Board Game Project

At least 2 users: Player and Account Administrator

Goal of the game: Use resources to build objects or acquire victory points. Game ends when a certain number of objects are owned by a particular player. Player with most victory points at end win!

Certain objects are worth more victory points or give a special action to take ( ie players build a lumber yard that helps them get wood faster).

Players can lock out other players from gathering resources by taking that resource slot for the turn.

First player “token” passes after each turn

Vision:

Executive Summary:

Our goal is to provide an enjoyable game for 2-4 players where each gets a sense of responsibility in maintaining their resources to be able to efficiently build, and gather points to win. Each player will take turns doing one of two actions: Gathering resources or building structures. These structures are worth a certain number of points which are stored with them that at the end of the game are tallied to see who the winner is. Throughout the game a player may take certain actions that may either help themselves get points or to prevent an opponent from gaining points. The game reaches the end when any player has built their fifth structure. After all players have had the same amount of turns as this ending player, all scores are checked and a winner is determined.

To bring about this goal we will be designing a system using C++. The different players will be instances of a single player class object, while the resources and structures will be two different classes. The players will be able to check information at different times through the command line to see their progress in the game. All data stored in the system will be available to be seen by the administrator, who can help solve player disputes if necessary. Games are not saved between uses, so they must be finished within one session.

Positioning:

The player begins a game in the area of which the game takes place on a board.

Stakeholder Descriptions:

The chief stakeholder is the professor, Mr. Bettens. He will be the final say on all aspects of the project on whether or not they meet the requirements. He will point to us in the direction of which the game makes sense in the development process.

The lesser stakeholders are each of us in the project team. We are stakeholders because like in the real world one would make money off the product they produce, we will get a grade based off of what we have done on the project.

The Player user is a person trying to interact with the system to play our game. They take actions to play and have the goal of winning the game.

The System administrator is a person maintaining our game and the running program. They attempt to keep all players at a level playing field, make sure users are authenticated, keep the system up, and conclude the game when players are done.

Product Overview:

Our product is a C++ project built in Visual Studio that will allow the users to be city builders and resource gatherers. They gather appropriate resources that will be useful to them in building structures which score points. The player with the most points at the end of the game is the winner.

Summary Of Benefits:

N/A

Summary of System Features:

* New Player capture
* User ID authentication
* System Administration for users, engine, and code.

Cost, Pricing, Schedule:

The product must be through its first iteration by November fifth and through its second iteration by December fifth.

Assumptions and Dependencies:

* The user of software has access to internet, and electronic device such as a phone, tablet, laptop, or personal computer.

Iteration Plan:

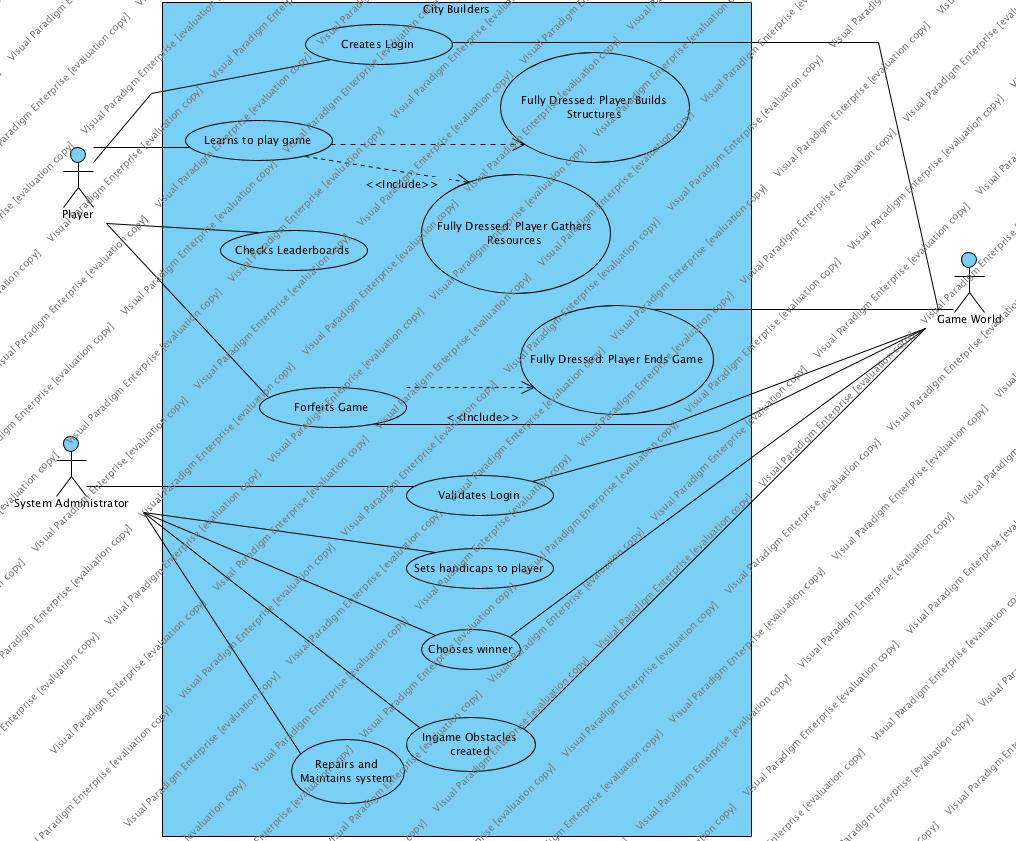
* There will be Three iterations, one Inception phase followed by two Elaboration phases. These iterations will follow the outline given in the course Syllabus. Roughly: Three weeks for the Inception, five weeks for the first Elaborations, and four weeks for the last elaboration.

Project Management:

* Project Work will be divided evenly amongst the three team members for the duration of the project.

Business Rules:

* Operating System
  + Must Be able to be run in both windows and Linux
* Time
  + Must be completed by the end of the course (12/5)
  + Specific iterations must be done at certain times (10/1, 11/5)
* Language
  + Must be built in C++
* Authentication
  + Has to have user Authentication
  + User must be 18 to consent, or get permission from someone 18 or older in order to play.
* External
  + Must be able to interface with an external system
* Different Users
  + Must have at least 2 distinct users in the system



Use Cases:

1. Player creates login
   1. Player wants to be able to use software. The player must create and ID name and password to allow the use of software whenever.
2. Administrator validates player login
   1. Administrator checks if username is allowed and that no other user has taken the same username upon login creation.
3. Player learns to play game
   1. The player wants to start the game and upon starting the game can see on the board what they must do to win. Admin notifies the player what they must do to win, and the player learns on the way as the game goes on.
4. Administrator sets handicaps to players
   1. Administrator sees a player has a noticeable lead over other players. Administrator can set some handicaps to the player to give the other players a chance to catch up.
5. Administrator chooses winner
   1. Administrator chooses winner if the game is held for too long or majority of the players wish to end the game sooner. Administrator must close the system after the game concludes.
6. Player checks leaderboards
   1. A player wishes to see how their statistics compare to other players. The player will select the leaderboards option in the menu. The player can see their ranking amongst other players. After they are finished, the player may select the option to return to the menu.
7. Player forfeits game
   1. A player would no longer like to play the game. The player will select the option to leave the game. All structures that they built and resources that they acquired will go back to the supply.
8. Administrator creates obstacles
   1. One player has achieved a sizeable advantage over the others. The system administrator was notified and chooses an appropriate response so that the players are back to a level playing field.
9. Administrator maintains the system
   1. The administrator must turn on the system in order for the players to be able to play. The administrator must also close the system after the game concludes.

**Fully Dressed Use Cases:**

1. **High Risk: Player Ends Game**
   1. A player would like to finish the game. They are able to produce their fifth structure on their turn and trigger the end of the game. The system checks to see if the end condition is met, and in this case it is. The system will allow the other remaining players of that turn to take their final turn. Afterwards, it calculates the winner by totaling the amount of points each player has and comparing it. In the event of a tie, the player with the most buildings built amongst the tied players is the winner. If there is still a tie, the player with the single highest worth building is the winner. If there is still a tie, the player with the most leftover resources is the winner. If there is still a tie, they share the title.
   2. Use Case: Player Ends Game

Scope: CityBuilder Game Application

Level: user goal

Primary Actor: Player

Stakeholders and interests:

-Player: Wants to win the game by building five structures the fastest.

-Administrator: Wants to make sure the win is delivered to the right person. Wants to make sure the win is legitimate.

-Developer: Wants to make sure the system works perfectly. Wants to make sure players can experience a game with no shortcomings.

-Evaluator: Wants to make sure our game design is good.

Preconditions: System is running, Players and Administrators are logged in, One player has enough buildings and resources

Success Guarantee: Player must build five structures before any other player.

Main Success Scenario:

1. System administrator turns the system on and starts a game session.

2. Players join the game session made by the system administrator.

3. System administrator starts the the game.

4. Players gather resources.

5. Players builds structures as they collect resources.

6. One player wins the game by building five structures before any other player.

7. The win for the game is recorded into the player’s account.

8. Players quit the game.

9. System administrator closes the game.

Extensions:

* If two or more players finish five structures in the same turn, the player with more victory points will win the game.
* If two or more players have the same amount of victory points in that turn, the player with more leftover resources will win the game.
* If there is still a tie, then there is a coin flip to see who wins.
* System Administrator may extend the game

Special Requirements:

* System works in both Linux and C++
* Shows all options in English
* System outputs victory screen in under 5 seconds

Technology and Data Variations List:

* System Administrator Override by entering user name and password

Frequency of Occurrence:

* Only once per game, since only one player can win the game.

Miscellaneous:

What is the amount of buildings needed for a win?

What happens if two players tie in every resource?

Can there be more than one winner?

1. **High Business Value:** Player Builds Structures
   1. A player would like to use resources to build a structure for points. First, the system checks if the player has the appropriate amount of resources to build the structure. In this example, the player is trying to build a bridge. A bridge requires three deposits of your wood resource bank and two steel deposits. The player has 4 wood and three steel in their inventory so the system sees that they are able to build the structure. The system takes the three wood and two steel from the inventory and uses those materials to build the bridge on behalf of that players name. After so the player is rewarded with 2 points which is the worth of such a bridge being built. This gives the user who plays this game a sense of responsibility in handling their resource count, and a feeling of controlling their environment with the right choices in deciding how they can proceed to winning the game.
   2. Use case: Player learns to play and build structures

Scope: City Builder Game Application

Level: user-goal

Primary Actor: Player

Preconditions: System is running, Players are logged in, is at the system, can input into the system, and has enough resources.

Stakeholders and interests:

Players- If the player is not able to build a structure, they cannot fully complete their turn, or gain a point in any way.

Administrator- The administrator needs to check if the users got the appropriate resources when the player decides to choose to build a structure.

Developers- need to make sure that the appropriate resources are assigned to the named actions visible by the players. They also need to evaluate the amount of points a certain structure gives, and that a structure is capable of being built in the world this game takes place. Not every building imaginable is capable of being built with separate ID and statistics assigned. There is a limit given to the portfolio of structures to choose from to build.

Evaluator- We get graded on if the project works, and this helps the project work

Success Guarantee: When player takes their turn to perform the action of building a structure they have greater than or equal to the amount of materials needed to build a structure.

Main Success Scenario:

Administrator starts the system

Players are logged in and authenticated

Player gathers resources and passes to next player

Player takes action to build a structure.

System checks for enough resources gathered by player who wants to build.

System authenticates amount of resources and rewards with points and structure if deemed successful.

Player gets new building and points.

Player ends turn.

Next player takes a turn.

Extensions:

System enters in game to check requirements.

Player is not able to build a structure where the other player has already built on a certain property.

Player wants to build structure despite no available space.

* Player uses action to destroy/take resources used on that building.
* Player gains at most 1 point for resources used to destroy structure.
* Player has the option to now use the right amount of resources to build their structure on that open property.
* System takes away a point from the player that had their structure destroyed.

Special Requirements:

* System shows updated resources within 5 seconds of the gather action being taken
* System works in both Linux and C++
* Shows all options in English

Technology and Data Variations List:

* System Administrator Override by entering user name and password

Frequency of Occurrence:

* Occurs in practically every other, if not every turn. Most crucial action needed to win game.

Miscellaneous:

What are the structure capacity?

Can a player take the same action more than once?

When can a player not take back their action?

Can a structure be worth more than another?

If there a structure limit on the world?

1. **Architecturally Significant:** Player Gathers Resources
   1. A player would like to gather a resource. During their turn, they choose which resource they would like to gather and the system adds the appropriate amount of that resource to that player’s supply. Their turn is now completed and the next player may take their turn.
   2. Use Case: Player Gathers Resources

Scope: City Builder Game Application

Level: user-goal

Primary Actor: Player

Stakeholders and interests:

Players- If the player is not able to gather a resource, they cannot play the game Administrator- The administrator needs to check if the users got the appropriate resources when the player goes to gather

Developers- need to make sure that the appropriate resources are assigned to the named actions visible by the players

Evaluator- We get graded on if the project works, and this helps the project work

Preconditions:

System is running, Players are logged in, Player is at the system, Player can input into the system

Success Guarantee:

Player has the resource gathered, Player is able to see all resources that they have, Player is able to spend resources on their next turn, Play is passed to the next player,

Main Success Scenario:

Administrator starts the system

Players are logged in and authenticated

Player takes the gather resource action

System gives the appropriate resource to the player

System updates the Player resource count

Player sees total resource count

Play passes to the next player

System gives control to the next player

Extensions:

System enters System Administrator Mode

Player is not able to gather resources because their opponent already took that action

Player wants to take back choice

* System takes back resources it gave to player
* System restores rolls back the choice to give the player their action back
* Control is returned to the player

Special Requirements:

* System shows updated resources within 5 seconds of the gather action being taken
* System works in both Linux and C++
* Shows all options in English

Technology and Data Variations List:

* System Administrator Override by entering user name and password

Frequency of Occurrence:

Happens every turn of the game

Miscellaneous:

What are the resource caps on the players?

Can a player take the same action more than once?

When can a player not take back their action?

How often can they take back an action?

Supplementary Specification

* Introduction
  + This section is to outline other items that are not addressed through our system. While our system is not as concerned with hardware requirements, there must be some minimum specifications required in order to run our game.
* Features List
  + N/A
* Functionality
  + The game data will be backlogged so that the system administrator will be able to review them at a later date.
  + All users will be authenticated through a login that they must create when they want to play. This will be stored as well for System administrator review
* Quality Attributes
  + Is usable by anyone over the age of 18 or any children that are able to get parental consent to make an account.
  + Is expected to have 99% availability while the system is running, barring extraneous errors.
  + Will respond at a decent time so that users do not lose interest. Responses between actions taking in the system should take no longer than 3 seconds to process, with an average much lower than that.
  + Will be supported by a system administrator while the game is running.
  + Can be adapted to run in both windows and linux
  + Configurable to run on most systems using these operating systems.
* Implementation Constraints
  + Must be done in C++ and be runnable in both windows and Linux.
* Purchased Components
  + N/A
* Free Open Source Components
  + Various C++ libraries used throughout the project.
* Interfaces
  + External: Mouse, keyboard
  + Internal: Command line window
* Application-Specific Domain (Business) Rules
  + A player can only take one action per turn
  + A player can gather a resource by spending their action to get an amount of resource.
  + A player builds a building by spending the appropriate resources to build it
  + A player scores points based off of the points awarded from buildings they built.
  + The game ends when all building plots are finished or a player has built 5 buildings
  + The winning player is the player with the most points at the end of the game
* Legal Issues
  + May run into problems with similar games that take player actions and make cities by spending resources. However, should not be an issue for this course, as it will more than likely fall into fair use.
* Information in Domains of Interest
  + N/A
* Reports
  + N/A
* Packaging
  + Submitted in a zip file on due dates that will be openable and runnable by the professor.
* Standards
  + Should not crash (excessively)
  + Should entice players to continue playing
  + Should store variables of player values so that they are accessible across turns.

Glossary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | Definition/Information | Format | Validation Rules | Aliases |
| Action | A choice made by the player. Can be build or gather. |  |  | Turn |
| Board | Where the players build their structures. |  |  |  |
| Build | To create a structure. |  |  |  |
| Gather | To gain a resource. |  |  | Collect |
| Player | A person who plays the game. Is able to take actions and win the game. |  | Has an authenticated log in |  |
| Plot | A space on the board taken up by a structure |  |  |  |
| Resource | An item used to build a building. Gathered by players. |  |  |  |
| Scoring | The act of scoring points. Achieved through building a structure |  |  | Victory Points |
| Spend | To use a resource. |  |  |  |
| Structure | A building in the city. Is able to be made by a player |  |  | Building |
| System Administrator | A person who maintains the system and authenticates users. |  |  |  |
|  |  |  |  |  |

Risk Registry:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Name | Description | Weight | Category | Mitigation Approach |
| 1 | Player Interest | If not enough players are interested in playing the game or not enough players are interested to keep playing the game, then people won’t be playing the game resulting in loss of profits. | H | Business | 1.Create marketing campaigns to help advertise the game  2.Create new downloadable content to create more interest in players new and old. |
| 2 | Underage Account  Creation | If the player creating the account is under 18, then the company may face some legal issues. | M | Technical | 1.Ask for date of birth upon account creation |
| 3 | Project Completion on Time | If the project isn’t complete by the deadline, then players won’t be able to purchase the game and stakeholders won’t be happy. | H | Schedule | 1.Make sure the team and project are on track to finish before the deadline |
| 4 | Memory  Requirement | If the player does not have sufficient memory in their system, then the game may not run as smoothly and can ruin the player’s experience. | L | Technical | 1.Create system requirements showing the required amount of memory needed.  2.Write efficient code to decrease the required amount of memory |
| 5 | Account Fraud | If a player is able to make more than one account, a player can manipulate the system to increase their win percentages. | M | Technical | 1.Player must be in presence of system administrator in order to create the account. |
| 6 | Player Turn Fairness | If a player who knew the system went first every turn, then the player would automatically win | L | Technical | 1.Player turns are determined by who is farthest from winning. |
| 7 | Player Quits Game | If 50% or more of the player leave, leaving one user left logged in, that user is automatically declared winner | L | Technical | 1.If all Players leave but one, then system recognizes and adjusts game for the amount of user left.  2. If All players leave, then the system automatically ends game, no winner is declared and the program terminates. |

GO/No- Go