<Executive Manager Sim>



[Overview](#_yj5nhqp5cf0j)

[Theme / Setting / Genre](#_5s48wntac2es)

[Core Gameplay Mechanics Brief](#_uzq23hfhdv6e)

[Targeted platforms](#_kvz0cxkhwt0s)

[Monetization model](#_421ijgnpyvmc)

[Project Scope](#_rdb2xo3rjh0s)

[- <Game Time Scale>](#_985ywpbk0q3f)

[- <Team Size>](#_ze1jo361gze)

[- <Licenses /Hardware / Other Costs>](#_w7dtj7hgn3h1)

[Influences (Brief)](#_155cm8v36jpc)

[- <Game Dev Tycoon>](#_c6nxu1rzd2cc)

[- <Football Manager series>](#_ssiemceczw16)

[The elevator Pitch](#_337xnergkz1b)

[Project Description](#_z7oe7x50rpf3)

[What sets this project apart?](#_s4h84uy3suza)

[Core Gameplay Mechanics (Detailed)](#_a8x4s87df6uk)

[<Employee System>](#_jyik8zbcjcio)

[<Agile System>](#_vno11ubdeuhc)

[<Production System>](#_xvyngoigro5v)

[<Reputation System>](#_kct9c2l3dr9p)

[<Turn System>](#_jh5z9w969ud)

[Story and Gameplay](#_6pmf08ssy6y0)

[Story (Brief)](#_v8lar3qa6jxl)

[Gameplay](#_22fr0nyjw3op)

[Assets Needed](#_6m1256af7s3j)

[- 2D](#_1wb69txjqarm)

[- Sound](#_f8xx8iwg5gs9)

[- Code (Nontechincal explanation)](#_ky1qxs88utre)

[Team Expenses](#_mcs2rc5hz47x)

[Success of a released game](#_f9h76fq5kc0i)

# Overview

## Theme / Setting / Genre

- Simulation/Strategy game

- Text based UI supported with 2D animations

## Core Gameplay Mechanics Brief

- <Employee System>

- <Agile System>

- <Production System>

- <Turn System>

- <Reputation System>

## Targeted platforms

- Windows

- Linux

## Monetization model

- Release on Steam with price tag between 5$-10$ according to market feedback

# Project Scope

## - <Game Time Scale>

- Cost: ~20k $

- Time Scale: 3 months

## - <Team Size>

- <Programmer>

-Monthly full time cost:2000$

-Unity scripting

- <Artist>

- Monthly full time cost:2000$

- All art design

- <Designer>

- Monthly full time cost:2000$

- GDD, project managing, occasional programming

## - <Licenses /Hardware / Other Costs>

-Unity (free)

-Steam greenlight license (~50$)

# Influences (Brief)

### **- <Game Dev Tycoon>**

### **- <Football Manager series>**

-The game has a UI like football manager games and usually cosists of menus, lists and statistics. Also much like a football manager game you handle teams of people, check their performance and satisfy your employers and customers.

# The elevator Pitch

In this game, players become a executive manager of a AAA game studio. They start with a relatively small company and form teams, hire/fire employees, handle deadlines, build a reputation and change studios. Game has a manager-games-like UI consists of menus, lists and statistics. Game is also turn based. Each turn fast forwards to a week later and lets player to handle upcoming weeks jobs.

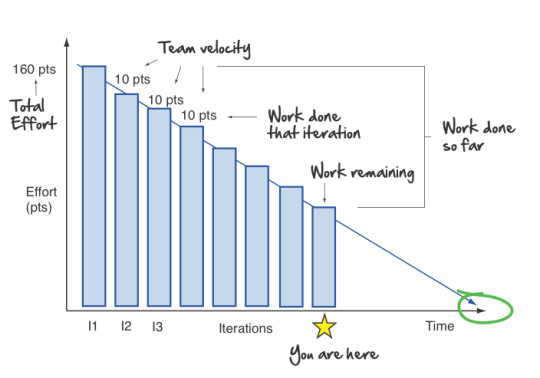
## 

# Project Description

In this turn based game executive manager simulation, the goal is to make it to the top game development studio and release a successful game there without going bankrupt. Game will have football manager like GUI, simple menus supported with satisfying effects and sounds.

Player will receive game project requests from their publisher which they funded by. Player should forms group of teams to make the given game by hiring team leaders and letting them build a team. Player decides type of the team and how big it will be and hiring employees will be made by the team leaders according to their skills. Type of the teams will affect the overall score of the game and therefore incoming revenue. These teams will produce neccessary story points each turn and when they reached predetermined target of storypoints of the project publisher will release the game.

Player will have tools to control what’s going on the company and assess the situtation. On each month graphs of burndown chart will be shown. These graphs will show the overall status of the game and each team’s performance. From these grahps player can see clearly the schedule, release date and if project can be released at that date or not.



Player must take actions like pushing teams, reforming teams, hiring/firing employees to make sure game can be out on the given date by the publisher.

On each release player’s reputation will be updated. If game is successful, reputation will rise and vice versa. When player reach at one of the predetermined reputation milestones, he/she can switch to a better studio with more funding and therefore bigger teams and projects.

If player fails to balance incomes and expenses, they will go bankrupt and game will over. They can try again from the latest game release (if there is any) or beginning of their career at current studio.

# 

## What sets this project apart?

- Macro management

- One of the few game development simulations

- Manager games feeling

# Core Gameplay Mechanics (Detailed)

## <Employee System>

- Player, can hire employees to lead the teams and to work for the teams. Player can hire new workers for a team from the relevant team menu by clicking “hire more employees” button and the related team leader will begin hiring. On the pop-up menu player can choose how many more workers he/she wants. Depending how qualified team leaders are, the employees they hire will be more qualified. Player can also command the team leader to “reform the team” so the related team will begin to replace low performance workers, or with the “fire employees” command team members will be reduced to the given number. Player should also hire team leaders manually and select from a list of applied potential team leaders. To do that player should create a new team. On a pop-up window he can see stats of applied people (min 3-max 5). Greater the reputation of the player, more qualified the applied CV’s will be. Without a leader that team cannot have any employees.

- Employees and the team they belong will be listed in “Teams” tab and related team tab beneath it. Player can see the teams’ overall stats, morale percantage, improvement status with a green increasing arrow or red decreasing arrow. Morale percantage allows workers work efficiently while improvement status indicates if these workers stats are likely to increase or decrease.

- Each employee has 3 stats. These are art, programming and design. These stats effect employees performance on the team they work in. Morale will also effect their performance and will increase as they get paid on time and games they release become successful.

- Player can command his/her team leader to “push” their employees harder which lower their morale but get things done faster, or go “easy” on them which keeps morale up but not much work will be done. There is also a balanced state which lowers morale in the long run and does decent work. Team leaders will be always on the chosen “state” and can be changed month to month.

## <Agile System>

* When player begins a game’s production, that game will require a specific amount of “storypoints” categorized same as the stats of employees (art, programming and design). Player must form the best choice of teams for the game considering these requirements, genre and target audience (detailed on <Production System>). Each month a report about production will be presented to the player showing how much work done and how more must be done with a “burndown chart”. Player can see clearly if this work pace is enough to release the game till the deadline and which teams are doing fine or struggling. With the light of these information player should take action to fix the issues by hiring/firing employees, pushing teams further or letting them relax, forming new teams, reforming/shutting down teams, delaying the release.

### 

## <Production System>

-After every new release of the game (or at the beginning of the game) a new game project request will come from the publisher.

-When player takes a job from their publisher, that game will have a randomly given genre, target audience and storypoint requirements. Player should form teams according to these information. These teams may include

* Programming Teams
  + AI
  + Physics
  + Sound
  + Scripting
  + Motion capture
  + Network
* Designer Teams
  + Puzzle Designer
  + Level Designer
  + Gameplay
  + Story
  + Narrative
* Art Teams
  + 3D
  + 2D
  + Experimental

Each main teams will produce storypoints related to them to complete the project. But sub-teams must support the genre and target audience for a profitable game. Also each target audience expects different features from different genres. Table below is explaining which genre and audience require what.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Req. Teams | Young | Adult | Everyone |
| Action | Gameplay, sound, level design, 2D or 3D art | 3D art | AI | 2D art |
| Adventure | Puzzle design, story, narrative, any art | Sound | Experimental art | 2D art |
| RPG | Gameplay, story, any art | Scripting | Narrative | 3D art |
| Simulation | Gameplay, physics, sound, 3D art | - | - | - |
| Strategy | AI, level design, any art | Gameplay | Puzzle designer | Network |
| Sport | AI, gameplay, physics, 3D art | Sound | Network | Motion capture |
| FPS | Gameplay, sound, level design, 3D art | Scripting | AI | Network |

These relations can be hinted to the player in a “guide book” in a subtle way. More right team choices made, more profitable the game will be. Therefore player’s reputation will be higher.

To form a team, the studio should be able to afford it and pay for it at the start. So a small budget studio can’t be able to access high-end tech and must grow larger.

Player can form as many teams as he/she like but it will cost and their focus on the game will be divided.

## 

## <Reputation System>

-Player will have a reputation and this reputation will affect quality of job applications, scope of the projects and ability to switch to more qualified game studios.

-Reputation points will be gained when a game is released. If game is not successful (if it does not reach the desired profit) reputation points will be negative.

-If player reforms teams or fires employees too often, his/her reputation will also decrease.

-When player reaches a predetermined reputation level a bigger company will make an offer to the player to work with them. Player can choose to accept or delay the offer. Offer will be repeated with a suitable version on every game release until player accepts it (if required reputation still stands).

## <Turn System>

-Each turn jumps to the next week. Each week team’s performance, morale and skills will be updated.

-Each month (4 turns) the burndown chart will be displayed, if a game is released and on the market at that moment its sales will be also displayed and budget will be updated.

-If budget goes down of zero, employees morale will go down enormously, publisher will get nervous and send messages to warn player. On the 3rd month of being at below zero, if player can’t release the game publisher will assess the situation. There is a chance that they will give player a bailout support according to the complation percent of the game. Closer the game to the release higher chance the bailout will be given. If a game is released when below zero, publisher will not give another project and player can only hope to recover with the upcoming revenue from the game. If bailout won’t be given and there is not a game to save the company, they will go bankrupt and game will be over. Player can start from the latest game release, beginning of their carrer on the current company or start over.

-Player will continue to rise and change businesses until they reach to the top studio. When player release a successful game in that top studio a congratulation message will appear and inform player that they beat the game. But if player wishes they can continue to play and more game projects will keep coming from the publisher. But there will be no more company switching.

# Story and Gameplay

## Story (Brief)

Player is a ambitious game designer made it to a small company’s executive manager position. And now his/her only goal is to work at the biggest game studio of the world.

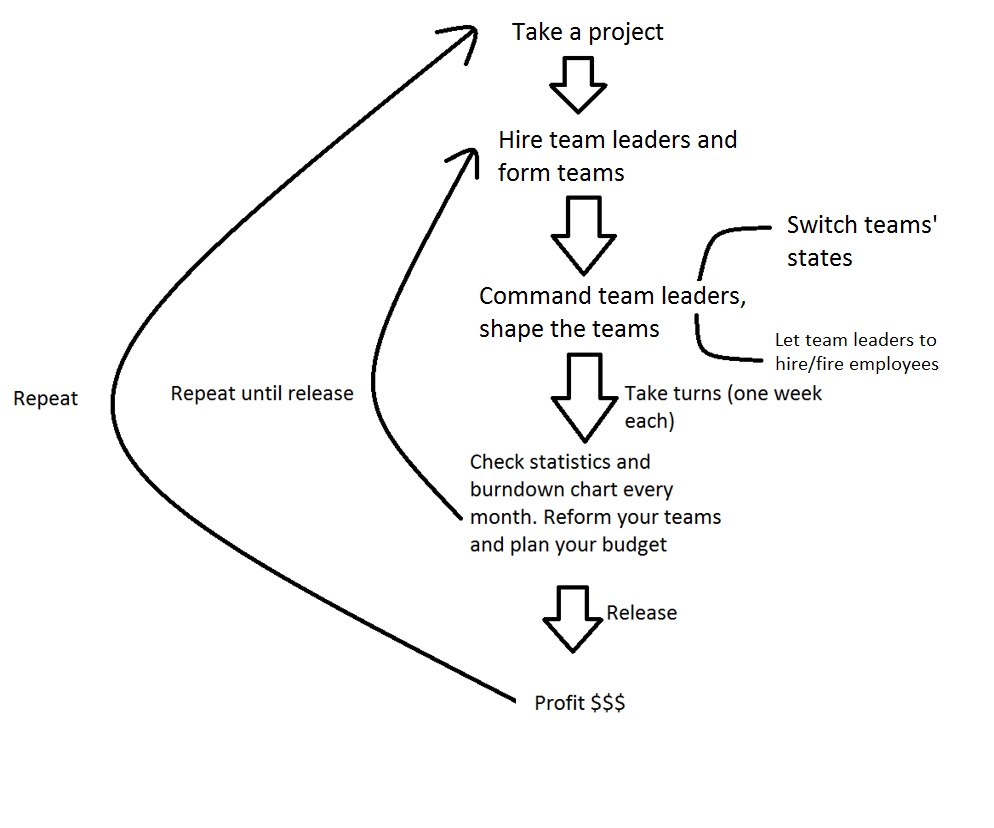
## Gameplay

Player will see menus of teams on the main screen. Can roam freely between tabs to see the teams or click a new tab button to form a new team. On each team menu there will be team leader’s name on the top and command buttons near it (commands reminder: hire employees/fire employees/reform the team, push them harder/balanced state/take it easy ). Leader’s stats can be seen in terms of programming, art and design. There is also overall morale of the team shown with an increasing or decreasing arrow. On the list below the leader’s row (which is bigger than other employees) there will be employee list. Player can see the names of the employees and stats but can’t interact with them. This will give player an idea about how qualified this team is.

When player wants to form a new team he needs to click the new tab button and choose a team from popup window including a predefined list. Player can see how much that team will cost to form and monthly costs to maintain it. After player chooses the team they must choose a team leader from another popup window with a list of applicants with random stats. These stats will affect the quality of the employees and must be related to the type of the team for efficiency (detailed on <employee system>).

When player clicks big “next week” button at the lower right corner of the screen teams will produce storypoints for the game, expenses and revenues will be calculated, employee stats and morale will be updated and date jumps to the next week. Every 4th turn player will see a pop-up window contains burndown chart. Player can’t interact with it, can only view and exit. Chart of the month can be viewed again from the tab “developing game” above. It also includes information about game like genre, target audience etc.

When game finishes “next week” button changes to “release the game” button and when player clicks it game will be released. As long as the game stays in the market it will produce additional income. After a release a new request from the publisher will pop up, informing player about the new game they will make. Player will follow this loop until he/she loses or wins the game.



Game loop

# 

# Assets Needed

## - 2D

- User friendly buttons

- Responsive GUI

- Pop-up windows

- Status icons

-Arrows (red decreasing,green increasing)

-Skill icons (programming,art,design)

- Empty burnwood charts to draw on

## - Sound

- Sound List

- Developing team related sounds to be used on related menus

- Keyboard typing sounds

- Murmuring

- Office sounds

- Sketching sounds

- Recording sounds

- GUI sounds

- Clicking

- Popup menu sounds

-Good news notification sound

-Bad news notification sound

- Incoming money sound (Ka-ching)

# 

# - Code (Nontechincal explanation)

## Team Expenses

Team expenses are determined by the employee number and salaries they take. Salaries are determined by the stats of that employee and stats are determined by the stats of that team leader who hired them.

When player commands a team leader to hire more employees, he can hire employees whose have stats min=(teamleader’s stat)/2, max=(teamleader’s stat). If team is a programming team, only programming stat of these employees will be shown and calculated, others are irrelevant.

While hiring an employee, algorithm will try its chances to hire the best quality of employee, if it fails it will roll dices for another.

If a team leader have a skillpoint of programming:8 and looking to hire programmers, algorithm will try to hire employee with programming:8 with a chance of

100/(teamleader’s stat-(teamleader’s stat/2))% which is 25%

“25%” is also our “base chance”. It is a random hit or miss. If it fails it will try to hire an employee with programming stat:previousstat-1 (which is 7) with a chance previouschance+basechance which is 50% and goes on until hits one. In a worst case scenario hired employee will have half of the team-leader’s stat. All variables are integers.

After team was hired, team’s expenses will be calculated. For each stat point, employees will cause 1000$ monthly expense included teamleaders.

|  |  |  |
| --- | --- | --- |
| Team leader | related stat:8 | salary:8000$ |
| employee1 | 4 | 4000 |
| employee2 | 6 | 6000 |
| employee3 | 5 | 5000 |
| employee4 | 6 | 6000 |
| employee5 | 5 | 5000 |
| employee6 | 7 | 7000 |
| employee7 | 8 | 8000 |
| employee8 | 6 | 6000 |
| employee9 | 7 | 7000 |
| employee10 | 5 | 5000 |
| employee11 | 6 | 6000 |
| employee12 | 5 | 5000 |
| employee13 | 7 | 7000 |
| employee14 | 6 | 6000 |
| employee15 | 8 | 8000 |
| employee16 | 5 | 5000 |
| employee17 | 6 | 6000 |
| employee18 | 7 | 7000 |
| total | 117 | 117,000 $ |

In the table above there is a example of a team with some employees. Their stats will determine how much storypoints they will create weekly and how much they will cost monthly. These information will be used as expenses and created storypoints while calculating incoming revenue.

## Success of a released game

Success of a game will depend on the type of the game, the teams that work on it and developing time. Each genre will have required teams to be successful (unknown to the player) (e.g. for action genre it is gameplay,sound and level designer teams). Also target audience will define additional team requirements for that game. For action games “young” audience will demand 3D, “adult” will demand good AI and “everyone” goes well with 2D graphics. But for every genre target audience will demand another feature. So if player makes an action games for adult audience they will need programmer teams for gameplay, sound and AI, designer teams for story. Player has to have one from each department so they will have a art team but because there are no restrictions about it so they can choose whatever they want. For this example there are 5 requirements (included art team). The number of met requirements will be the percentage of the game’s success.

Again with the action-adult example, lets say player formed gameplay,sound,story, AI and 3D teams. It meets only 4 requirements, therefore game’s success rate will be

(4\*(100/5))/100=80%

Story team is irrelevant and won’t calculated in the incoming revenue.

Since the example game is the 80% success, our incoming revenue will be

2x(80%x(expenses of gameplay, sound, AI and 3D teams))

This is main revenue. Polish bonus or delay penalty will be added for the final revenue.

If player can finish the game before the release date, game will release at the predefined date but until that date game will be polished. Teams will continue to create storypoints and the ratio between these will be the polish bonus rate.

polish bonus = main revenue\*(storypoints produced while polishing)/(storypoints produced while making the game).

So the final revenue is the main revenue+polish bonus.

If player misses the deadline, the ratio between the delayed weeks and predefined schedule will be delay penalty rate.

delay penalty= (main revenue without expenses in delayed weeks)\*((weeks it took after deadline)/(weeks from beginning to deadline))\*(-1).

So the final revenue is the main revenue+delay penalty.

