Multiplayer Platform Game

Martin Petterson
Oskar Kvist
Christoffer Ekeroth
Misael Berrios Salas

January 4, 2008

Contents

1 Preface **2**

1.1 Expected Readership 2

1.2 Version History 2

2 Introduction **3**

2.1 Who are the users and what problem does the system solve for them? 3

2.2 The main uses of the system 4

2.2.1 Usage narratives 4

2.3 The context/environment in which the system is to be used . 5

2.4 The scope of the system 5

2.5 The main factors that need to be taken in to account when designing and building the system 6

2.6 Technologies and Risks 7

3 Glossary **9**

4 User Requirements Definition **10**

4.1 Functional Requirements 10

4.1.1 Testing Functional Requirements 10

4.1.2 List of Functional Requirements 10

4.2 Non-functional requirements 19

4.3 Use cases 20

4.3.1 Start a race 20

4.3.2 Terminate race 20

4.3.3 Use the boxing glove power-up on a monster 21

4.3.4 Overtake opponent using speed boost power-up 21

4.3.5 Avoid difficult section of game stage using levitation power-up 22

4.3.6 Jump across a gap between two platforms 23

5 System Architecture **24**

5.1 Logic 24

5.2 Graphics 25

5.3 Sound 25

5.4 Player Input 25

5.5 Physics 26

6 System Requirements Specification **27**

7 System Evolution **33**

1 Preface

1.1 Expected Readership

The expected readership of this document is all potential stakeholders of this project. For a game production this usually includes the publisher, the developer and the console manufacturer (if the game is developed for a console). In this case, however, both the publisher and the developer are the project team.

1.2 Version History

This is version 1.0.

2 Introduction

The aim of this project is to create a multiplayer platform game (see 4 in the Glossary section). Traditionally, platform games like the Super Mario Bros. (Nintendo, 1985) or Mega Man (Capcom, 1987) series are singleplayer experiences. Our aim is to introduce an element of competition to the traditional platform game formula, letting up to four players compete to be the first to reach the end of the game stage.

The game will have the classic 2D gameplay of old platform games, but the graphics will be in 3D. Each game stage is littered with dangers and obstacles that the players must avoid. Items that can be picked up to grant a player special abilities or give her other advantages are also scattered throughout the game stage. Avoiding nasties and obstacles while picking up items and using them efficently are key to winning the race.

The game will be played on the Xbox 360 games console, with up to four players sharing one screen (the screen is not split into several smaller screens), using one Xbox 360 controller each.

2.1 Who are the users and what problem does the system solve for them?

Our intended user base consists of casual gamers in the age range 8-26. They play games for at least a couple of hours a month and usually in short bursts. Wikipedia defines a casual gamer by the criteria below[1], and this will be the definition used in the document.

> Casual gamer is a term used to describe a type of video game player whose time or interest in playing games is limited compared with a hardcore gamer. They can be found playing games across all genres and of varying levels of complexity and difficulty. However, they do not devote the amount of time or practice to video games spent by hardcore gamers. Whereas a hardcore gamer may approach gaming with the mindset of a professional, a casual gamer would think more like a hobbyist.
>
> Casual gamers can conceivably consist of any people who show more than a passing interest in video games, therefore it is difficult to categorize them as a group. For this reason games which attempt to appeal to the casual player tend to strive for simple rules and ease of game play, the goal being to present a pick-up-and-play experience that people from any age group or skill level could enjoy.
>
> Not all gamers who play casual games are themselves casual gamers.

[1] http://en.wikipedia.org/wiki/Casual_gamer (23/11/07)

The problem we intend to solve is the need for a simple multiplayer game that can be enjoyed by a group of up to four people in a very social context. The success of games like Smash Brothers (Nintendo, 1999) and Guitar Hero (RedOctane, 2005) indicates that there is a demand for games that are supposed to be played in a social context, and so we believe that the gameplay experience we provide will prove valuable to our users.

As the Wikipedia entry mentions, "hardcore" gamers can still appreciate casual games, and so, even though they are not our target demographic, they might still be potential users of our product.

2.2 The main uses of the system

The main uses of the system are to entertain a group of players. The game is meant to be played with friends in a living room setting, with a typical session lasting for about 60 minutes.

2.2.1 Usage narratives

Usage narrative one:

> Kalle is hosting a party at his house, and Alice, an 18 year old female, is one of the guests. Kalle's friend Bob suggests they play our game and invites Alice to join. Kalle and Bob play the game frequently. Alice has never played the game before and is not an experienced video game player, but the concept and controls are simple so she is able to get through the game stage even though it's her first time. Kalle wins and Bob comes in second. Even though Alice lost she is eager to play again since she enjoyed the game and thinks she could perform a lot better next time. During the game the excited participants attracted the attention of the other guests who are now eager to try the game as well.

Usage narrative two:

> Christoffer, a 24 year old male, has invited some of his friends over to watch a movie he's rented, but the movie turns out to be lacklustre and they decide to stop watching and play our game instead. Christoffer and his friends play the game regularly and use advanced tactics and techniques. They pick their favourite characters and start the race through the game stage. The competition is fierce, but in a decisive moment Christoffer gets a power-up that slows his opponents down, allowing him to win the race. Even though Christoffer and his friends have played the game for a long time they still enjoy it because they feel they're still learning new things about the game.

2.3 The context/environment in which the system is to be used

The game is going to be playable on a consumer Xbox 360 using the functionality granted by joining the XNA (see 6 in the Glossary section) Creators Club (see 2 in the Glossary section), with Xbox 360 gamepads. It is reasonable to assume that the Xbox 360 version will be played on both high- and low definition televisions, in a living room setting.

2.4 The scope of the system

Topic:	In:	Out:
Singleplayer:		X
Same-screen multiplayer:	X	
Network multiplayer:		X
AI opponents:		X
Cross-platform compatability:		X
3D graphics:	X	
Physics engine:	X	
Save capabilities:		X
Sound effects:	X	
Music:	X	

Single player:
Our primary focus is on providing an enjoyable multiplayer experience, and so a singleplayer mode will not be available in this version of the game.

Same-screen multiplayer/Network multiplayer:
We decided to focus on implementing only same-screen multiplayer for two reasons: First, network multiplayer does not fit the purpose of the game. Second, as of now XNA does not support networking of any kind, and since the only library we can use on the Xbox 360 is XNA we would not be able to provide this service on both platforms.

AI opponents:
We decided against including AI-controlled players for solitaire play or substitutes for human players, since the aim of the project is to provide an enjoyable multiplayer experience.

Cross-platform compatability:
Our only target platform is Xbox 360.

3D graphics:
We will use 3D graphics since we decided that drawing and animating char-

acters in a high enough level of detail to be sufficient for high-definition displays would be far too time-consuming. The cartoon-style characters do not need to be highly detailed and thus will not be as time consuming to model as characters in more realistic games. We also chose to use 3D graphics for aesthetic reasons, as we wanted to make our game stand out from other platform games.

Physics engine:
We will integrate an open-source 2D physics engine into our game. This might sound overambitious, but we will mainly use it for collision detection since code for collision handling would have to be written anyway, using the physics engine might end up saving us time.

Save capabilities:
Since this is a multiplayer game played in short sessions there is no need for a save feature.

Sound effects:
Sound effects are a very important part of games and so we have decided to include sound effects to make our game come alive. We will not have to make the sound effects ourselves since there are numerous websites with high-quality, royalty-free sound effects available.

Music:
We would like to have our own music in the game, but if time becomes too much of an issue we have the option of using Creative Commons-licensed music instead.

2.5 The main factors that need to be taken in to account when designing and building the system

- As previously stated, the game needs to be easy to pick up and play, even if the user is not an experienced console or PC gamer. We need to find ways to keep the control scheme and basic gameplay concepts simple enough to be accessible to non-frequent videogame users.
- The Xbox 360 outputs at a native resolution of 1280 x 720, something that needs to be considered when designing graphics routines and making graphical content.
- Since up to four players share the same screen we need to devise a method for making the onscreen action readable while at the same time keeping all of the player characters visible.

- Our only input device is the Xbox 360 gamepad,and its advantages and limitations need to be taken into account. For example, the directional pad on the Xbox 360 is notorious for its lack of responsiveness.

2.6 Technologies and Risks

C#:
Games using XNA must be written in C#. However, all group members are familiar with C# and it is a very user-friendly language so the use of C# should not present any risks.

Farseer:
Farseer is the name of the open-source physics engine we plan on using. Integrating Farseer will probably not take longer than writing the code for collision detection and handling ourselves, but there is a definite risk that we have underestimated the time it will take us to integrate Farseer. The consequences of this would be either a delay in the project or having to write the collision code ourselves.

XNA:
Not all group members are experienced with XNA. The consequences of this is that it may take some time for them to get up to speed. However, using the comprehensive documentation bundled with XNA they should be able to pick it up quickly.

XNA on Xbox 360 (deployment):
Microsoft claims that the process of deploying and running XNA code on the Xbox 360 is very simple and fast. However, none of the team members have ever done this themselves, so we are unable to verify Microsoft's claims.

XNA on Xbox 360 (performance):
The Xbox 360 was developed with native C++ code in mind, and so XNA applications can not utilize the full potential of the Xbox 360. As a consequence of this our game might run too slow to be enjoyable.

Game Stage Editor:
Since creating game stages by hand is time-consuming we have decided to develop a very rudimentary game stage editor for our game. However, there is a risk that developing the game stage editor might take too much time.

3D Modeling:
No one in the group is experienced in the field of 3D modeling, which might present a problem since we've decided to use 3D graphics. The consequences of this might be that the creation of models takes up too much of the group

members' time or that the models end up looking poor. However, we hope to sidestep this issue somewhat by using cartoony graphics, which need not be as detailed as those in a more realistic game.

CVS:
We will use a concurrent versioning system to store and synchronize the code base. None of us have ever used a CVS before, which means that instead of saving us time it might end up delaying the project.

3 Glossary

1. *Creative Commons* is a license that permits anyone to use the author's work, as long as it's not for a commercial purpose.

2. *Creators Club* is a subscription service that is required in order to be able to play XNA games on an Xbox 360.

3. *Game stages* are the playing fields of the game, similar to race tracks in racing.

4. *Platform games* are a genre of videogames where the player must navigate through a side-scrolling environment (usually presented in two dimensions), often having to jump past obstacles, traps or enemies.

5. *Scrolling* is the act of sliding the game presentation of the screen.

6. *XNA* (XNA's Not Acronymed) is both a C# game programming library and an extension to Visual C# Express. Games created with XNA can run on both Windows Vista/XP and Xbox 360.

4 User Requirements Definition

4.1 Functional Requirements

4.1.1 Testing Functional Requirements

Since there are no well-defined inputs or outputs it is difficult to specify robust tests. However, each functional requirement can be tested by playing the game and observing the output on the screen.

4.1.2 List of Functional Requirements

1. **Basic Concepts**

   (a) The game shall be playable by two to four players simultaneously.

   (b) Each player controls one character in the game. (The player characters will be referred to as "players" for the rest of this document)

   (c) The players shall compete against each other on the same Xbox 360 machine.

   (d) The game shall be played using the Xbox 360 gamepad. Each player shall have a gamepad of her own.

   (e) The game takes place on a *game stage* (see 2a) with a starting position and a finish point, and with obstacles inbetween.

   (f) The game shall allow the players to move in two dimensions. Three-dimensional movement shall not be possible. The two-dimensional plane in which the characters move shall be parallell to the screen. Moving upwards in the game world shall correspond to moving upwards on the screen. Moving downwards in the game world shall correspond to moving downwards on the screen. Downwards is defined as the direction of the game's gravity (see 2c). Moving left or right on the screen shall correspond to moving left or right in the game world.

   (g) The winner of the game shall be decided by who is the first to reach the finish point (see 6a) of the game stage.

   (h) The finish point (see 2(a)iv) shall be positioned to the far right of every game stage, and the starting points (see 2e) shall be positioned to the far left.
   Rationale: Always having the finish point to the right makes the game easier to grasp.

   (i) It shall be possible to directly affect your opponents and the outcome of the game by interacting with them in various ways.

2. **Game Environment**

(a) The game shall feature the following types of objects on its game stages:

   i. *Platforms* (see 2b)
   ii. *Power-up dispensers* (see 2d)
   iii. *Starting points* (see 2e)
   iv. The *finish point* is the point the players are trying to get to.
   v. *Traps* (see 2f)
   vi. *Monsters* (see 2g)
   vii. *Springboards* are objects that catapult players or monsters that touch them, in a set direction.
   viii. *Slippery ground* is special ground that makes the player slide (see 3(b)ii) while the player runs on it.

(b) Platforms are the basic building blocks of every game stage. They shall have the following properties:

   i. Players or monsters (see 2g) shall not be able to pass through platforms in any way.
   ii. Platforms shall be unaffected by gravity (see 2c).
   iii. Platforms can be positioned anywhere in a game stage.
   iv. Platforms shall either be stationary, or move along a set path.
   v. Platforms shall be rectangular with the sides parallel to the edges of the screen.

(c) There shall be gravity in the game. All game objects that are affected by gravity will accelerate downwards at a constant rate of acceleration if they are not standing on a platform (see 2b).

(d) Power-up dispensers are stationary objects that the players shall be able to pass through in order to gain power-ups.

   i. The power-up dispensers shall continually cycle through all available power-ups, showing them to the player.
   ii. A player shall gain the power-up that is currently shown (see 2(d)i) when she touches the power-up dispenser.
   iii. A player shall not be able to get more than one power-up from the same power-up dispenser.
   Rationale: The players should not be able to get two (or even more if they apply them immediately) power-ups from the same power-up dispenser, since it nullifies the strategy of saving power-ups for later.

(e) The starting points are the points the players are starting out at, at the beginning of a race. They shall have the following properties:

i. There shall be four starting points on each game stage. *Rationale:* There can be up to four players playing the game simultaneously.

ii. The starting points shall be positioned close together. *Rationale:* Having them close together minimizes the unfair advantage of starting first.

(f) Traps shall have the following properties:

i. Traps cause the player to lose a life (see 6b) when colliding with them.

ii. Traps shall either be constantly active, or periodically turn themselves on and off.

iii. Traps shall either move along a set path or have a fixed location.

iv. Players shall be able to avoid a trap in at least one of the following ways:

A. Jump over it.

B. Wait it out.

C. Go around it.

(g) Monsters shall have the following properties:

i. Monsters shall be computer-controlled, moving characters.

A. Monsters shall move back and forth on a platform (see 2b), changing direction when reaching an edge of the platform or an obstruction.

ii. Monsters are affected by gravity (see 2c).

iii. Monsters shall be able to get stunned.

A. While stunned, the monster shall have its horizontal velocity set to 0.

B. The monster shall not be able to cause a player to lose a life (see 6b) for the duration of the stun effect.

C. The stun effect shall be visible to the players.

iv. Colliding with monsters:

A. Players shall not be able to pass through monsters and monsters shall not be able to pass through other monsters.

B. If a player collides horizontally with a monster the player shall lose a life (see 6b).

C. If a player lands on a monster, one of two things shall happen depending on the type of monster—some monsters become stunned (see 2(g)iii), others cause the player to lose a life (see 6b).

D. If a player collides with a monster located above her, she shall lose a life (see 6b).

v. Players shall be able to affect monsters with power-ups (see 4) in the same way that they can affect other players.

vi. The player shall be able to avoid a monster in at least one of the following ways:

A. Jump over it.

B. Wait it out.

C. Go around it.

D. Use a power-up to disable it in some way.

3. **Player Behavior**

(a) Player Motion

i. Vertical and horizontal movement shall be independent.

ii. A player shall not be able to occupy the same space as another player.

iii. If a player does not give the game any movement input, the game shall assume that the player wants to stand still, and decelerate (see 3(b)ii) to a standstill.

iv. A player can change her direction of horizontal motion both when standing on a platform and when in the air, but not when stunned (see 3g).

v. Players shall be affected by gravity (see 2c).

(b) Players shall be able to run.

i. There shall be a maximum running speed. A player's maximum running speed can be altered by certain power-ups (see 4d).

ii. Acceleration and Deceleration:

A. If the player starts running from a standstill then accelerate to maximum running speed.

B. If motion is forward and switches to backward (or vice versa), then decelerate to 0 and then accelerate to maximum running speed.

C. The rate of acceleration and deceleration when running shall be constant, until maximum running speed is attained, in the case of acceleration, or until the player stands still, in the case of deceleration. A player's rate of acceleration and deceleration can be altered by certain power-ups (see 4d) and slippery ground (see 2(a)viii).

(c) Players shall be able to jump.

   i. A player must be standing on a platform (see 2b) in order to jump.

   ii. When a player jumps, she instantly gets an upwards velocity. Since players are affected by gravity (see 2c) at all times, the velocity is steadily reduced and eventually the player starts falling.

(d) Players shall be able to pick up power-ups (see 4d) by touching a power-up dispenser.

(e) Players shall be able to apply a gained power-up at will.

(f) Players shall be able to become invulnerable.

   i. Whenever a player loses a life (see 6b) she becomes invulnerable for a short amount of time.

   ii. An invulnerable player shall not lose life due to actions which would otherwise cause her to lose life. The one exception from this is ending up outside the screen (see 3j), which shall always cause the player to lose life.

   iii. The invulnerability effect shall be visible to the players.

(g) Players shall be able to get stunned.

   i. While stunned, the player shall be unable to affect her own movement.

   ii. While stunned, the player shall decelerate (see 3(b)ii) her horizontal movement to a standstill.

   iii. While stunned, the player shall be unable to apply power-ups.

   iv. The stun effect shall be visible to the players.

(h) Players shall be able to stun (see 3g) an opponent by landing on her.
Rationale: This provides another way, besides power-ups, to interact with the other players that is available all the time.

(i) If a player lands on another player or monster (see 3h and 2(g)ivC) she shall automatically jump (see 3c).

(j) Players shall be able to end up outside the screen. If a player ends up outside the screen, she shall lose a life (see 6b), and then respawn (see 3k) if she has any lives left (see 6b).

(k) Players shall be able to respawn.

   i. Players shall respawn in a safe place, so that they do not lose another life immediately after respawning.

   ii. Players shall respawn close to the middle of the screen. Since players shall spawn in a safe place (see 3(k)i), spawning exactly in the middle of the screen will not always be possible.

4. **Power-ups**

   (a) The game shall have power-up dispensers (see 2d) at set locations throughout its game stages that players shall be able to touch to gain a power-up.

   (b) Players shall be able to store up to two power-ups at any time and apply either one whenever they pleases.
   Rationale: The ability to store power-ups allows for more strategic gameplay.

   (c) When a player applies a power-up, the power-up shall be consumed.

   (d) The following power-ups shall be available:

      i. A *boxing glove* power-up that can be used to punch enemies off platforms or into traps.

         A. The boxing glove, when applied, shall travel horizontally in the direction the player that applied it was facing, and retain its vertical position.

         B. If the boxing glove hits another player or a monster the victim shall instantly get a horizontal velocity greater than the maximum running speed (see 3(b)i), in the direction the boxing glove was traveling and become stunned (see 3g).

         C. If the boxing glove hits another player, a monster (see 2g) or a platform it shall be removed from the game.

      ii. A *retractable claw* that can be used to grab onto an enemy and then toss the enemy.

         A. The claw, when applied, shall travel horizontally in the direction the player that applied it was facing, and retain its vertical position.

         B. If the claw hits a platform it shall be removed from the game.

         C. If the claw hits another player or a monster the victim shall be grabbed by the claw and pulled towards the player that applied the power-up.

         D. If the player has grabbed onto an opponent with the claw power-up, she can toss the opponent at will. Tossing an opponent removes the claw from the game.

         E. Tossing an opponent shall result in the opponent instantly getting a vertical velocity upwards, and a horizontal velocity in the direction the tossing player is facing.

F. If the player has grabbed onto an opponent with the claw power-up and does not toss the victim within a set amount of time, the victim shall be released and the claw removed from the game.

iii. *Speed boost*

A. When applied, the maximum running speed (see 3(b)i) and the rate of acceleration and deceleration (see 3(b)ii) shall be increased for the player that applied the power-up, for a set amount of time.

iv. *Levitation*

A. When applied, the gravitational acceleration (see 2c) of the player that applied the power-up shall be reduced, for a set amount of time.

v. *Fetter*

A. When applied, a fetter will be placed at the position of the player than applied the power-up.

B. The fetter shall be affected by gravity (see 2c).

C. If a player or monster (see 2g) comes within range of a fetter after it has been placed, the fetter shall grab onto it.

D. The fetter shall not be able to catch (see 4(d)vC) the player that placed it.

E. If a player or monster has been caught by a fetter (see 4(d)vC) she will have her maximum movement speed and jumping height reduced, for a set amount of time. When the player or monster is released the fetter is removed from the game.

vi. *Shrinking ray*

A. The shrinking ray, when applied, shall travel horizontally in the direction the player that applied it is facing, and retains its vertical position.

B. If the shrinking ray hits another player or a monster the victim will be shrunk and have her maximum movement speed and jumping height reduced, for a set amount of time.

C. If the shrinking ray hits another player, a monster or a platform the ray shall be removed from the game.

vii. *Banana peels*

A. The banana peels power-up, when applied, places three small banana peels close to the player that applied the power-up.

B. The banana peels shall be affected by gravity (see 2c).

C. If a player or monster steps on a banana peel, she will become stunned (see 3g) and have her rate of acceleration and deceleration decreased (see 3(b)ii), for a set amount of time.

D. The banana peels shall not be able to affect the player that applied the power-up.

5. **The Screen**

   (a) The game shall provide information about how many lives the players have left and what power-ups they currently have.
   Rationale: The amount of lives a player has left and the player's power-ups are important information that need to be accessible.

   (b) All players shall occupy the same screen. The game will not split the screen into smaller areas reserved for each player.
   Rationale: Splitting the screen reduces the feeling of competing against other players, since the player would be unable to see all of her opponents at once. Having several small screens would also make it harder for the player to get an overview of the game world.

   (c) The field of view of the screen shall be fixed; the camera shall not zoom in and out.
   Rationale: The camera shall not zoom in because zooming in reduces the players' overview of the game stage. On the other hand, zooming out can make things hard to see. Therefore, it is best to find an optimal field of view and stick with it.

   (d) The camera shall follow the player who is in the lead. The player in the lead shall be about one third of the screen from the right edge of the screen.
   Rationale: Since the finish point of every game stage is on the far right (see 1h), the player in the lead will be the rightmost player. Therefore, two thirds of the screen will be behind the leader because that is where the other players will be. At the same time, there must be sufficient space in front of the leader so that she can see objects coming up ahead.

   (e) If a player gets too far behind the leader she will end up outside the screen (see 3j) since the camera follows the leader (see 5d).

6. **Win and Loss Conditions**

   (a) The player who reaches the finish point of the game stage first shall win. The game shall continue until all players have either lost all of their lives (see 6b) or reached the finish point.

(b) A player starts the game with a certain amount of lives. If a player loses all of her lives she shall automatically forfeit the game.
Rationale: The game should encourage skillful navigation through its game stages.

(c) If there is only one player left, she shall win.
Rationale: The players that have forfeited the race shall not have to wait for the single remaining player.

7. **Presentation**

(a) The game shall feature music. There shall be at least one song, at least two minutes long.

(b) The game shall feature sound effects for the following events:

i. A player jumps

ii. A player loses a life

iii. A player reaches the finish point

iv. A player uses a power-up

v. A player lands on another player

vi. A player lands on a platform

8. **Character and Game Stage Selection**

(a) Players shall be able to select characters. Two players will not be able to select the same character.
Rationale: Being able to select different characters helps differentiate the players and helps the players identify with their on-screen representations.

(b) Apart from appearance, the characters shall be identical. Choice of character does not affect gameplay.
Rationale: Making all characters the same makes balancing the game easier. It also makes the game more fair, since two players cannot select the same character (see 8a).

(c) There shall be at least three game stages in the game.

(d) A player shall be able to select which game stage to play on.

(e) The player who gets to select the game stage is selected at random.
Rationale: Letting a random player select which game stage to play on is a fair method of game stage selection.

9. **Pauses and Pause Options**

(a) Any player shall be able to pause the game at any time during a race.

(b) Only the player that paused the game shall be able to resume it.

(c) While the game is paused, the player that paused the game shall be able to choose one of the following alternatives:

i. Resume the race

ii. Restart the race at the same game stage

iii. Abort the race

10. **The End of a Race**

(a) When a race ends, the players shall be informed about in what places they finished or, in the case that a player lost all of her lives (see 6b), that she lost all of her lives.

4.2 Non-functional requirements

1. The game shall have a refresh rate of at least 30 frames per second, at all times.
   Rationale: The screen refresh time must be high enough to provide smooth movement and scrolling. Low refresh times will produce slow or choppy animation and scrolling, which will detract from the gameplay experience.

2. It shall not take the game more than 100 milliseconds to respond to user input.
   Rationale: Low response time to input is important in order for the game to feel responsive, which is an important quality for games.

3. The game shall be playable on any Xbox 360 console as long as it's equipped with an Xbox 360 memory card or hard-drive.
   Rationale: We don't want to exclude users by making our game larger than the size of the smallest available storage units.

4. It shall not take more than ten seconds between starting the game and getting to the menu screen. When players have selected their characters and a game stage it should not take more than five seconds to actually start playing.
   Rationale: A short game session might only last for half an hour, and if it takes five minutes to start a game that means one sixth of the session is spent on not playing.

5. The control scheme shall not include more than three buttons and one directional pad. The player shall not be required to move her fingers between buttons while playing.
   Rationale: If the control scheme uses too many buttons then inexperienced users might become uninterested in playing the game.

6. The control scheme shall not map more than one function to each button. Users shall not be required to use combinations of buttons to perform an atomic action, and the functions mapped to each button should not change during the course of the game.

4.3 Use cases

4.3.1 Start a race

Primary actor: The players
Stakeholders and Interests:
The players: Want to select a game stage and start playing the game. Each player wants to select a character. Some players might want to select certain characters.
Preconditions: The game software has been started and has finished loading. Each player has an Xbox 360 gamepad.
Minimal Guarantee: Each player has selected a character. A game stage has been selected. The race starts.
Success Guarantee: Each player has selected the character they wanted to select.
Trigger: The players want to start a new race.
Main Success Scenario:
1. Players confirm that they are to participate in this race.
2. Players select the character they want to play as.
3. System selects a player at random to choose the game stage.
4. Selected player chooses a game stage.
5. System starts the race.

Extensions:
2a. Another player selects the character the player wants to select.
  2a1. The player selects a different character.
Frequency of Occurrence: Once before the start of each race.
Open Issues: None

4.3.2 Terminate race

Primary actor: The players
Stakeholders and Interests:
The players: Want to terminate the race.
Preconditions: A race is currently in progress.
Minimal Guarantee: Race ends
Success Guarantee: None
Trigger: The players decide that they want to terminate the race.
Main Success Scenario:
1. A player pauses the game.
2. Player chooses to terminate the race.

3. System terminates race.
4. System resets itself to its initial state.
Extensions:
None
Frequency of Occurrence: Once every few races.
Open Issues: Must all players agree that the race shall be terminated?

4.3.3 Use the boxing glove power-up on a monster

Primary actor: A player
Stakeholders and Interests:
The player: Wants safe passage.
Preconditions: The player has obtained a boxing glove power-up.
Minimal Guarantee: The power-up is consumed.
Success Guarantee: The player passes safely.
Trigger: The player notices a monster is in the way.
Main Success Scenario:
1. Player comes within range to use the boxing glove power-up.
2. Player applies boxing glove power-up.
3. Boxing glove hits monster.
4. Monster is punched off the platform.
5. Player passes safely.
Extensions:
3a. Boxing glove misses monster.
  3a1. Monster is unaffected.
  3a2. Player jumps over monster.
    3a2a. Player mistimes his jump and collides with the monster.
      3a2a1. Player loses a life.
  3a3. Player passes safely.
4a. Monster is not punched off the platform.
  4a1. Monster remains on the platform.
  4a2. Player jumps over monster.
    4a2a. Player mistimes his jump and collides with the monster.
      4a2a1. Player loses a life.
  4a3. Player passes safely.
Frequency of Occurrence: A few times each race.
Open Issues: None

4.3.4 Overtake opponent using speed boost power-up

Primary actor: A player
Stakeholders and Interests:
The player: Wants to overtake his opponent.

The player's opponent: Wants to remain in front of the player.
Preconditions: The player has obtained a speed boost power-up.
Minimal Guarantee: The power-up is consumed.
Success Guarantee: The player overtakes his opponent.
Trigger: The player feels he's in a good position to overtake his opponent.
Main Success Scenario:
1. Player applies speed boost power-up.
2. System increases the maximum speed of the player.
3. Player jumps over his opponent and overtakes her.

Extensions:
3a. Opponent jumps and obstructs player.
    3a1. Player remains behind his opponent.
3b. Opponent applied fetter power-up.
    3b1. Player gets caught by the fetter.
    3b2. Player slows down and remains behind his opponent.
3c. Opponent applies speed boost power-up.
    3c1. Opponent's maximum speed is increased.
    3c2. Player is unable to overtake his opponent and remains behind his opponent.
Frequency of Occurrence: A few times each race.
Open Issues: None

4.3.5 Avoid difficult section of game stage using levitation power-up

Primary actor: A player
Stakeholders and Interests:
The player: Is unsure if she can make it past a difficult section of the game stage by normal means without losing life.
The player's opponents: Might want to force the player into the difficult section of the game stage in order to overtake the player.
Preconditions: The player has obtained a levitation power-up.
Minimal Guarantee: The power-up is consumed.
Success Guarantee: The player makes it past the difficult section without losing any lives.
Trigger: The player notices a difficult section of the game stage ahead of her.
Main Success Scenario:
1. Player approaches the difficult section of the game stage.
2. Player applies levitation power-up.
3. System reduces the gravitational forces it exerts on the player.
4. Player jumps.
5. Player lands safely on the other side.

Extensions:

1a. An opponent uses a boxing glove power-up, punching the player into something that causes the player to lose life.

1a1. The player loses a life.

3a. The player misjudges her jump, landing in or on something that causes her to lose life.

3a1. The player loses a life.

Frequency of Occurrence: A few times each race.

Open Issues: None

4.3.6 Jump across a gap between two platforms

Primary actor: A player

Stakeholders and Interests:

The player: Wants to get from one platform to another without losing life.

Preconditions: None

Minimal Guarantee: None

Success Guarantee: Player lands safely on the other side

Trigger: Player noitces a gap she has to cross.

Main Success Scenario:

1. Player approaches the gap between the two platforms.
2. Player jumps.
3. Player lands safely on the other side.

Extensions:

3a. Player misjudges her jump and falls into the gap.

3a1. Player ends up outside the screen.

3a2. Player loses a life.

3a3. Player respawns.

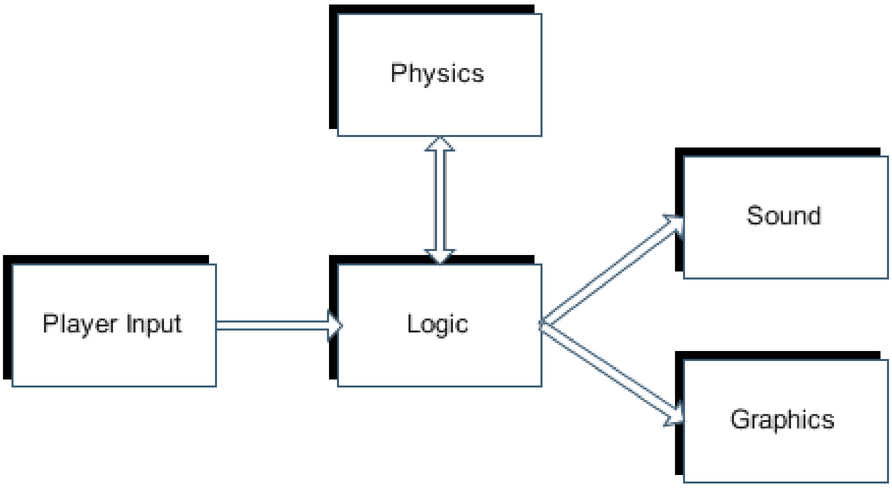
3a3a. Player does not have any lives left.

3a3a1. Player forfeits the race.

Frequency of Occurrence: Many times each race.

Open Issues: None

5 System Architecture



5.1 Logic

The Logic module is responsible for:

- Keeping track of the game state (e.g. player positions, game stages, etc.).
- Handling menus.
- Enforcing game rules during a race.
- Loading game stages from the hard drive.

Input: The Logic module will get its input from the Player Input module and the Physics module. The input from the Player Input module will consist of player actions and menu events. The input from the Physics module will consist of updated game object positions and rotations (after the Logic module has requested them).
Output: The Logic module will produce output to several other modules.

- Game events to the Sound module.
- Game objects, camera position and user interface elements to the Graphics module.
- Game objects and movement information to the Physics module.

5.2 Graphics

The Graphics module is responsible for:

- Drawing everything in the game.
- Loading graphical content from the hard drive.

Input: The Graphics module will get its input from the Logic module. The input will consist of game objects, camera position and user interface elements.
Output: The output from the Graphics module will be the graphics on the screen.

5.3 Sound

The Sound module is responsible for:

- Playing sound effects.
- Playing music.
- Loading sound content from the hard drive.

Input: The Sound module will get its input from the Logic module. The input will consist of game events.
Output: The output from the Sound module will be the sounds played in the speakers.

5.4 Player Input

The Player Input module is responsible for:

- Getting input from the players and interpret it as game commands.
- Loading input configuration from the hard drive.

Input: The Player Input module will get its input from the players. The input will consist of key presses on the Xbox 360 gamepad.
Output: The output from the Player Input module will be player action and menu events. The output from the Player Input module will be used by the Logic module.

5.5 Physics

The Physics module is responsible for:

- Calculating positions and rotations of game objects.
- Loading physics configuration (e.g. maximum player running speed) from the hard drive.

Input: The Physics module will get its input from the Logic module. The input will consist of game objects and information about where they are going or want to go.
Output: The output from the Physics module will be updated positions and rotations of the game objects. The output from the Physics module will be used by the Logic module.

6 System Requirements Specification

Note: Throughout this section we will use t as a unit of measurement that refers to the tile size (see system requirement 1(a)i). Seconds will be written s, and thus t/s and t/s^2 corresponds to tiles per second and tiles per square second, respectively.

1. **Game Environment**
   (a) *Platforms*, in addition to having the properties described in user requirement 2b, shall be *tile-based*, which entails the following:
      i. There is a smallest possible height and width of a platform. The smallest possible height and width are equal. The length of the sides of the smallest possible platform is hereafter referred to as the *tile size*.
      ii. The height and width of a general platform are always multiples of the tile size (see system requirement 1(a)i).
      iii. Stationary platforms can only be placed on a grid, where each cell's height and width is the tile size (see system requirement 1(a)i). Such a cell is called a *tile*, and the grid is called the *tile grid*.
      iv. Moving platforms (see user requirement 2(b)iv) shall have the following properties:
         A. Moving platforms shall be able to move vertically and horizontally along any set path, but shall not be able to rotate.
         B. Moving platforms can move independently of the tile grid (see system requirement 1(a)iii).
   (b) *Springboards* is a special type of platform (see system requirement 1a) with the following properties:
      i. If a player or monster comes in contact with the upwards-facing side of the springboard, the velocity of the player or monster shall be set to 8 t/s upwards. Its horizontal velocity remains unchanged.
   (c) *Slippery ground* is a special type of platform which reduces the rate of acceleration and deceleration (see user requirement 3(b)ii) of players when standing (or running) on the platform to 1 t/s^2. (Elaborating user requirement 2(a)viii)
   (d) *Gravity* shall accelerate the objects that are affected by it downwards at a rate of 2 t/s^2. (Elaborating user requirement 2c)
   (e) *Power-up dispensers*, in addition to having the properties described in user requirement 2d, shall have the following properties:

i. Power-up dispensers shall be the size of a tile (see system requirement 1(a)iii).

ii. Power-up dispensers can only be placed on the tile grid (see system requirement 1(a)iii).

(f) *Monsters*, in addition to having the properties described in user requirement 2g, shall have the following properties:

i. Monsters shall move horizontally at a velocity of 3 t/s in whichever direction they are facing.

ii. Monsters shall not accelerate and decelerate—when they encounter an obstruction (see system requirement 1(f)v), their horizontal velocity shall be set to 3 t/s in the opposite direction.

iii. Monsters shall be rectangular with the width and height 1 t

iv. If a monster gets stunned (see user requirement 2(g)iii), the following shall happen:

A. The monster is stunned for 3 s.

B. The monster shall have its horizontal velocity set to 0 t/s for the duration of the stun.

C. The stun effect shall be visible to the players.

D. After the stun effect has worn off, the monster resumes normal action.

v. An *obstuction* (see user requirement 2(g)iA) is any of the following:

A. Another monster

B. A horizontal platform wall

C. A player

D. A trap

2. **Player Properties and Behavior**

(a) A player character in the game world shall be rectangular with height 1.5 t and width 1 t.

(b) Horizontal Movement:

i. The maximum running speed (see user requirement 3(b)i) shall be 5 t/s

ii. The rate of acceleration and deceleration (see user requirement 3(b)ii) shall both be 3 t/s^2, when running on a platform.

iii. When in the air, the rate of acceleration and deceleration (see user requirement 3(b)ii) shall both be 2 t/s^2.

(c) Vertical Movement:

i. When a player jumps (see user requirement 3c) she shall instantly get an upwards velocity of 6 t/s.

ii. When a player "lands on" (see system requirement 3f) another player or a monster, she shall instantly get an upwards velocity of 6 t/s.

(d) After losing a life, players shall be invulnerable (see user requirement 3f) for 3 s.

(e) If a player gets stunned (see user requirement 3g) the stun effect shall last 3 s.

(f) Respawning (see user requirement 3k):

i. If a player ends up outside the screen (see user requirement 3j), there shall be a 2 s delay before she respawns.

ii. A *safe place* (see user requirement 3(k)i) for players to respawn at shall have the following properties:

A. Players shall land on a platform after respawning.

B. Players shall not spawn inside other players, monsters or platforms.

(g) A player shall be considered to be outside the screen (see user requirement 3j) if the player rectangle is completely outside the screen (see system requirement 5).

3. **Collision Detection and Handling**

(a) Everything in the game shall be rectangular.

(b) A player or a monster shall be considered standing on a platform if the bottom of the player touches the top of the platform.

(c) A player is considered colliding with a trap if the player's rectangle intersects the trap's rectangle.

(d) A player or a monster shall never intersect a platform.

(e) If a player's or monster's path of motion is obstructed by another player, a monster or a platform the player's or monster's velocity in the direction of the obstruction is set to 0 t/s

(f) That a player "lands on" an opponent or monster as in user requirement 3h and 2(g)ivC shall mean that the base of the player rectangle touches the top of the opponent's or monster's rectangle.

4. **Power-ups**

(a) Boxing Glove Power-up:

  i. The boxing glove, when applied, will travel horizontally in the direction the player that applied it was facing, at a velocity of 12 t/s

  ii. If the boxing glove hits another player or a monster the victims horizontal velocity shall instantly be set to 9 t/s in the direction the glove was travelling. The victim shall also become stunned (see user requirements 3g and 2(g)iii).

(b) Retractable Claw Power-up:

  i. The retractable claw, when applied, will travel horizontally in the direction the player that applied it was facing, at a velocity of 12 t/s

  ii. If the claw hits another player or monster, the following shall happen:

    A. The victim loses control of her own movement.

    B. The victim is dragged towards the player that applied the power-up at a velocity of 5 t/s. If the victim is obstructed while dragged, the normal rules of collisions apply (see system requirement 3).

    C. When the victim has come so close to the player that applied the power-up that they touch eachother, the player that applied the power-up gains the ability to toss the victim.

  iii. If the player tosses the victim, the following shall happen:

    A. The victim becomes stunned (see user requirements 3g and 2(g)iii).

    B. The victim instantly get an upwards velocity of 5 t/s.

    C. The victim instantly get a horizontal velocity of 5 t/s in the direction the tossing player is facing.

  iv. If the player that applied the power-up does not toss the victim within 2 s after she has gained the ability to toss (see system requirement 4(b)iiC), the victim shall be released, and the claw removed from the game.

(c) Speed Boost Power-up:

  i. When applied, the maximum speed of the player that applied it shall be increased to 7 t/s for the duration of the speed boost effect.

  ii. When applied, the rate of acceleration and deceleration of the player that applied it shall be increased by 1 t/s^2 from whatever it should have been otherwise, for the duration of the speed boost effect.

  iii. The duration of the speed boost effect shall be 5 s.

- (d) Levitation Power-up:
  - i. When applied, the gravitational acceleration of the player shall be reduced to 1.5 t/s^2 for the duration of the levitation effect.
  - ii. The duration of the levitation effect shall be 7 s.
- (e) Fetter Power-up (elaborating user requirement 4(d)v):
  - i. If a player or monster comes within a radius of 1.5 t of the fetter after it has been placed the player or monster shall be grabbed by the fetter.
  - ii. If a player or monster has been grabbed by the fetter they shall have their maximum horizontal movement speed reduced to 3 t/s and jumping velocity (see user requirement 3(c)ii) reduced to 4 t/s for the duration of the fetter effect.
  - iii. The duration of the fetter effect shall be 4 s.
- (f) Shrinking Ray Power-up:
  - i. The shrinking ray, when applied, will travel horizontally in the direction the player that applied it was facing, at a velocity of 12 t/s
  - ii. If the shrinking ray hits another player or a monster the following shall happen:
    - A. The victim's maximum running speed (see user requirement 3(b)i) shall be reduced to 3 t/s for the duration of the shrinking ray effect.
    - B. The victim's jumping velocity (see user requirement 3(c)ii) shall be reduced to 4 t/s for the duration of the shrinking ray effect.
    - C. The duration of the shrinking ray effect shall be 4 s.
- (g) Banana Peels Power-up:
  - i. The banana peels power-up, when applied, shall spawn three banana peel objects, distributed as follows: One peel will spawn at the player's feet and the two remaining peels will spawn 1 t in front of and behind the first peel respectively.
  - ii. The banana peel obejcts shall be rectangular with the height 0.1 t and the width 0.5 t.
  - iii. If a player or a monster intersects the banana peel object, the following shall happen:
    - A. The victim shall become stunned (see user requirements 3g and 2(g)iii)
    - B. The victims rate of acceleration and deceleration (see user requirement 3(b)ii) shall be set to 1 t/s^2 for the duration of the banana peels effect.

C. The duration of the banana peels effect shall be 3 s.

5. **The Screen**

   (a) The screen shall cover an area in the game world that is 30 tiles high, and 17 tiles wide (see system requirement 1(a)iii).

   (b) Scrolling

      i. The screen shall scroll to the right in such a way that any one of the players never come closer than one third of the screen width to the right edge of the screen.

      ii. The screen shall not scroll to the right faster than the movement speed of the rightmost player.

      iii. The screen shall not scroll to the right unless any one of the players is just one third away from the right edge of the screen.

      iv. The screen shall not be able to scroll up, down, or to the left.

6. **Win and Loss Conditions**

   (a) Each player shall start the game with 3 lives. (Elaborating user requirement 6b)

7 System Evolution

The development of the system, which is a game, is based on the assumption that users will have fun playing it. Most of the requirements are designed in order to make the game more fun.

But what makes a game fun? It is a hard question, but there are those who have tried to answer it. Chris Crawford and Raph Koster have both tried and written books on the subject[2]. They reached largely the same conclusion; people play games to learn things, and learning is fun.

Some might argue that learning is not fun. Some children do not like going to school, for example. But that is not quite the same kind of learning that Chris Crawford and Raph Koster are talking about. They are talking about learning by playing games. Children of all cultures do it, and even some animals. They do it to practise things that they are going to need when they enter adulthood. Arguably, today in human socities, children do not need to learn how to hunt, hide, climb, etc.—things that games, like Hide and Seek, teach. But those have been important skills from an evolutionary perspective.

Even if we do not learn things that are valuable for our survival when we play games like Chess, Poker or other modern games, computer games included, the process of learning is largely the same for Chess as for Hide and Seek. The player figures out and tries different tactics, fails or succeeds, and improves.

When the game offers no more to learn, it stops being fun. Simple games like Tic-tac-toe does not offer much to learn, and consequently most people do not think that it is a fun game. Games that are too complex, on the other hand, are too difficult to make sense of, and learning anything from them is hard. And so they too are not fun. Of course, different people will think different games are interesting and just complex enough.

In general, games that are easy to make sense of, yet have lots of things to learn, are good games. Having human opponents in the game also helps make it more interesting and fun, because the player does not only need to master the rules of the game, but also master her opponents.

Based on the theories above, one can conclude that games as a medium will stay entertaining for a long while. However, long time users of the system will demand new challenges and updates in order to maintain interest in the game. Things that can keep these users entertained include: balance fixes, new game stages, new power-ups, the ability to create their own games stages, etc.

Games as a medium will no doubt evolve. Games today are different from older games in many respects, including more advanced graphics, better user

[2] *The Art of Computer Game Design* by Chris Crawford, 1982, `http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html`
A Theory of Fun by Raph Koster, 2004, Paraglyph Press

interfaces, and more content. In order to keep attracting new players and sustain the interest of current players, the game must keep up with this development.

Xbox 360 will eventually be replaced by a more powerful console, and in order to survive, the game must adapt and be ported to a new console.