# CS Games 2017


# Relay Programming I

| | |
|---|---|
| Participants | 3 |
| Workstations | 3 |
| Value | 7% |
| Duration | 3 hours |

# Software development simulator

You've just been hired by an Indie video game start-up in Silicon Valley. This studio's specialty is simulation games, and you've been assigned to their latest title: Software Development Simulator.

## Game concept

The player shall begin the game with a 100 000$ budget, on week 0, with no employees. The player may enlist resources, or choose an action, then advance by a week. After each week, the game shall compute the cost of all resources and subtract it from the budget. Progression of the project must also increase in proportion to the allocated resources. The goal of the game is to finish the project before being bankrupt, in fewer than 52 weeks.

## Victory condition

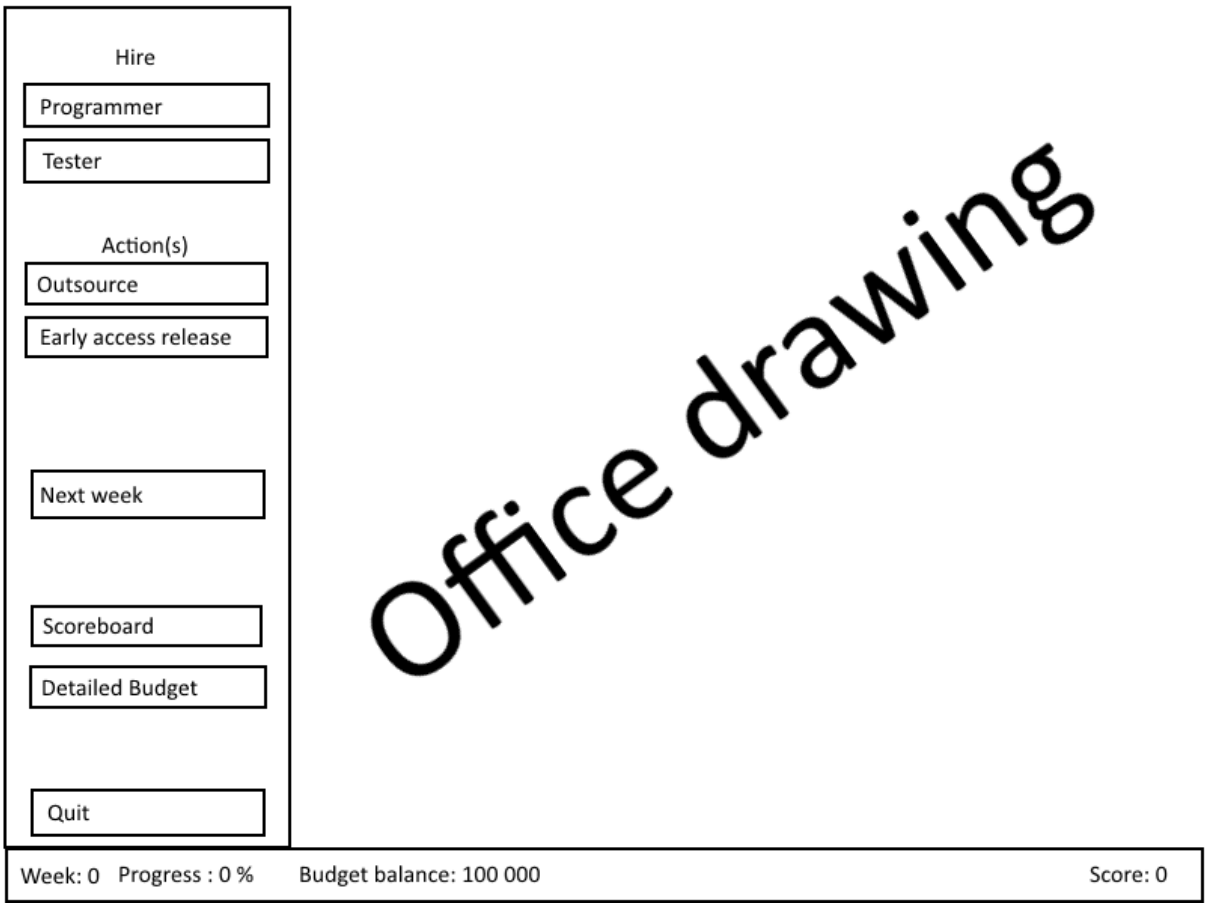
Victory is achieved when the project reaches 100% progress, in under 52 weeks, with a remaining budget equal or greater than 0.

## Defeat conditions

- Remaining budget becomes smaller than 0$.
- 52 weeks are elapsed before the project reaches 100% completion.

## User Interface prototype

User Interface design is already completed. You must use this prototype.




## Features

- ❏ Display the current week (week #). (1 pt)
- ❏ Display remaining budget. (1 pt)
- ❏ Display project completion next to the budget. (1 pt)
- ❏ Display "x" programmer(s) (desk images) in the office area. (3 pts)
- ❏ Display "x" tester(s) (desk images) in the office area. (3 pts)
- ❏ Display the "next week" button and implement progression to the next week (including budget and progression computation). (5 pts)
- ❏ Implement a screen displaying victory and defeat. (3 pts)
    - ❏ A new game starts automatically, without requiring software reset. (2 pts)

- ❏ Display a Hire Programmer button and implement its action. (3 pts)
    - A programmer costs 1 200$/week.
    - A programmer makes the project progress 1 point/week during the first 4 weeks.
    - After 4 weeks, he makes the project progress 1.4 point/week.
- ❏ Display a Hire Tester button and implement its action. (3 pts)
    - A tester costs 800$/week.
    - A tester makes the project progress 0.4 point/week during the first four weeks.
    - After 4 weeks, he makes the project progress 0.6 point/week.
- ❏ Display an outsourcing button and implement its action. (3 pts)
    - One time cost of 2 000$ for a one time random progression between 1 to 6 points.
    - May be used at most 3 times/game.
- ❏ Display an Early Access Release button and implement its action. (5 pts)
    - Instantly grants 15 000$ more budget.
    - Starting the following week, costs (weeks_since_early_access * 1 000$) / week in perpetuity to offset user frustration.
    - So, Activate Early Access on week #0: +15 000$, week #1: -1 000$, week #2: -2 000$, etc.
    - May only be used once.

- ❏ Implement a screen displaying a detailed view of the budget for the week, including the following elements: (5 pts):
    - Display current balance (budget remaining before this week's deductions)
    - Programmer costs
    - Tester costs
    - Outsourcing costs
    - Cost or revenue from Early Access Release
    - Balance at the end of the week

- ❏ Implement a deterministic scoring system for a game. (2 pts)
    - ❏ Display current score in the lower right corner. (1 pt)
    - ❏ Take the player's name at the end of a game and persist scores. (2 pts)
    - ❏ Add a button to display previous player names and their scores. (3 pts)
- ❏ Add a button to quit the game. (2 pts)
- ❏ Display the current project name. (1 pt)
    - ❏ Allow the player to modify the current project name. (2 pts)
- ❏ Display the company name in the office area (1 pt) (your choice, 1 bonus point if it's related to your team name)
- ❏ Implement a cheating system to allow faster progression. (4 pts)
    - Document your cheats in a readme file.
- ❏ Align programmer and tester images so that there is no space between desks. (3 pts)
- ❏ Include a “readme” file in your solution, document completed features and the execution of your software. (1 pt)
- ❏ Include a “run.sh” file to start your software. (1 pt)