run. I am currently (very slowly) teaching myself C# in my spare time, I have considered the prospect of translating my game from VBScript into C# as a side project and use that language instead. Obviously I would seek approval before implementing the switch.

## Motivation

I have limited programming experience (one year at AS Level learning basic VBScript a considerable amount of time ago and Mark’s module) so I feel that clinging to the safety of VBScript is wise to ensure that I’m not trying to do the impossible, but pushing my knowledge of it further by making a game is a good step. I am a regular poker player (and member of the Poker Society) so I believe that the game will hold my continued interest and fuel my passion to make it as good as possible. My knowledge of the rules of poker (such as hand ranking) will aid me in testing to ensure that everything works as expected. I feel that the “bare bones” of the project will be a fairly easy challenge for me to complete within the timescale, but as I progress I will be able to push my goals further and implement more parts to the project (I intend to do everything listed above and possibly generate more ideas as I feel that there is never an excuse for doing a half-hearted job, but I don’t want to start with the goalposts way out of sight and feel overwhelmed). I expect to learn how to implement concepts that I haven’t yet done within VBScript – the computer player, adding a second player, having a currency that is divided correctly between those involved in the pot (if multiple players are “all in” then sometimes a split-pot occurs, where one or more players cannot win all of the chips). Implementing only using 5 of the 7 cards available to each player and the computer players assessing “draws” will be particularly rewarding (the idea of betting when you almost have a flush or straight, but need another card in the next round(s) and obviously having 2-3-4-5 is better than A-2-4-5 as the first has 8 cards that can be dealt to create a straight whereas the second only has 4).

## Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Tasks | Description | Deliverable | Priority |
| 1 |  | **Preparation** |  | Must |
|  | 1.1 | Plan the relevant tasks | Complete proposal template |  |
|  | 1.2 | Design the user interface | Sketch of planned UI |  |
|  |  |  |  |  |
| 2 |  | **Prototype** |  | Must |
|  | 2.1 | Code for dealing out a single poker hand from a pack of 52 cards and display to the user in pictures onscreen |  |  |
|  | 2.2 | Code to work out what hand has been dealt |  |  |
|  | 2.3 | Test no errors occur (especially choosing most valuable hand available) |  |  |
|  |  |  |  |  |
| 3 |  | **Full Hold’em** |  | Must |
|  | 3.1 | Expand to Deliver “whole cards” to prepare for numerous players |  |  |
|  | 3.2 | Expand to implement rounds (pre-flop, flop, turn, river – ready for betting) |  |  |
|  | 3.3 | Compare hands against each other every round to see who is winning at each point |  |  |
|  | 3.4 | Test – comparing hands correctly at each stage |  |  |
|  |  |  |  |  |
| 4 |  | **Currency/Betting** |  | Must |
|  | 4.1 | Generate “chips” for each player with option of betting in each turn for human player – check, bet ½ current pot, current pot value, double the pot, treble the pot or all in |  |  |
|  | 4.2 | Force each player to match the bet to stay in the game this hand. |  |  |
|  | 4.3 | Implement “blind bet” for 2 players each pre-flop |  |  |
|  | 4.4 | Implement moving dealer to change who plays “blind bet” |  |  |
|  | 4.5 | Code decision-making from computer players (whether to match a bet or not) |  |  |
|  | 4.6 | Test – ensure betting doesn’t have to occur, but if it does it must be matched, folded or raised |  |  |
|  |  |  |  |  |
| 5 |  | **Computer Betting** |  | Must |
|  | 5.1 | Code to allow computer player to assess the hand they currently have and make a reasonable bet based on it |  |  |
|  | 5.2 | Expand to cover each round and re-raises |  |  |
|  | 5.3 | Test to ensure computer acts as expected |  |  |
|  | 5.4 | Expand to implement intelligent bets (occasional bluffing and “limping-in” with a good hand to mix things up) |  |  |
|  | 5.5 | Add alerts to inform player of what computer players have done each round |  |  |
|  | 5.6 | Test |  |  |
|  |  |  |  |  |
| 6 |  | **Split Pot** |  | Must |
|  | 6.1 | Adjust betting code to ensure that a player can only win the amount of chips that their bet entitles them to |  |  |
|  | 6.2 | Test |  |  |
|  |  |  |  |  |
| 7 |  | **Predictive Betting** |  | Must |
|  | 7.1 | Computer can bet based on the likelihood of them getting a good hand (i.e. high cards pre-flop, flush/straight draws later on) |  |  |
|  | 7.2 | Test |  |  |
|  |  |  |  |  |
| 8 |  | **Levels** |  | Must |
|  | 8.1 | Implement a regular increase to the amount of each blind bet. |  |  |
|  | 8.2 | Implement increasing “ante” to be paid by every player each hand |  |  |
|  | 8.3 | Test |  |  |
|  |  |  |  |  |
| 9 |  | **User Interface** |  | Would |
|  | 9.1 | Find suitable background image so that game appears to be played at a poker table |  |  |
|  | 9.2 | Add computer player names and avatars |  |  |
|  | 9.3 | Request player name at start of game |  |  |
|  | 9.4 | Create a start menu – options to change background image and computer players names and avatars |  |  |
|  | 9.5 | Test |  | Must |
|  |  |  |  |  |
| 10 |  | **Second Human Player** |  | Would |
|  | 10.1 | Add to start menu – option for multiplayer mode allowing a human to take the place of one of the computer players |  |  |
|  | 10.2 | Separate turns with a pop-up message to prevent players from seeing each other’s hand |  |  |
|  | 10.3 | Test |  | Must |
|  |  |  |  |  |
| 11 |  | **Final Testing** |  | Should |
|  | 11.1 | Run through the test scenarios I have documented during development | List of features/conditions that are not working yet |  |
|  | 11.2 | Fix anything in list that doesn’t work |  |  |
|  | 11.3 | Get suggestions for improvement from other users | List of potential improvements |  |
|  |  |  |  |  |
| 12 |  | **Improvements** |  | Would |
|  | 12.1 | Assess feedback given to me in 11.3 into two groups: feasible and unfeasible (based on time left). Rank feasible in order of depth that they give to the game |  |  |
|  | 12.2 | Implement above list starting from top working through, ensuring to test thoroughly throughout – there is no point adding 3 new mechanics that don’t work properly, I’d rather add 1 that works perfectly or none at all. |  | Could |
|  |  |  |  |  |
| 13 |  | **Report** |  | Must |
|  | 13.1 | Write report | Word document with screen shots |  |
|  | 13.2 | Thoroughly check through report and final test of project before submitting project and report |  |  |
|  |  |  |  |  |

## Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks | Start Date | End Date | Milestone | Decision to be made |
| 1.1 and 1.2 | 14/01/2015 | 23/01/2015 | 1 | Yes/No from module staff |
| 2 | 24/01/2015 | 31/01/2015 | 2 | Working properly? |
| 3 | 01/02/2015 | 13/02/2015 | 3 | Working properly? |
| 4 | 14/02/2015 | 20/02/2015 | 4 | Working properly? |
| 5 | 22/02/2015 | 27/02/2015 | 5 | Working properly? |
| 6 and 7 | 28/02/2015 | 06/03/2015 | 6 | Working properly? |
| 8 | 07/03/2015 | 13/03/2015 | 7 | Working properly? |
| 9 | 14/03/2015 | 20/03/2015 | 8 | Working properly? |
| 10 | 21/03/2015 | 27/03/2015 | 9 | Working properly? |
| 11 | 28/03/2015 | 03/04/2015 | 10 | Working properly? |
| 12 | 04/04/2015 | 10/04/2015 | 11 | Working properly? |
| 13.1 | 11/04/2015 | 18/04/2015 | 12 | Proof-read? |
| 13.2 | 18/04/2015 | 24/04/2015 | 13 | Project ready to turn in? |

## Risk Analysis

|  |  |
| --- | --- |
| Risk | Contingencies |
| My computer could break down | Keep 3 USB Stick backups  Make use of free slots in Smeaton  Check extra SOFT051 tutorials for free slots for extra programming time |
| Something becomes too difficult | Contact tutors about what  Put extra hours in during that week to try to solve problems  Take a break – sometimes taking a step away from coding helps dramatically  Consider altering schedule  Very worst case scenario, consider tutorial & keep a note of it for report |
| Required to go home for x time | Assess whether it is entirely necessary  Take USB stick(s) to continue working from home |
| Unable to stick to schedule | Put extra hours in  Consider skipping non-essential parts (9, 10, 11.3 and 12) as last resort |
| Illness | Work from my bed  Put extra hours in when I’ve recovered |