

FOOT GAME+ 2D

GAME DESIGN DOCUMENT



Auteurs	Approbateurs	Validation
Amine Rhazi Nouaïm Souiki	Amine Rhazi Aimad Goussa Imane Srhir	
Rédigé le : 29-10-2014	Approuvé le :	Validé le :

Diffusion	Externe
À :	Aznam Yacoub Amine Hamri
Copies à :	Amine Rhazi Aimad Goussa Nouaïm Souiki Imane Srhir Amal Weslati Naoufal Sabihi

Document de référence :

Libellé	Document
Tutoriel « Comment rédiger un bon GDD ».	Modele_gdd.pdf

Historique :

N° Version	Auteurs	Approbateurs	Date	Historique des évolutions
1.0	Amine Rhazi	Toute l'équipe	31/10/2014	Création du document. Rédaction de la section 2.1
1.1	Nouaïm Souiki	Toute l'équipe	31/10/2014	Rajout des sections 2.3, 4, 5, 6, 7, 8, 9
1.2	Aimad Goussa	Toute l'équipe	31/10/2014	Rajout de la section 3
1.3	Naoufal Sabihi	Toute l'équipe	31/10/2014	Rajout du mode mono-joueur.
1.4	Imane Srhir	Toute l'équipe	31/10/2014	Rajout de la section 1
1.5	Amal Weslati	Toute l'équipe	31/10/2014	Rajout du mode multi-joueurs
1.6	Nouaïm Souiki		01/11/2014	Correction header

				Complétion des sections 6.8 et 6.9
--	--	--	--	---

Table des matières

1. Présentation générale	6
1.1 Philosophie	6
1.2 Questions fréquentes	6
1.2.1 Qu'est-ce que ce jeu ?	6
1.2.2 Où le jeu se passe-t-il ?	6
1.2.3 Qu'est-ce que je contrôle ?	6
2. Mécaniques de jeu	6
2.1 Gameplay.....	6
2.1.1 Description	6
2.1.2 Eléments du Gameplay.....	6
2.3 Moteur du jeu.....	8
2.3.1 Généralités	8
2.3.3 Caméras.....	8
2.3.4 Lumières et Ombres	9
2.3.5 Effets Spéciaux.....	9
2.3.6 Gestion du son.....	9
2.3.7 Gestion physique	9
2.3.8 Multi-joueurs	9
2.3.9 Support multi-langues	10
3. Interface utilisateur	10
3.1 generalites	10
3.2 Les périphériques de contrôle.....	10
3.3 Les procédures fonctionnelles.....	11
3.3.1 Au démarrage.....	11
3.3.2 Configuration du jeu.....	11
4. Editeurs (univers et entités)	11
5. L'Univers du jeu	11
5.1 Généralités	11
5.2. Endroits – Lieux	12
5.3. Echelle	12
5.4. Temps et Climat.....	12
6. Les entités de l'univers	12
6.1. Généralités	12
6.2. Les entités jouables	12

6.3. Les entités non jouables	13
6.4. Les objets.....	13
6.5. Les compétences	13
6.6. La création d'entités.....	15
6.7. Les évolutions des différentes entités.....	15
6.8. Orientation et Déplacements	15
6.9. Interactions inter-entités et entités-univers	15
➤ Interaction inter-entités :	15
➤ Interaction entités-univers :	16
6.10. Moyens de locomotion.....	16
7. Histoire	16
7.1. Synopsis.....	16
7.2. Découpage interactif du scénario, dialogues	16
7.3. Références et clins d'œil	16
8. Choix Artistiques : graphisme et animation	17
8.1. Generalites et objectifs	17
8.2. Aspects 2D	17
8.3. Animations.....	17
8.4. Cinématiques.....	17
8.5. Effets spéciaux.....	18
8.6. Interface utilisateur	18
9. Choix Artistiques : musiques et bruitages	19
9.1. GENERALITES et objectifs	19
9.2. Interface utilisateur et ambiance sonore	19
9.3. Musiques	19
9.4. Bruitages.....	19
9.5. Voix.....	20
9.6. Effets spéciaux.....	20

1. PRESENTATION GENERALE

1.1 PHILOSOPHIE

Nous développons un prototype d'un jeu vidéo de type arcade avec des parties rapides soit contre ordinateur soit entre 2 joueurs, l'idée est de prendre du bon temps et jouer une bonne partie de football.

1.2 QUESTIONS FREQUENTES

1.2.1 QU'EST-CE QUE CE JEU ?

C'est un jeu vidéo d'arcade footballistique en deux dimensions avec deux équipes de 3 joueurs, vue de dessus.

1.2.2 OU LE JEU SE PASSE-T-IL ?

Le jeu se passe dans un terrain de football.

1.2.3 QU'EST-CE QUE JE CONTRÔLE ?

Le joueur ne peut contrôler qu'un seul attaquant ou défenseur à la fois.

Le gardien n'est pas contrôlable.

➤ DE COMBIEN DE PERSONNAGES AI-JE LE CONTRÔLE ?

Le joueur a le contrôle de deux personnages, l'attaquant et le défenseur, il ne peut pas contrôler les deux en même temps.

➤ QUEL EST LE BUT DU JEU ?

Le but du jeu est de marquer des buts pour battre l'équipe adverse, autrement dit contrôler la balle et être capable de la faire rentrer dans la zone de but de l'autre côté du terrain.

➤ QU'EST-CE QUI REND MON JEU DIFFÉRENT ?

Ce qui rend ce jeu original, c'est l'absence de la fatigue, des fautes et des sorties en touche.

2. MECANIKES DE JEU

2.1 GAMEPLAY

2.1.1 DESCRIPTION

Le jeu sera facile à contrôler puisque l'avatar va se déplacer d'un point en utilisant les flèches de direction, après, lorsqu'on tape la touche de passe ou de frappe en libérant la touche de direction la balle va se déplacer vers la direction choisie.

La trajectoire de la balle sera calculée par le *Physics Engine*.

2.1.2 ELEMENTS DU GAMEPLAY

Le joueur va être en interaction principalement avec les deux avatars qu'il contrôle, il les déplace dans les directions voulues ; il interagit aussi avec le ballon puisqu'il effectue des

passes avec ou éventuellement des frappes ; on peut aussi considérer qu'il interagit avec les avatars adverse puisqu'il essaie de les arrêter et leur enlever la balle.

➤ **DONNEES STATISTIQUES ET PHYSIQUES :**

Pendant le jeu, lorsque l'avatar perd le ballon si il entre en collision avec l'avatar adverse et si l'avatar adverse a tenté de tirer sur le ballon.

➤ **DESCRIPTION DE L'IA**

La machine va interagir avec les actions du joueur, par exemple lorsqu'il se déplace avec un avatar qui a le ballon, il y aura au moins un avatar de l'équipe adverse qui va s'approcher de lui et essayer de lui enlever la balle.

Et lorsqu'il frappe la balle vers les buts, le gardien va se déplacer vers la direction de la balle.

➤ **MODES DE JEU**

Les modes de jeu à implémenter obligatoirement sont

1. Player vs Player
2. Player vs AI
- 3.

❖ On pourra éventuellement rajouter un mode jeu en réseaux.

➤ **GESTION DE L'AIDE :**

Un menu d'aide sera implémenté.

Soit dans la GUI directement, soit via un état du jeu.

➤ **GESTION DE LA DUREE**

La durée de chaque partie dure 5 minutes avec temps additionnel inclus.

➤ **MODES MONO-JOUEUR**

○ *GENERALITES*

Le mode mono joueur réfère à un mode de jeu impliquant un seul joueur durant la partie.

○ *LES OBJECTIFS :*

Le joueur a essentiellement un objectif qui est de gagner le match, et cela en marquant plus de buts que l'équipe adverse.


○ *CONDITIONS DE VICTOIRE OU D'ECHEC :*

La partie se termine lorsque le temps du match s'écoule.

L'équipe ayant le meilleur score sera déclarée gagnante.

Si le score est identique de part et d'autre, il y aura égalité.

❖ On pourra gérer le cas de match nul via les prolongations, mais pour cela il faudra implémenter un type de match de tournois ou de compétition.

 A spécifier plus longuement.

- *PAUSE*
On peut mettre le jeu en pause, dans cet état, le joueur aura un menu de choix, s'il veut reprendre ou quitter le jeu
- *PERSONNALISATION*
Avant de commencer la partie, le joueur a la possibilité de personnaliser les paramètres de jeu depuis l'écran Paramètres. Choisir la durée du match, la couleur du maillot de l'équipe et le ballon qu'il va utiliser (cf.

➤ **MODE MULTI-JOUEURS**

- *GENERALITES :*
Dans le cadre général des jeux vidéo, le multi-joueurs permet à plusieurs personnes de participer ensemble et simultanément à une même partie. Cette fonctionnalité peut se réaliser en partageant le même matériel.
- *LES OBJETS DE QUETES OU OBJECTIFS :*
Les matchs ne peuvent se disputer qu'avec 3 joueurs par équipe.

Chaque équipe essaie d'arriver à la zone de l'équipe adverse et de mettre le ballon dans les buts de cette dernière.
- *CONDITIONS DE VICTOIRE ET D'ECHEC :*
L'équipe gagnante est celle qui envoie le plus grand nombre de fois possibles le ballon dans le but de l'équipe adverse, automatiquement, l'autre équipe est perdante
- *NOMBRE DE JOUEURS :*
Deux gamers qui s'affrontent.
- *MODE ECRAN DIVISE :*
Dans notre cas, on aura besoin juste d'un seul écran puisque les deux joueurs jouent sur le même écran et le même terrain mais avec des avatars différents.

2.3 MOTEUR DU JEU

2.3.1 GENERALITES

Le rendu graphique du jeu est géré par un moteur graphique interne au jeu, qui constitue une composante de la deuxième couche de l'architecture du jeu (donc une composante de la couche *Engine*).

Il s'agit d'un rendu en 2 dimensions basique.

2.3.3 CAMERAS

Une seule caméra est implémentée pour ce jeu.

La caméra est une vue à la 3^e personne, portant sur le stade en entier, depuis le dessus vertical du stade. Elle doit englober le stade et les tribunes des spectateurs.

Pendant tout le cours du match, la caméra ne bouge jamais.

Cette vue est appelée GlobalView, et est la vue principale du jeu.

- ❖ Optionnellement, la caméra peut offrir trois vues supplémentaires à ces occasions :
 - Aux moments des coups d'envoi (i.e. en début de périodes, et après la réalisation d'un but), la caméra offre une vue centrée sur le ballon, et ne contenant que le cercle du coup d'envoi, en plus des joueurs de terrain des deux équipes adverses (cf. section 6.2). Cette vue sera appelée StartingView.
 - Lorsque le gardien tient la balle de ses deux mains (la caméra offre une vue sur le gardien qui va dégager le ballon). Ceci se produit soit après une sortie de balle en corner (donc dégagement sur six-mètre), soit après que le gardien ait réalisé un arrêt sur le tir d'un joueur adverse (donc dégagement aléatoire par le gardien). La vue offerte est centrée sur le gardien, et ne comprend que la surface de réparation du gardien auteur du dégagement. Ceci durera le temps que le gardien dégage le ballon. Cette vue sera appelée GoalKickBallView.
 - Lorsqu'un but est marqué, une vue centrée uniquement sur les cages, durant 2-3 secondes maximum, montrant la balle au fond de la cage, avec le gardien à côté. Cette vue sera appelée AfterGoalView.

2.3.4 LUMIERES ET OMBRES

L'éclairage utilisé est un éclairage standard, sans aucun effet spécial, qui permet seulement de rendre visibles les entités graphiques.

2.3.5 EFFETS SPECIAUX

Il n'y a aucun effet spécial dans ce jeu.

2.3.6 GESTION DU SON

Le son est géré de manière interne par le moteur son.

2.3.7 GESTION PHYSIQUE

Le côté physique est géré par un moteur physique qui gère :

- le calcul du point où la balle va se retrouver,
- la collision entre joueur et balle (les deux ne doivent pas se traverser),
- la collision entre deux joueurs (les deux ne doivent pas se traverser).

2.3.8 MULTI-JOUEURS

Dans ce jeu, deux modes multi-joueurs sont envisagés, le premier indispensable, le deuxième laissé à la possibilité des programmeurs :


1. Un mode 2 joueurs sur une même machine,

❖ Optionnellement :

2. Un mode réseau de type client-serveur, géré par un moteur réseau spécifique, qui gère une Game Room hébergée chez le premier joueur qui demande le commencement d'une partie

en réseau. Le deuxième joueur devra alors s'inviter dans la Game Room afin d'y affronter le premier joueur.

Ce deuxième mode est optionnel, et ne sera implémenté que sur le temps libre restant aux programmeurs.

 A spécifier ...

2.3.9 SUPPORT MULTI-LANGUES

La seule langue gérée par le jeu est l'anglais, conformément aux souhaits du client.

3. INTERFACE UTILISATEUR

3.1 GENERALITES





L'interface permet au gamer une prise en main intuitive et simple de l'application.

3.2 LES PERIPHERIQUES DE CONTROLE

Le seul périphérique de contrôle géré par l'application est le clavier.

Les commandes sont telles que décrites ci-dessous.

➤ COMMANDES PLAYER 1

Touches	Action sur l'avatar
	Déplacement vers la droite
	Déplacement vers la gauche
	Déplacement vers le bas
	Déplacement vers le haut
-	Tir
*	Passe
Entrer	Mettre la partie en pause

➤ COMMANDES PLAYER 2

Touches	Action sur l'avatar
D	Déplacement vers la droite
Q	Déplacement vers la gauche
S	Déplacement vers le bas
Z	Déplacement vers le haut
T	Tir
Y	Passe
Espace	Mettre la partie en pause

Le *Player 2* n'existe pas dans le mode *Player vs AI*.

- ❖ Au fur et à mesure qu'on rajoutera des fonctionnalités, les commandes de jeu seront complétées ou modifiées, selon les optionalités discutées dans les autres sections.

3.3 LES PROCEDURES FONCTIONNELLES

3.3.1 AU DEMARRAGE

L'utilisateur doit choisir le mode du jeu c'est-à-dire si il souhaite jouer contre l'ordinateur ou en mode multi-joueurs contre un autre joueur.

3.3.2 CONFIGURATION DU JEU

Une fois qu'il a choisi son mode de jeu, le joueur doit choisir une équipe pour lui et une autre équipe pour l'ordinateur dans le cas où il souhaite jouer avec un ordinateur ; dans le cas où il souhaite jouer en mode multi-joueurs l'utilisateur ne choisit que son équipe.

L'autre sera choisie par l'autre joueur.

Pour plus de détails sur la question des états du jeu, cf. 8.6. Interface utilisateur.

4. EDITEURS (UNIVERS ET ENTITES)

Dans ce jeu aucun éditeur de niveau n'est proposé, étant donné qu'ils ne cadrent pas avec la nature du jeu.

5. L'UNIVERS DU JEU

5.1 GENERALITES

Le monde dans lequel se déroule le jeu est similaire au monde réel.

- ❖ On pourra optionnellement imaginer un monde plus original : nous avons pensé à l'univers Lego®. Ainsi, il faudra remplacer les avatars standards par des avatars représentés par des personnages de Lego®, ce qui change bien entendu quantité de spécifications, et, si cette idée est implémentée, le jeu s'appellera alors *Lego® Foot 2D*.

5.2. ENDROITS – LIEUX

Le seul endroit du jeu est le stade.

5.3. ECHELLE

L'échelle doit permettre de visualiser le stade depuis le haut vertical, de façon à englober tout le terrain jouable et uniquement cela.

- ❖ L'échelle correspond en fait à la vue en cours. Si les autres vues discutées en section 2.3.3 Caméras, alors l'échelle sera modifiée en conséquence.

5.4. TEMPS ET CLIMAT

Les temps et climats ne sont pas gérés dans ce jeu.

Par défaut, un match peut se dérouler de jour (le temps sera alors ensoleillé) ou de nuit ; ceci sera à choisir dans un menu de l'état n°6 du StateManager : l'état ChooseEnvironment.

La durée d'un match est de 5 minutes par défaut.

- ❖ Optionnellement, on pourra par la suite proposer des durées différentes (5 min, 10 min ou 15 min) pour un match, et permettre aux joueurs la possibilité d'une mi-temps de 30 secondes à une minute.

6. LES ENTITES DE L'UNIVERS

6.1. GENERALITES

Les entités du jeu sont les avatars et la balle.

6.2. LES ENTITES JOUABLES

Les entités jouables sont les joueurs de terrain (i.e. attaquant ou défenseur) : *deux* par équipe.

Ce sont des entités graphiques gérées par le moteur graphique, et obligatoirement initialisée depuis une image chargée.

On ne fait pas la différence entre défenseur et attaquant.

Ils se trouvent en début de chaque période, et après un but, au centre du terrain. Les deux avatars de terrain se trouvent alors à une équidistance réelle de 0.5 m du centre du terrain. En face de la balle, l'attaquant se trouvera sur la droite de la balle, le défenseur sur la gauche de la balle.

Peut se déplacer sur le terrain sans la balle, avec la balle aux pieds, passer ou tirer la balle.

La passe et le tir de balle se fait via deux commandes différentes (cf. la section traitant des commandes de jeu).

Les passes et tirs sont orientés suivant la direction de la vitesse du joueur.

6.3. LES ENTITES NON JOUABLES

1. **La balle** : une entité graphique initialisée depuis une image chargée par le moteur graphique, ou alors directement dessinée via une méthode spécifique de sa classe (ceci est indifférent pour la suite, et laissé aux bons soins des programmeurs).

Au début du match, la balle est au centre du terrain.

Si jamais la balle sort en touche, elle rebondit sur les rebords du stade.

Si elle sort en corner, elle revient en six-mètre immédiatement.

Elle peut être passée ou tirée de la part d'un joueur.

Les calculs du point de chute de la balle et des trajectoires après rebonds se font via le moteur physique.

2. **Gardien de but** : un par équipe.

- 3.

❖ Optionnellement :

Tous les avatars de l'équipe adverse seront des entités non jouables.

Elles devront donc obéir à une intelligence artificielle, gérée par le moteur d'IA.

L'IA pourra être classée par ordre de difficulté, la difficulté du jeu étant à choisir dans le menu des options par le gamer désirant se confronter à l'IA.

Ceci devra être spécifié plus en profondeur si l'implémentation de cette fonctionnalité est décidée.

6.4. LES OBJETS

Aucun objet n'est à récolter dans le jeu.

- ❖ Optionnellement, les joueurs pourront récolter trophées de tournois, des médailles pour chaque niveau de difficulté contre AI ...

A spécifier plus en détail si les développeurs ont le temps de s'y atteler.

6.5. LES COMPETENCES

Les compétences doivent être définies selon le mode du jeu choisi :

- Concernant le mode *Player vs Player*, les avatars n'ont pas de compétences prédéfinies.
 - ❖ Optionnellement (si l'idée des trophées est implémentée), les compétences des avatars de terrains et du gardien pourront alors évoluer en fonction du nombre de trophées remportés. Une limite sera tout de même fixée à leur compétence.
 - ❖ Une autre idée originale, liée à l'univers Lego, serait d'attribuer, comme limite ultime des compétences de chacun des personnages jouables, une technique relevant de l'histoire personnelle du personnage dans l'univers Lego.
- En ce qui concerne le mode *Player vs AI*, les compétences des avatars non jouables sont classées selon la fonction de l'avatar :
 - **L'attaquant** (premier avatar de terrain) : il doit chercher à éviter les rencontres avec les avatars jouables, et se diriger vers les buts adverses dès qu'il en trouve la possibilité.

Il doit tirer dans le cadre des cages dès que le champ est libre (i.e. : pas d'avatars de terrains devant lui).

Si son champ de vision rencontre un avatar de terrain adverse, il doit tirer de façon à l'éviter, même si le tir ne sera pas cadré.

La vitesse de l'attaquant sera modérée (vitesse réelle : 3 km/h, à convertir en vitesse de jeu par le moteur physique).

- ❖ Optionnellement, si plusieurs difficultés sont implémentées, creuser ces pistes (à approfondir si cette optionalité est décidée) : une difficulté plus élevée peut consister en ce que l'attaquant tire toujours du côté des cages où le gardien adverse ne se trouve pas (i.e. : si le gardien adverse est vers la gauche de ses cages, l'attaquant tire dans le côté droit des cages, et vice versa), et, si le gardien adverse est au centre de ses cages, le tir aura lieu systématiquement aux angles droit des cages (i.e. passant très près des poteaux).

Une passe est effectuée automatiquement à l'avatar partenaire le plus proche (le calcul de l'avatar le plus proche se fait via l'*AI Engine*).

Pour ce qui est de la vitesse, on pourra faire en sorte que l'attaquant sprinte dès qu'il a le champ libre devant lui : si un avatar jouable adverse est sur ses flancs, il sprinte tout droit ; si un avatar adverse est sur sa diagonale, il sprinte en empruntant la diagonale symétrique (i.e. : si un avatar adverse est sur sa diagonale droite, l'attaquant sprinte en empruntant la diagonale gauche).

Le sprint consiste à tripler la vitesse normale de l'avatar. La montée progressive en vitesse devra être gérée par le moteur physique.

- **Le défenseur** (deuxième avatar de terrain) : doit se mettre toujours en travers du chemin de l'avatar adverse en essayant de lui prendre la balle (en cela, « prendre la balle » revient exactement à essayer de tirer sur la balle alors qu'elle est aux pieds de l'avatar adverse).

Doit toujours rester dans sa moitié de terrain.

- ❖ Dans les difficultés supérieures, l'implémentation des tacles par le défenseur pourra être envisagée.

Le défenseur pourra aussi tenter de tirer directement dans les buts adverses dès que le champ se libère devant lui (tout en restant dans moitié de terrain).

Une difficulté encore au-dessus consistera à ce que défenseur et attaquant intervertisse leur rôles afin de surprendre l'équipe adverse.

Ceci sera évidemment à spécifier avec beaucoup plus de rigueur si cette optionalité est décidée.

- **Le gardien de but** : reste toujours dans sa surface des six-mètres, et ne réagit (en sautant dans la direction appropriée), que lorsque la balle était dans son champ de vision.

Cette réaction ne doit pas être simultanée au tir de l'attaquant adverse : elle la suivra après une durée Δt à évaluer en fonction des essais (les programmeurs en ont la responsabilité). Cependant, cette durée de latence du gardien ne devra pas dépasser les quelques millisecondes.

- ❖ La durée de latence du gardien adverse sera diminuée en fonction du niveau de difficulté.

Pour une difficulté plus accrue, le gardien adverse pourra aussi cumuler les compétences des avatars de terrain, et ainsi pourra sortir de sa surface des six-mètre pour tacler un avatar adverse, ou même tirer au but dès que le champ lui est libéré.

Ceci sera à spécifier dans les détails en cas d'implémentation de cette option.

Pour un gardien du gamer, ces compétences ne pourront être acquises que si des trophées ont été remportés par le gamer, en jouant avec ce gardien.

6.6. LA CREATION D'ENTITES

La création d'entité se fait au moment du choix de l'équipe : dans l'état 4 du StateManager (l'état ChooseTeam).

6.7. LES EVOLUTIONS DES DIFFERENTES ENTITES

Les avatars n'évoluent pas dans la version basique du jeu.

- ❖ Les avatars pourront évoluer à la condition que le gamer remporte des trophées en jouant avec eux.

Les évolutions des avatars seront à implémenter selon le type d'avatar :

- Pour l'attaquant, sa vitesse de pointe sera d'autant plus rapide qu'il aura remporté de trophées.
- Pour le défenseur, les tacles seront plus puissants (leur vitesse et efficacité sera multipliée en fonction du nombre de trophées remportés).

6.8. ORIENTATION ET DEPLACEMENTS

Les avatars vont se déplacer sur le terrain uniquement.

Etant donné que la vue sur le terrain est verticale globale, alors le déplacement d'un avatar relativement au gamer est celui-ci :

DEPLACEMENT DE L'AVATAR	CONTROLE PAR LE GAMER
DROITE	Haut
GAUCHE	Bas
TOUT DROIT	Gauche ou droite selon la direction de la vitesse de l'avatar

Cf. Les périphériques de contrôle pour les commandes utilisateur correspondantes.

L'*orientation* du déplacement d'un avatar est donnée par un vecteur, qui est différent du vecteur de la *vitesse* déplacement de l'avatar.

6.9. INTERACTIONS INTER-ENTITES ET ENTITES-UNIVERS

- **INTERACTION INTER-ENTITES :**
Les collisions inter-entités seront gérées par le moteur physique.

<i>Type de la collision</i>	<i>Conséquence sur le jeu</i>
<i>Collision entre avatars de la même équipe</i>	Il ne se passe rien de particulier, un avatar entrant en collision avec son partenaire de jeu ne peut plus se déplacer : il doit le contourner ; Il ne doit pas le traverser et s'il possède la balle, la collision ne lui enlèvera pas la balle.
<i>Collision entre avatars, l'un des deux possédant la balle</i>	Si celui qui n'a pas la balle essaye de tirer ou de passer, il récupère la balle de l'adversaire. On ne fait pas de différence ici entre un avatar jouable (attaquant ou défenseur) et avatar non-jouable (gardien) .
<i>Collision entre avatar et balle</i>	La balle reste aux pieds de l'avatar. Cf. 6.5. Les compétences pour les actions sur la balle

➤ INTERACTION ENTITES-UNIVERS :

<i>Type de la collision</i>	<i>Conséquence sur le jeu</i>
<i>Collision entre la balle et un mur de corner</i>	La balle revient immédiatement aux pieds du gardien adverse à l'avatar ayant provoqué la collision
<i>Collision entre la balle et un poteau ou une barre des cages</i>	Rebond de la balle (dont la trajectoire est calculée par le <i>Physics Engine</i>).
<i>Collision entre la balle et un mur de touche</i>	Rebond de la balle (dont la trajectoire est calculée par le <i>Physics Engine</i>).

6.10. MOYENS DE LOCOMOTION

Aucun moyen de locomotion n'est implémenté dans ce jeu.

7. HISTOIRE

7.1. SYNOPSIS

Il n'y a aucun synopsis dans ce jeu.

7.2. DECOUPAGE INTERACTIF DU SCENARIO, DIALOGUES

Il n'y a aucun scénario dans ce jeu.

7.3. REFERENCES ET CLINS D'ŒIL

- ❖ Si l'idée de l'univers Lego© est retenue et implémentée, alors des références pourront être implémentées en dehors de la phase de jeu (dans les images de fond des menus, dans les postures des entités, etc...).

✚ A spécifier plus clairement.

8. CHOIX ARTISTIQUES : GRAPHISME ET ANIMATION

8.1. GENERALITES ET OBJECTIFS

Les entités graphiques seront soit à charger depuis une image (via le moteur graphique), soit dessinées, en début du programme, par une méthode qui leur est privée et stockée en mémoire tout au long du jeu.

8.2. ASPECTS 2D

Les entités à charger depuis une image sont :

1. Le stade : image de jour, ou image de nuit, selon ce que le gamer aura choisi dans l'état 5 ChooseEnvironment du StateManager.
2. Les joueurs : les images standards sont à charger en fonction des équipes constituées. Chaque joueur choisi possède plusieurs images de lui-même en mouvement (24 images par action du joueur, dans l'idéal).
Par exemple, le mouvement « *tirer sur la balle* » requiert à lui seul 24 images du joueur, décrivant chaque portion du mouvement.

8.3. ANIMATIONS

Les animations concernent quelques entités graphiques, et seront gérées par le moteur graphique.

Les entités pouvant s'animer sont :

- Les joueurs (de terrain et gardien),
- La balle.

Les entités ne pouvant pas s'animer sont :

- Le stade.
 - ❖ On peut animer le stade de cette façon : lorsqu'un joueur tire, de l'herbe s'envole autour de son pied au point où il a frappé la balle.
✚ A spécifier ...

L'animation d'une entité devra utiliser plusieurs images de l'entité et les charger à la création de l'entité en mémoire.

Ces images seront affichées successivement afin de donner l'illusion de l'animation.

L'idéal serait que chaque entité possède 24 images à afficher, afin que la fluidité des animations soit optimale.

8.4. CINEMATIQUES

Les cinématiques ne sont pas traitées dans ce jeu.

- ❖ On pourra envisager des animations pour :

- Après un but, une scène montrant la joie de l'équipe marquante et l'acclamation du public.
- Après un arrêt réussi, le gardien dégageant la balle.
 - ✚ A spécifier plus clairement, notamment la nature des cinématiques (images de synthèse, vidéos intégrées ... ?).

8.5. EFFETS SPECIAUX

Il n'y a pas d'effets spéciaux dans ce jeu.

8.6. INTERFACE UTILISATEUR

L'interface utilisateur est détaillée plus en détail en 3. Interface utilisateur.

L'interface utilisateur est constituée uniquement de l'écran du jeu.

- ❖ Une idée intéressante serait d'effectuer les choix utilisateurs depuis les états du menu du jeu.
Les états du jeu sont au nombre de 8 :
 1. Etat d'introduction (*Intro*) :
Un fond d'écran caractéristique du jeu, où il faut appuyer sur une touche afin d'accéder à l'état suivant,
 2. Etat de menu (*Menu*) :
Un écran de transitions permettant de choisir entre 3 états : Start (qui conduit à l'état *ChooseMode*), Options (qui conduit à l'état *Options*), ou Exit (qui met fin à l'application).
 3. Etat du choix du mode de jeu (*ChooseMode*),
 4. Etat du choix d'équipe (*ChooseTeam*) :
Dans le mode 1 *Player vs Player*, deux équipes seront constituées. L'écran de l'état sera alors un écran splitté, chaque écran servant de choix d'équipe pour un gamer.
 5. Etat du choix de l'environnement, i.e. stade de nuit ou de jour (*ChooseEnvironment*),
 6. Etat d'options (*Options*) :
Cet état devra essentiellement offrir le choix de diminuer le volume sonore.
 - ❖ Il pourra aussi offrir la possibilité de changer les commandes standards du jeu, et de remettre les commandes par défaut.
 7. Etat de jeu en cours (*PlayingGame*) :
La partie de football se déroule,
 8. Etat de pause au-cours du jeu (*PauseGame*) :
Un état de pause simple, pendant lequel la partie est suspendue, et donc les animations sont stoppées, ainsi que le chronomètre,
 9. Etat de fin d'une partie (*OverGame*) :
Fin de partie, propose de recommencer la partie, ce qui ramène le joueur à l'état 2 de menu, ou bien de quitter la partie, ce qui détruit l'application et met fin au jeu.
 - ❖ Optionnellement : affichage des statistiques de jeux pour chaque équipe dans cet état.
 - ✚ A préciser plus clairement : quelles statistiques ? Calculées par quelles composantes du système ?

Les différents états de jeu seront gérés par un *State Manager*, interne au core kernel.

Ce *State Manager* exécutera les transitions entre les différents états du jeu suivant l'automate du document : « [..\2.conception\iteration2\2.Conception_gestionnaire_etats.docx](#) ».

9. CHOIX ARTISTIQUES : MUSIQUES ET BRUITAGES

9.1. GENERALITES ET OBJECTIFS

Les musiques et bruitages sont gérés par le *Sound Engine*, qui est lui-même géré par le *core kernel* (comme tous les moteurs de l'application).

9.2. INTERFACE UTILISATEUR ET AMBIANCE SONORE

Les états décrits précédemment auront droit à des musiques de fonds spécifiques.

Certaines musiques de fond sont communes pour plusieurs états.

Les états qui auront une musique de fond unique sont :

- *Intro*
- *PlayingGame*
- *Pause*
- *OverGame*

Les états partageant la même musique de fond sont *Menu* et *Options* d'un côté, et *ChooseMode* et *ChooseTeam* de l'autre.

9.3. MUSIQUES

Voici les choix de musiques, pour chaque état :

- *Intro* : *ZeldaTheme1*,
- *Menu/Options* : *SoccerTheme1*,
- *ChooseMode/ChooseTeam* : non décidé,
- *PlayingGame* : non décidé,
- *Pause* : non décidé,
- *OverGame* : *ZeldaTheme2*.

Les musiques choisies sont temporaires et sont encore susceptibles de changer.

9.4. BRUITAGES

Les bruitages obligatoirement gérés doivent être :

Tirs ou passes de balle par un joueur.

1. Lorsque se produit un but : un son « *GOOOOAAAAAAL* » est produit, parallèlement à un son d'acclamations des tribunes des spectateurs.
Ces deux sons doivent être produit en parallèles, le premier dure le temps de l'enregistrement du son, le deuxième dure 10 secondes exactement depuis le que le but ait été marqué.

❖ Optionnellement, on pourra gérer ces cas :

1. Lorsque la balle touche un poteau des cages,
 2. Lorsque la balle retombe au sol, après un tir ou un dégagement.
- ✚ A spécifier plus rigoureusement.

9.5. VOIX

Aucune voix n'est prise en charge dans le jeu.

9.6. EFFETS SPECIAUX

A priori, aucun effet spécial n'est géré par le *Sound Engine*.