**EECS 481 Software Engineering**

**Group Contract**

**Modes of Communication**

We will use phones to communicate about urgent things (such as missing a meeting or significant codebase issues). Each group member is individually responsible for getting their phone number to the other team members. Teja will create the group distribution list for email. All communication that is not of urgent nature will be sent through email. The maximum expected time for replies for email will be 24 hours and by text will be six hours.

The group will also be taking advantage of Git based revision control, syncing with a repo on GitHub.com. This will be discussed later, under the “Codebase Management…” heading.

**Meetings**

We will have weekly meetings on central campus at Shapiro Library on Saturdays 2 pm to 3 pm. The time and location are subject to change, however, but only by consent of the entire group. We will all do our work individually and the meetings will be for updating each other on what we’ve done and deciding what to work on next, as well as testing the code as a whole. It is ok for a member to miss a meeting occasionally as long as they let the entire group know beforehand. We will have a three strikes policy: if a group member gets three strikes, the team will have grounds to fire them. If a group member misses a group meeting without letting the group know or without a good reason, it counts as a strike.

Preparedness will be important for the weekly meetings to be productive, as we will be coordinating the next installment of project development and testing what has been completed. Because of this, group members are expected to take on an appropriate amount of work each week such that they can complete it on time. This work assignment will be determined at the meetings and will be decided by the individual group members for themselves (this process is detailed under “Personnel”).

If a group member chooses too little work or significantly more than they should have (and not completed it) for two meetings, it will count as a strike. Every meeting thereafter, if they do not complete their expected work it will be another strike. If a group member communicates that they will not be able to complete their work on time and asks for help, it may not count against the member. This stipulation is left vague intentionally, as if a group member consistently ends up in such a situation the group can still decide to penalize them.

**Personnel**

Everyone will have an equal standing in the group. At meetings the group will discuss how to best partition the project’s components and decide who works on what part. If there are two people who want to work on the same part of code, either they will find a way to compromise or we will do a coin flip. As mentioned above, we will have a three strikes system for gauging responsibility. If a group member doesn’t ask for help and does not finish their work in a reasonable amount of time, it will count as a strike.

The group will not track of the amount of time each person is contributing; instead we will keep weekly work assignments fairly equal in size and difficulty. This way everyone has, on average, the same workload. It also means that we can judge contribution by how much one completes rather than how long one works. Weekly workloads do not need to exactly match those of other group members, though. Instead, they can fluctuate for individual members. For example, a group member may choose to work more during one week to offset working less the next and vice versa (this is to facilitate interviews, other class work, etc…). Finally, if someone exceeds expectations for their work we will buy them food or remove a strike from their record (if the member has any).

**Codebase Management and Collaboration Techniques (Approach pt. I)**

Although weekly meetings and email communication will help the team members stay coordinated, revision control will be the cornerstone of the team’s coordination. Work will be assigned and tested at weekly meetings, but throughout the week group members will be developing on their own. To coordinate the changes being made, a private GitHub.com repo will be used. Each team member can use the Git application of their choice to sync with the repo, but they must use Visual Studio 2012 and the same libraries and language as the rest of the team. This information is given more detail below under the “Programming Focus Areas…” heading.

The use of Git means that the team’s work will be constantly merged and members will be made aware of conflicts early, rather than waiting until our weekly meeting to catch such flaws. This also means that every team member has access to the latest version of the code, allowing informed design decisions to be made independently and on the fly rather than at meetings.

Assisting other team members will also be done through Git. For example, if a team member needs help getting a function completed for the Kinect input, another team member can simply edit a file on their behalf, commit it, and sync the change. This simplifies what could otherwise be a mess of email attachments and potential confusion.

**Programming Focus Areas and Team Breakdown (Approach pt. II)**

The program will be written in C# due to its similarity to C++, the core language in the EECS program. C# also features useful abstractions and integration with the Kinect SDK and Open GL. Open GL will be used to assist the project team with graphics and physical interactions between game elements. The Kinect SDK will provide the team with basic libraries for user input though the Microsoft Kinect. There will be three main areas of focus: the game’s physics, Kinect based user tracking, and the game’s graphics and design.

Team members will break out into focus areas of their choice, with a developmental focus on Kinect input until it is functioning. From that point onward graphical design and physical modeling of ball behavior will be the focus. A team (or single group member, at times) will focus on level design and artistic design at all times. Variety and visual design quality will be the primary means of making the game captivating and cannot be neglected.

**Firing and Quitting Policies**

The quitting and firing policies for our group will follow the processes described in the class syllabus. If a group member feels that they are doing a disproportionate amount of work for the project, they can choose to quit the group. They will get the same grade for that portion of the project, and they have to let the rest of the group know in writing by email or hard copy.

If a team member gets three strikes, the rest of the team has the right to initiate the firing process. To start the process the team will email them with the work they need to do in order to stay in the group. They will have three days to complete that work and if they fail to do so, the team will let that person and Professor Chesney know in writing that they are fired from the group.

**Hiring Policy**

Because there may be other groups with members who quit or are fired, our group will retain the right to hire new members. New members will be hired in only by unanimous consent of the current group members. Interviews will be conducted, and the reason a potential hire left their group will be given careful consideration and made very clear to the current group members. To get an accurate picture, one or more of the current group members must interview a member of the group a potential hire left. Any information from the potential hire’s former group will be disclosed to them, and they will be given an opportunity to comment on it or defend themselves (if necessary).

If a new hire is brought in, they will be brought up to speed by collaborating with an assigned partner from the team. That partner will work closely with the new hire for a week, showing the hire what the group is working on and how our workflow functions. After than first week, the new hire will be asked to take an assignment at our weekly meeting and begin their own work.

**Signatures**

Teja Ravipati

Andrew Ashburn

Zhao Huang

Chris Jeakle

Alison Christiansen